tldr pages book

Simplified and community-driven man pages

Generated on Mon Nov 10 20:37:56 2025

Website: https://tldr.sh

GitHub: https://github.com/tldr-pages/tldr

Android

am

Android activity manager.

More information: https://developer.android.com/tools/adb#am.

- Start the activity with a specific component and package [n]ame:

```
am start -n {{com.android.settings/.Settings}}
```

- Start an intent [a]ction and pass [d]ata to it:

```
am start -a {{android.intent.action.VIEW}} -d {{tel:123}}
```

- Start an activity matching a specific [a]ction and [c]ategory:

```
am start -a {{android.intent.action.MAIN}} -c
{{android.intent.category.HOME}}
```

- Convert an intent to a URI:

```
am to-uri -a {{android.intent.action.VIEW}} -d {{tel:123}}
```

bugreport

Show an Android bug report.

This command can only be used through **`adb shell`**.

More information: https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreport.

- Display a complete bug report of an Android device:

```
bugreport
```

bugreportz

Generate a zipped Android bug report.

This command can only be used through **`adb shell`**.

More information: https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreportz.

- Generate a complete zipped bug report of an Android device:

```
bugreportz
```

- Show the progress of a running `bugreportz` operation:

```
bugreportz -p
```

- Write the content of an Android bug report to `stdout`:

```
bugreportz -s
```

- Display help:

```
bugreportz -h
```

- Display version:

```
bugreportz -v
```

cmd

Android service manager.

More information: https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/cmd/.

- [l]ist all running services:

```
cmd -l
```

- Call a specific service:

```
cmd {{service}}
```

- Call a service with specific arguments:

```
cmd {{service}} {{argument1 argument2 ...}}
```

dalvikvm

Android Java virtual machine.

More information: https://source.android.com/docs/core/runtime.

- Start a specific Java program:

```
dalvikvm -classpath {{path/to/file.jar}} {{classname}}
```

dumpsys

Get information about Android system services.

This command can only be used through **`adb shell`**.

More information: https://developer.android.com/tools/dumpsys.

- Get diagnostic output for all system services:

```
dumpsys
```

- Get diagnostic output for a specific system service:

```
dumpsys {{service}}
```

- List all services `dumpsys` can give information about:

```
dumpsys -l
```

- List service-specific arguments for a service:

```
dumpsys {{service}} -h
```

- Exclude a specific service from the diagnostic output:

```
dumpsys --skip {{service}}
```

- Specify a [t]imeout period in seconds (defaults to 10s):

```
dumpsys -t {{8}}
```

getprop

Show information about Android system properties.

More information: https://manned.org/getprop.

- Display information about Android system properties:

```
getprop
```

- Display information about a specific property:

```
getprop {{property}}
```

- Display the SDK API level:

```
getprop {{ro.build.version.sdk}}
```

- Display the Android version:

```
getprop {{ro.build.version.release}}
```

- Display the Android device model:

```
getprop {{ro.vendor.product.model}}
```

- Display the OEM unlock status:

```
getprop {{ro.oem_unlock_supported}}
```

- Display the MAC address of the Android's Wi-Fi card:

```
getprop {{ro.boot.wifimacaddr}}
```

input

Send event codes or touchscreen gestures to an Android device.

This command can only be used through **`adb shell`**.

More information: https://developer.android.com/reference/android/view/KeyEvent.html#constants_1.

- Send an event code for a single character to an Android device:

```
input keyevent {{event_code}}
```

- Send a text to an Android device (`%s` represents spaces):

```
input text "{{text}}"
```

- Send a single tap to an Android device:

```
input tap {{x_position}} {{y_position}}
```

- Send a swipe gesture to an Android device:

```
input swipe {{x_start}} {{y_start}} {{x_end}} {{y_end}} {{duration_in_ms}}
```

- Send a long press to an Android device using a swipe gesture:

```
input swipe {{x_position}} {{y_position}} {{x_position}} {{y_position}} {{duration_in_ms}}
```

logcat

Dump a log of system messages, including stack traces when an error occurred, and information messages logged by applications.

More information: https://developer.android.com/tools/logcat.

- Display system logs:

```
logcat
```

- Write system logs to a [f]ile:

```
logcat -f {{path/to/file}}
```

- Display lines that match a `regex`:

```
logcat --regex {{regex}}
```

- Display logs for a specific PID:

```
logcat --pid {{pid}}
```

- Display logs for the process of a specific package:

```
logcat --pid $(pidof -s {{package}})
```

pkg

Package management utility for Termux.

More information: https://wiki.termux.com/wiki/Package_Management.

- Upgrade all installed packages:

```
pkg upgrade
```

- Install a package:

```
pkg install {{package}}
```

- Uninstall a package:

```
pkg uninstall {{package}}
```

- Reinstall a package:

```
pkg reinstall {{package}}
```

- Search for a package:

```
pkg search {{package}}
```

pm

Display information about apps on an Android device.

More information: https://developer.android.com/tools/adb#pm.

- List all installed apps:

```
pm list packages
```

- List all installed [s]ystem apps:

```
pm list packages -s
```

- List all installed [3]rd-party apps:

```
pm list packages -3
```

- List apps matching specific keywords:

```
pm list packages {{keyword1 keyword2 ...}}
```

- Display a path of the APK of a specific app:

```
pm path {{app}}
```

screencap

Take a screenshot of a mobile display.

This command can only be used through **`adb shell`**.

More information: https://developer.android.com/tools/adb#screencap.

- Take a screenshot:

```
screencap {{path/to/file}}
```

settings

Get information about the Android OS.

More information: https://web.archive.org/web/20240525010124/https://adbinstaller.com/commands/adb-shell-settings-5b670d5ee7958178a2955536.

- List the settings in the `global` namespace:

```
settings list {{global}}
```

- Get a value of a specific setting:

```
settings get {{global}} {{airplane_mode_on}}
```

- Set a specific value of a setting:

```
settings put {{system}} {{screen_brightness}} {{42}}
```

- Delete a specific setting:

```
settings delete {{secure}} {{screensaver_enabled}}
```

wm

Show information about the screen of an Android device.

This command can only be used through **`adb shell`**.

More information: https://web.archive.org/web/20240420064706/https://adbinstaller.com/commands/adb-shell-wm-5b672b17e7958178a2955538.

- Display the physical size of an Android device's screen:

```
wm size
```

- Display the physical density of an Android device's screen:

```
wm density
```

Cisco-ios

clock

Set the system clock.

More information: https://www.cisco.com/c/en/us/td/docs/ios/fundamentals/command/reference/cf_book/cf_c1.html#clock.

- Enter privileged execution mode:

```
clock set {{23}}:{{59}}:{{59}} {{31}} {{april}} {{2000}}
```

- Auto negotiate with the far end of the link, defaulting to active-clock:

```
clock active prefer
```

- Auto negotiate with the far end of the link, defaulting to passive-clock:

```
clock passive prefer
```

- Show the current clock mode negotiated by the firmware:

```
clock show interfaces
```

configure

Enter configuration mode.

More information: https://www.cisco.com/c/en/us/td/docs/ios/fundamentals/command/reference/cf_book/cf_c1.html#configure_check_syntax.

- Configure from the terminal:

  ```
  {{[conf|configure]}} {{[t|terminal]}}
  ```

- Back out one level in configuration mode:

  ```
  exit
  ```

- Exit out of configuration mode:

  ```
  end
  ```

crypto

Manage cryptography.

Accessed in configuration mode.

More information: https://www.cisco.com/c/en/us/td/docs/security/asa/asa-cli-reference/A-H/asa-command-ref-A-H/crypto-is-cz-commands.html.

- Generate an `rsa` key:

```
crypto key generate rsa
```

- Define a modulus for a key:

```
crypto key generate rsa modulus {{1024}}
```

- Remove all keys:

```
crypto key zeroize
```

delete

Delete individual files.

More information: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/command/reference/fund/n5k-fund-cr/n5k-fund_cmds_d.html#delete.

- Delete a file from flash memory:

```
delete {{vlan.dat}}
```

dir

List files.

More information: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/command/reference/fund/n5k-fund-cr/n5k-fund_cmds_d.html#dir.

- List files in the current working directory:

```
dir
```

- List files in a specific filesystem:

```
dir {{flash}}:
```

- List available filesystems:

```
dir ?
```

enable

Enter privileged execution mode.

More information: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/security/d1/sec-d1-cr-book/sec-cr-e1.html#wp3307186499.

- Enter privileged execution mode:

```
enable
```

erase

Delete preset things.

More information: https://www.cisco.com/c/en/us/td/docs/ios/ios_xe/fundamentals/configuration/guide/2_xe/cf_xe_book/cf_config-files_xe.html.

- Erase the startup configuration:

```
erase startup-config
```

- Erase a filesystem:

```
erase {{flash}}:
```

interface

Manage interfaces.

Accessed in configuration mode.

More information: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/interface/command/ir-cr-book/ir-i1.html.

- Configure a VLAN:

  ```
  interface vlan {{1}}
  ```

- Set an interface to be active or inactive (this is run inside the interface command):

  ```
  {{no shutdown|shutdown}}
  ```

ip

Manage IP configurations.

Accessed in configuration mode.

More information: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr/command/ipaddr-cr-book.html.

- Set the SSH version:

```
ip ssh version {{2}}
```

- Set the address of the device (This is done under `interface command`):

```
ip address {{10.0.0.1}} {{255.255.255.0}}
```

- Set the address to be determined with DHCP (This is done under `interface command`):

```
ip address dhcp
```

- Define a domain name:

```
ip domain-name {{example.com}}
```

line

Manage lines.

Accessed in configuration mode.

More information: https://www.cisco.com/c/en/us/td/docs/routers/sdwan/command/iosxe/qualified-cli-command-reference-guide/m-line-commands.pdf.

- Configure lines from 0 to 15:

```
line vty 0 15
```

login

Manage console and virtual line authentication.

Accessed in configuration mode under **`line`**.

More information: https://www.cisco.com/c/en/us/td/docs/routers/sdwan/command/iosxe/qualified-cli-command-reference-guide/m-line-commands.pdf.

- Use local username and password for authentication:

```
login local
```

?

Get context sensitive.

More information: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/fundamentals/configuration/15mt/fundamentals-15-mt-book/cf-cli-basics.html#GUID-223128D2-FB6D-418D-86C6-06D1D0DA51B3.

- Get available commands:

```
?
```

- Get storages that are listable:

```
dir ?
```

- Show what IP information is viewable:

```
ip show ?
```

reload

Control reboot behavior.

More information: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/command/reference/fund/n5k-fund-cr/n5k-fund_cmds_r.html#reload.

- Reboot the system:

```
reload
```

- Reboot after n seconds:

```
reload in {{n}}
```

- Cancel pending reboots:

```
reload cancel
```

show

Show various system information.

More information: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/command/reference/fund/n5k-fund-cr/n5k-fund_cmds_show.html.

- Show switch IP addresses:

```
{{[sh|show]}} ip interface brief
```

- Show specific interface configuration:

```
{{[sh|show]}} ip interface {{vlan1}}
```

- Show vlan configuration:

```
{{[sh|show]}} vlan
```

- Show currently running configuration:

```
{{[sh|show]}} running-config
```

- Show SSH configuration:

```
{{[sh|show]}} ip ssh
```

transport

Manage line transport protocols.

Accessed in configuration mode under **`line`**.

More information: https://www.cisco.com/c/en/us/td/docs/routers/sdwan/command/iosxe/qualified-cli-command-reference-guide/m-line-commands.pdf.

- Restrict line protocols to `ssh`:

```
transport input ssh
```

- Restrict line protocols to `telnet`:

```
transport input telnet
```

username

Manage users.

Accessed in configuration mode.

More information: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/security/s1/sec-s1-xe-3se-3850-cr-book/sec-s1-xe-3se-3850-cr-book_chapter_0110.html.

- Set up an admin account:

```
username {{admin}} privilege 15 secret {{password}}
```

write

Write data to memory.

More information: https://www.oreilly.com/library/view/cisco-ios-in/0596008694/re869.html#book-content.

- Write current configuration to memory:

```
write memory
```

- Delete the configuration in memory:

```
write erase
```

- Display help:

```
write ?
```

Common

!

Reuse and expand the shell history in **sh**, Bash, Zsh, **rbash**, and **ksh**.

More information: https://gnu.org/software/bash/manual/bash.html#Event-Designators.

- Substitute with the previous command and run it with `sudo`:

```
sudo !!
```

- Substitute with a command based on its line number found with `history`:

```
!{{number}}
```

- Substitute with a command that was used a specified number of lines back:

```
!-{{number}}
```

- Substitute with the most recent command that starts with a string:

```
!{{string}}
```

- Substitute with the arguments of the latest command:

```
{{command}} !*
```

- Substitute with the last argument of the latest command:

```
{{command}} !$
```

- Substitute with the last command but without the last argument:

```
!:-
```

- Print last command that starts with a string without executing it:

```
!{{string}}:p
```

$

Expand a Bash variable.

More information: https://gnu.org/software/bash/manual/bash.html#Shell-Variables.

- Print a variable:

```
echo ${{VARIABLE}}
```

- Run variable contents as a command:

```
${{VARIABLE}}
```

- Print the exit status of the previous command:

```
echo $?
```

- Print a random number between 0 and 32767:

```
echo $RANDOM
```

- Print one of the prompt strings:

```
echo ${{PS0|PS1|PS2|PS3|PS4}}
```

- Expand with the output of `command` and run it. Same as enclosing `command` in backtics:

```
$({{command}})
```

- List how many arguments the current context has:

```
echo $#
```

- Print out a Bash array:

```
echo ${{{array_name[@]}}}
```

%

Manage jobs.

More information: https://www.gnu.org/software/bash/manual/bash.html#Job-Control-Basics.

- Bring the current job to front:

```
%
```

- Bring the previous job to front:

```
%-
```

- Bring the job number `n` to front:

```
%{{n}}
```

- Bring a job whose command starts with `string` to front:

```
%{{string}}
```

- Bring a job whose command contains `string` to front:

```
%?{{string}}
```

- Resume a suspended job:

```
%{{1}} &
```

,

Run commands without installing them.

More information: https://github.com/nix-community/comma.

- Run a command:

```
, {{command -with -flags}}
```

- Add a command to a child shell:

```
, {{[-s|--shell]}} {{command}}
```

- Clear the cache:

```
, {{[-e|--empty-cache]}}
```

2to3

Automated Python 2 to 3 code conversion.

This module has been deprecated since 3.11 and has been removed since 3.13.

For reference, see: https://github.com/python/cpython/blob/8d42e2d915c3096e7eac1c649751d1da567bb7c3/Doc/whatsnew/3.13.rst?plain=1#L188.

More information: https://manned.org/2to3.

- Display the changes that would be performed without performing them (dry-run):

```
2to3 {{path/to/file.py}}
```

- Convert a Python 2 file to Python 3:

```
2to3 --write {{path/to/file.py}}
```

- Convert specific Python 2 language features to Python 3:

```
2to3 --write {{path/to/file.py}} --fix {{raw_input}} --fix {{print}}
```

- Convert all Python 2 language features except the specified ones to Python 3:

```
2to3 --write {{path/to/file.py}} --nofix {{has_key}} --nofix {{isinstance}}
```

- List all available language features that can be converted from Python 2 to Python 3:

```
2to3 --list-fixes
```

- Convert all Python 2 files in a directory to Python 3:

```
2to3 --output-dir {{path/to/python3_directory}} --write-unchanged-files --nobackups {{path/to/python2_directory}}
```

- Run 2to3 with multiple threads:

```
2to3 --processes {{4}} --output-dir {{path/to/python3_directory}} --write --nobackups --no-diff {{path/to/python2_directory}}
```

7z

File archiver with a high compression ratio.

More information: https://manned.org/7z.

- [a]dd a file or directory to a new or existing archive:

```
7z a {{path/to/archive.7z}} {{path/to/file_or_directory}}
```

- Encrypt an existing archive (including filenames):

```
7z a {{path/to/encrypted.7z}} -p{{password}} -mhe=on {{path/to/archive.7z}}
```

- E[x]tract an archive preserving the original directory structure:

```
7z x {{path/to/archive.7z}}
```

- E[x]tract an archive to a specific directory:

```
7z x {{path/to/archive.7z}} -o{{path/to/output}}
```

- E[x]tract an archive to `stdout`:

```
7z x {{path/to/archive.7z}} -so
```

- [a]rchive using a specific archive type:

```
7z a -t{{7z|bzip2|gzip|lzip|tar|zip}} {{path/to/archive}} {{path/to/file_or_directory}}
```

- [l]ist the contents of an archive:

```
7z l {{path/to/archive.7z}}
```

- Set the level of compression (higher means more compression, but slower):

```
7z a {{path/to/archive.7z}} -mx={{0|1|3|5|7|9}} {{path/to/file_or_directory}}
```

7za

File archiver with a high compression ratio.

Similar to **7z** except that it supports fewer file types but is cross-platform.

More information: https://manned.org/7za.

- [a]rchive a file or directory:

```
7za a {{path/to/archive.7z}} {{path/to/file_or_directory}}
```

- Encrypt an existing archive (including file names):

```
7za a {{path/to/encrypted.7z}} -p{{password}} -mhe={{on}}
{{path/to/archive.7z}}
```

- E[x]tract an archive preserving the original directory structure:

```
7za x {{path/to/archive.7z}}
```

- E[x]tract an archive to a specific directory:

```
7za x {{path/to/archive.7z}} -o{{path/to/output}}
```

- E[x]tract an archive to `stdout`:

```
7za x {{path/to/archive.7z}} -so
```

- [a]rchive using a specific archive type:

```
7za a -t{{7z|bzip2|gzip|lzip|tar|...}} {{path/to/archive.7z}}
{{path/to/file_or_directory}}
```

- [l]ist the contents of an archive:

```
7za l {{path/to/archive.7z}}
```

- Set the level of compression (higher means more compression, but slower):

```
7za a {{path/to/archive.7z}} -mx={{0|1|3|5|7|9}} {{path/to/
file_or_directory}}
```

7zr

File archiver with a high compression ratio.

Similar to **7z** except that it only supports 7z files.

More information: https://manned.org/7zr.

- [a]rchive a file or directory:

```
7zr a {{path/to/archive.7z}} {{path/to/file_or_directory}}
```

- Encrypt an existing archive (including file names):

```
7zr a {{path/to/encrypted.7z}} -p{{password}} -mhe={{on}}
{{path/to/archive.7z}}
```

- E[x]tract an archive preserving the original directory structure:

```
7zr x {{path/to/archive.7z}}
```

- E[x]tract an archive to a specific directory:

```
7zr x {{path/to/archive.7z}} -o{{path/to/output}}
```

- E[x]tract an archive to `stdout`:

```
7zr x {{path/to/archive.7z}} -so
```

- [l]ist the contents of an archive:

```
7zr l {{path/to/archive.7z}}
```

- Set the level of compression (higher means more compression, but slower):

```
7zr a {{path/to/archive.7z}} -mx={{0|1|3|5|7|9}} {{path/to/
file_or_directory}}
```

[

Check file types and compare values.

Returns a status of 0 if the condition evaluates to true, 1 if it evaluates to false.

More information: https://gnu.org/software/bash/manual/bash.html#index-test.

- Test if a given variable is equal/not equal to the specified string:

```
[ "${{variable}}" {{=|!=}} "{{string}}" ]
```

- Test if a given variable is [eq]ual/[n]ot [e]qual/[g]reater [t]han/[l]ess [t]han/[g]reater than or [e]qual/[l]ess than or [e]qual to the specified number:

```
[ "${{variable}}" -{{eq|ne|gt|lt|ge|le}} {{integer}} ]
```

- Test if the specified variable has a [n]on-empty value:

```
[ -n "${{variable}}" ]
```

- Test if the specified variable has an empty value ([z]ero length):

```
[ -z "${{variable}}" ]
```

- Test if the specified [f]ile exists:

```
[ -f {{path/to/file}} ]
```

- Test if the specified [d]irectory exists:

```
[ -d {{path/to/directory}} ]
```

- Test if the specified file or directory [e]xists:

```
[ -e {{path/to/file_or_directory}} ]
```

[[

Check file types and compare values.

Returns a status of 0 if the condition evaluates to true, 1 if it evaluates to false.

More information: https://gnu.org/software/bash/manual/bash.html#index-_005b_005b.

- Test if a given variable is equal/not equal to the specified string:

```
[[ ${{variable}} {{==|!=}} "{{string}}" ]]
```

- Test if a given string conforms the specified glob/regex:

```
[[ ${{variable}} {{==|=~}} {{pattern}} ]]
```

- Test if a given variable is [eq]ual/[n]ot [e]qual/[g]reater [t]han/[l]ess [t]han/[g]reater than or [e]qual/[l]ess than or [e]qual to the specified number:

```
[[ ${{variable}} -{{eq|ne|gt|lt|ge|le}} {{integer}} ]]
```

- Test if the specified variable has a [n]on-empty value:

```
[[ -n ${{variable}} ]]
```

- Test if the specified variable has an empty value ([z]ero length):

```
[[ -z ${{variable}} ]]
```

- Test if the specified [f]ile exists:

```
[[ -f {{path/to/file}} ]]
```

- Test if the specified [d]irectory exists:

```
[[ -d {{path/to/directory}} ]]
```

- Test if the specified file or directory [e]xists:

```
[[ -e {{path/to/file_or_directory}} ]]
```

]

This shell keyword is used to close out **[**.

- View documentation for the `[` keyword:

  ```
  tldr [
  ```

]]

This shell keyword is used to close out **[[**.

- View documentation for the `[[` keyword:

  ```
  tldr [[
  ```

^

Bash builtin to quick substitute a string in the previous command and run the result.

Equivalent to **`!!:s^string1^string2`**.

More information: https://gnu.org/software/bash/manual/bash.html#Event-Designators.

- Run the previous command replacing `string1` with `string2`:

  `^{{string1}}^{{string2}}`

- Remove `string1` from the previous command:

  `^{{string1}}^`

- Replace `string1` with `string2` in the previous command and add `string3` to its end:

  `^{{string1}}^{{string2}}^{{string3}}`

- Replace all occurrences of `string1`:

  `^{{string1}}^{{string2}}^:g&`

- Print the substituted command without running it:

  `^{{string1}}^{{string2}}^:p`

a2ping

Convert images into EPS or PDF files.

More information: https://manned.org/a2ping.

- Convert an image to PDF (Note: Specifying an output filename is optional):

```
a2ping {{path/to/image.ext}} {{path/to/output.pdf}}
```

- Compress the document using the specified method:

```
a2ping --nocompress {{none|zip|best|flate}} {{path/to/file}}
```

- Scan HiResBoundingBox if present (defaults to yes):

```
a2ping --nohires {{path/to/file}}
```

- Allow page content below and left of the origin (defaults to no):

```
a2ping --below {{path/to/file}}
```

- Pass extra arguments to `gs`:

```
a2ping --gsextra {{arguments}} {{path/to/file}}
```

- Pass extra arguments to external program (i.e `pdftops`):

```
a2ping --extra {{arguments}} {{path/to/file}}
```

- Display help:

```
a2ping {{[-h|--help]}}
```

aapt

Android Asset Packaging Tool: compile and package an Android app's resources.

More information: https://manned.org/aapt.

- List files contained in an APK archive:

```
aapt list {{path/to/app.apk}}
```

- Display an app's metadata (version, permissions, etc.):

```
aapt dump badging {{path/to/app.apk}}
```

- Create a new APK archive with files from the specified directory:

```
aapt package -F {{path/to/app.apk}} {{path/to/directory}}
```

ab

Apache HTTP server benchmarking tool.

More information: https://httpd.apache.org/docs/current/programs/ab.html.

- Execute 100 HTTP GET requests to a given URL:

```
ab -n 100 {{url}}
```

- Execute 100 HTTP GET requests, in concurrent batches of 10, to a URL:

```
ab -n 100 -c 10 {{url}}
```

- Execute 100 HTTP POST requests to a URL, using a JSON payload from a file:

```
ab -n 100 -T {{application/json}} -p {{path/to/file.json}} {{url}}
```

- Use HTTP [k]eep-Alive, i.e. perform multiple requests within one HTTP session:

```
ab -k {{url}}
```

- Set the maximum number of seconds ([t]imeout) to spend for benchmarking (30 by default):

```
ab -t {{60}} {{url}}
```

- Write the results to a CSV file:

```
ab -e {{path/to/file.csv}}
```

abduco

Terminal session manager.

More information: https://manned.org/abduco.

- List sessions:

```
abduco
```

- [A]ttach to a session, creating it if it doesn't exist:

```
abduco -A {{name}} {{bash}}
```

- [A]ttach to a session with `dvtm`, creating it if it doesn't exist:

```
abduco -A {{name}}
```

- Detach from a session:

```
<Ctrl \>
```

- [A]ttach to a session in [r]ead-only mode:

```
abduco -Ar {{name}}
```

ac

Print statistics on how long users have been connected.

More information: https://manned.org/ac.8.

- Print how long the current user has been connected in hours:

```
ac
```

- Print how long users have been connected in hours:

```
ac -p
```

- Print how long a particular user has been connected in hours:

```
ac -p {{username}}
```

- Print how long a particular user has been connected in hours per [d]ay (with total):

```
ac -dp {{username}}
```

accelerate

A library that enables the same PyTorch code to be run across any distributed configuration.

More information: https://huggingface.co/docs/accelerate/index.

- Print environment information:

  ```
  accelerate env
  ```

- Interactively create a configuration file:

  ```
  accelerate config
  ```

- Print the estimated GPU memory cost of running a Hugging Face model with different data types:

  ```
  accelerate estimate-memory {{name/model}}
  ```

- Test an Accelerate configuration file:

  ```
  accelerate test --config_file {{path/to/config.yaml}}
  ```

- Run a model on CPU with Accelerate:

  ```
  accelerate launch {{path/to/script.py}} {{--cpu}}
  ```

- Run a model on multi-GPU with Accelerate, with 2 machines:

  ```
  accelerate launch {{path/to/script.py}} --multi_gpu --
  num_machines 2
  ```

ack

A search tool like **`grep`**, optimized for developers.

See also: **`rg`**, which is much faster.

More information: https://beyondgrep.com/documentation.

- Search for files containing a string or `regex` in the current directory recursively:

```
ack "{{search_pattern}}"
```

- Search for a case-insensitive pattern:

```
ack {{[-i|--ignore-case]}} "{{search_pattern}}"
```

- Search for lines matching a pattern, printing only the matched text and not the rest of the line:

```
ack {{[-o|--output '$&']}} "{{search_pattern}}"
```

- Limit search to files of a specific type:

```
ack {{[-t|--type]}} {{ruby}} "{{search_pattern}}"
```

- Do not search in files of a specific type:

```
ack {{[-t|--type]}} no{{ruby}} "{{search_pattern}}"
```

- Count the total number of matches found:

```
ack {{[-c|--count]}} {{[-h|--no-filename]}}
"{{search_pattern}}"
```

- Print the file names and the number of matches for each file only:

```
ack {{[-c|--count]}} {{[-l|--files-with-matches]}}
"{{search_pattern}}"
```

- List all the values that can be used with `--type`:

```
ack --help-types
```

acme.sh --dns

Use a DNS-01 challenge to issue a TLS certificate.

More information: https://github.com/acmesh-official/acme.sh/wiki.

- Issue a certificate using an automatic DNS API mode:

```
acme.sh --issue --dns {{gnd_gd}} --domain {{example.com}}
```

- Issue a wildcard certificate (denoted by an asterisk) using an automatic DNS API mode:

```
acme.sh --issue --dns {{dns_namesilo}} --domain
{{example.com}} --domain {{*.example.com}}
```

- Issue a certificate using a DNS alias mode:

```
acme.sh --issue --dns {{dns_cf}} --domain {{example.com}} --
challenge-alias {{alias-for-example-validation.com}}
```

- Issue a certificate while disabling automatic Cloudflare/Google DNS polling after the DNS record is added by specifying a custom wait time in seconds:

```
acme.sh --issue --dns {{dns_namecheap}} --domain
{{example.com}} --dnssleep {{300}}
```

- Issue a certificate using a manual DNS mode:

```
acme.sh --issue --dns --domain {{example.com}} --yes-I-know-
dns-manual-mode-enough-go-ahead-please
```

acme.sh

Shell script implementing ACME client protocol, an alternative to **`certbot`**.

See also: **`acme.sh dns`**.

More information: https://github.com/acmesh-official/acme.sh.

- Issue a certificate using webroot mode:

```
acme.sh --issue {{[-d|--domain]}} {{example.com}} {{[-w|--webroot]}} /{{path/to/webroot}}
```

- Issue a certificate for multiple domains using standalone mode using port 80:

```
acme.sh --issue --standalone {{[-d|--domain]}} {{example.com}} {{[-d|--domain]}} {{www.example.com}}
```

- Issue a certificate using standalone TLS mode using port 443:

```
acme.sh --issue --alpn {{[-d|--domain]}} {{example.com}}
```

- Issue a certificate using a working Nginx configuration:

```
acme.sh --issue --nginx {{[-d|--domain]}} {{example.com}}
```

- Issue a certificate using a working Apache configuration:

```
acme.sh --issue --apache {{[-d|--domain]}} {{example.com}}
```

- Issue a wildcard (*) certificate using an automatic DNS API mode:

```
acme.sh --issue --dns {{dns_cf}} {{[-d|--domain]}} {{*.example.com}}
```

- Install certificate files into the specified locations (useful for automatic certificate renewal):

```
acme.sh {{[-i|--install-cert]}} {{[-d|--domain]}} {{example.com}} --key-file /{{path/to/example.com.key}} --fullchain-file /{{path/to/example.com.cer}} --reloadcmd "{{systemctl force-reload nginx}}"
```

act

Execute GitHub Actions locally using Docker.

More information: https://manned.org/act.

- List the available jobs:

```
act {{[-l|--list]}}
```

- Run the default event:

```
act
```

- Run a specific event:

```
act {{event_type}}
```

- Run a specific job:

```
act {{[-j|--job]}} {{job_id}}
```

- Do [n]ot actually run the actions (i.e. a dry run):

```
act {{[-n|--dryrun]}}
```

- Show verbose logs:

```
act {{[-v|--verbose]}}
```

- Run a specific workflow with the push event:

```
act push {{[-W|--workflows]}} {{path/to/workflow}}
```

acyclic

Make a directed graph acyclic by reversing some edges.

Graphviz filters: **`acyclic`**, **`bcomps`**, **`comps`**, **`edgepaint`**, **`gvcolor`**, **`gvpack`**, **`mingle`**, **`nop`**, **`sccmap`**, **`tred`**, & **`unflatten`**.

More information: https://graphviz.org/pdf/acyclic.1.pdf.

- Make a directed graph acyclic by reversing some edges:

```
acyclic {{path/to/input.gv}} > {{path/to/output.gv}}
```

- Print if a graph is acyclic, has a cycle, or is undirected, producing no output graph:

```
acyclic -v -n {{path/to/input.gv}}
```

- Display help:

```
acyclic -?
```

adb connect

Connect to an Android device wirelessly.

More information: https://developer.android.com/tools/adb.

- Pair with an Android device (address and pairing code can be found in developer options):

```
adb pair {{ip_address}}:{{port}}
```

- Connect to an Android device (port will be different from pairing):

```
adb connect {{ip_address}}:{{port}}
```

- Disconnect a device:

```
adb disconnect {{ip_address}}:{{port}}
```

adb devices

List connected Android devices.

More information: https://manned.org/adb.

- List devices:

```
adb devices
```

- List devices and their system info:

```
adb devices -l
```

adb disconnect

This command has been moved to **`adb connect`**.

- View documentation for `adb disconnect`:

  ```
  tldr adb connect
  ```

adb forward

Forward socket connections to a connected Android device or emulator.

More information: https://developer.android.com/tools/adb.

- Forward a TCP port to the only connected emulator or device:

```
adb forward tcp:{{local_port}} tcp:{{remote_port}}
```

- Forward a TCP port to a specific emulator or device (by device ID / [s]erial number):

```
adb -s {{device_ID}} forward tcp:{{local_port}} tcp:{{remote_port}}
```

- List all forwardings:

```
adb forward --list
```

- Remove a forwarding rule:

```
adb forward --remove tcp:{{local_port}}
```

- Remove all forwarding rules:

```
adb forward --remove-all
```

adb install

Push packages to a connected Android device or emulator.

More information: https://developer.android.com/tools/adb.

- Push an Android application to an emulator/device:

```
adb install {{path/to/file.apk}}
```

- Push an Android application to a specific emulator/device (overrides `$ANDROID_SERIAL`):

```
adb -s {{serial_number}} install {{path/to/file.apk}}
```

- [r]einstall an existing app, keeping its data:

```
adb install -r {{path/to/file.apk}}
```

- Push an Android application allowing version code [d]owngrade (debuggable packages only):

```
adb install -d {{path/to/file.apk}}
```

- [g]rant all permissions listed in the app manifest:

```
adb install -g {{path/to/file.apk}}
```

- Quickly update an installed package by only updating the parts of the APK that changed:

```
adb install --fastdeploy {{path/to/file.apk}}
```

adb kill-server

Stop the Android Debug Bridge (adb) server, disconnecting devices and emulators.

More information: https://manned.org/adb#head14.

- Kill the adb server if it is running:

```
adb kill-server
```

adb logcat

Dump a log of system messages.

More information: https://developer.android.com/tools/logcat.

- Display system logs:

```
adb logcat
```

- Display lines that match a `reg[e]x`:

```
adb logcat -e {{regex}}
```

- Display logs for a tag in a specific mode ([V]erbose, [D]ebug, [I]nfo, [W]arning, [E]rror, [F]atal, [S]ilent), filtering other tags:

```
adb logcat {{tag}}:{{mode}} *:S
```

- Display logs for React Native applications in [V]erbose mode [S]ilencing other tags:

```
adb logcat ReactNative:V ReactNativeJS:V *:S
```

- Display logs for all tags with priority level [W]arning and higher:

```
adb logcat *:W
```

- Display logs for a specific PID:

```
adb logcat --pid {{pid}}
```

- Display logs for the process of a specific package:

```
adb logcat --pid $(adb shell pidof -s {{package}})
```

- Color the log (usually use with filters):

```
adb logcat -v color
```

adb pair

This command has been moved to **`adb connect`**.

- View documentation for `adb pair`:

  ```
  tldr adb connect
  ```

adb reboot

Reboot a connected Android device or emulator.

More information: https://manned.org/adb.

- Reboot the device normally:

```
adb reboot
```

- Reboot the device into bootloader mode:

```
adb reboot bootloader
```

- Reboot the device into recovery mode:

```
adb reboot recovery
```

- Reboot the device into fastboot mode:

```
adb reboot fastboot
```

adb reverse

Reverse socket connections from a connected Android device or emulator.

More information: https://developer.android.com/tools/adb.

- List all reverse socket connections from emulators and devices:

```
adb reverse --list
```

- Reverse a TCP port from the only connected emulator or device to localhost:

```
adb reverse tcp:{{remote_port}} tcp:{{local_port}}
```

- Reverse a TCP port from a specific emulator or device (by device ID / [s]erial number) to localhost:

```
adb -s {{device_ID}} adb reverse tcp:{{remote_port}} tcp:{{local_port}}
```

- Remove a reverse socket connections from an emulator or device:

```
adb reverse --remove tcp:{{remote_port}}
```

- Remove all reverse socket connections from all emulators and devices:

```
adb reverse --remove-all
```

adb shell pm list packages

List installed, known, or filtered packages on an Android device.

More information: https://developer.android.com/tools/adb.

- List all installed packages:

```
adb shell pm list packages
```

- List all packages and their associated APK file paths:

```
adb shell pm list packages -f
```

- Only list disabled packages:

```
adb shell pm list packages -d
```

- Only list enabled packages:

```
adb shell pm list packages -e
```

- Only list system packages:

```
adb shell pm list packages -s
```

- Only list third-party (non-system) packages:

```
adb shell pm list packages -3
```

- Show the installer for each package:

```
adb shell pm list packages -i
```

adb shell pm list

List users, packages, permissions, instrumentation, permission groups, features and libraries managed by the package manager.

More information: https://developer.android.com/tools/adb.

- List all installed packages:

```
adb shell pm list packages
```

- Print all users on the system:

```
adb shell pm list users
```

- Print all known permission groups:

```
adb shell pm list permission-groups
```

- Print all known permissions:

```
adb shell pm list permissions
```

- List all test packages:

```
adb shell pm list instrumentation
```

- Print all features of the system:

```
adb shell pm list features
```

- Print all the libraries supported by the current device:

```
adb shell pm list libraries
```

adb shell pm

Android Package Manager tool.

More information: https://developer.android.com/tools/adb.

- List installed packages:

```
adb shell pm list packages
```

- Install an app package from a given path:

```
adb shell pm install /{{path/to/app.apk}}
```

- Uninstall a package from the device:

```
adb shell pm uninstall {{package}}
```

- Clear all app data for a package:

```
adb shell pm clear {{package}}
```

- Enable a package or component:

```
adb shell pm enable {{package_or_class}}
```

- Disable a package or component:

```
adb shell pm disable-user {{package_or_class}}
```

- Grant a permission for an app:

```
adb shell pm grant {{package}} {{android.permission.CAMERA|
android.permission.ACCESS_FINE_LOCATION|
android.permission.READ_CONTACTS|...}}
```

- Revoke a permission for an app:

```
adb shell pm revoke {{package}} {{android.permission.CAMERA|
android.permission.ACCESS_FINE_LOCATION|
android.permission.READ_CONTACTS|...}}
```

adb shell

Run shell commands on a connected Android device or emulator.

More information: https://developer.android.com/tools/adb.

- Start a remote interactive shell on the emulator or device:

```
adb shell
```

- Get all the properties from emulator or device:

```
adb shell getprop
```

- Revert all runtime permissions to their default:

```
adb shell pm reset-permissions
```

- Revoke a dangerous permission for an application:

```
adb shell pm revoke {{package}} {{permission}}
```

- Trigger a key event:

```
adb shell input keyevent {{keycode}}
```

- Clear the data of an application on an emulator or device:

```
adb shell pm clear {{package}}
```

- Start an activity on emulator or device:

```
adb shell am start -n {{package}}/{{activity}}
```

- Start the home activity on an emulator or device:

```
adb shell am start -W -c android.intent.category.HOME -a
android.intent.action.MAIN
```

adb start-server

Start the Android Debug Bridge (adb) server to allow adb connections from devices or emulators.

More information: https://manned.org/adb#head14.

- Start the adb server if it is not running:

```
adb start-server
```

adb uninstall

Uninstall a package.

More information: https://manned.org/adb.

- Uninstall a package:

```
adb uninstall {{com.example.app}}
```

- Uninstall a package, but keep user data:

```
adb uninstall -k {{com.example.app}}
```

adb

Android Debug Bridge: communicate with an Android emulator instance or connected Android devices.

Some subcommands such as **`shell`** have their own usage documentation.

More information: https://developer.android.com/tools/adb.

- Check whether the adb server process is running and start it:

```
adb start-server
```

- Terminate the adb server process:

```
adb kill-server
```

- Start a remote shell in the target emulator/device instance:

```
adb shell
```

- Push an Android application to an emulator/device:

```
adb install -r {{path/to/file.apk}}
```

- Copy a file/directory from the target device:

```
adb pull {{path/to/device_file_or_directory}} {{path/to/local_destination_directory}}
```

- Copy a file/directory to the target device:

```
adb push {{path/to/local_file_or_directory}} {{path/to/device_destination_directory}}
```

- List all connected devices:

```
adb devices
```

- Specify which device to send commands to if there are multiple devices:

```
adb -s {{device_ID}} {{shell}}
```

AdGuardHome

A network-wide software for blocking ads & tracking.

More information: https://github.com/AdguardTeam/AdGuardHome.

- Run AdGuard Home:

  ```
  AdGuardHome
  ```

- Specify a configuration file:

  ```
  AdGuardHome --config {{path/to/AdGuardHome.yaml}}
  ```

- Store the data in a specific work directory:

  ```
  AdGuardHome --work-dir {{path/to/directory}}
  ```

- Install or uninstall AdGuard Home as a service:

  ```
  AdGuardHome --service {{install|uninstall}}
  ```

- Start the AdGuard Home service:

  ```
  AdGuardHome --service start
  ```

- Reload the configuration for the AdGuard Home service:

  ```
  AdGuardHome --service reload
  ```

- Stop or restart the AdGuard Home service:

  ```
  AdGuardHome --service {{stop|restart}}
  ```

adscript

Compiler for Adscript files.

More information: https://github.com/Amplus2/Adscript.

- Compile a file to an object file:

```
adscript --output {{path/to/file.o}} {{path/to/input_file.adscript}}
```

- Compile and link a file to a standalone executable:

```
adscript --executable --output {{path/to/file}} {{path/to/input_file.adscript}}
```

- Compile a file to LLVM IR instead of native machine code:

```
adscript --llvm-ir --output {{path/to/file.ll}} {{path/to/input_file.adscript}}
```

- Cross-compile a file to an object file for a foreign CPU architecture or operating system:

```
adscript --target-triple {{i386-linux-elf}} --output {{path/to/file.o}} {{path/to/input_file.adscript}}
```

afconvert

Convert between AFF and raw file formats.

More information: https://manned.org/afconvert.1.

- Use a specific extension (default: `aff`):

```
afconvert -a {{extension}} {{path/to/input_file}} {{path/to/
output_file1 path/to/output_file2 ...}}
```

- Use a specific compression level (default: 7):

```
afconvert -X{{0..7}} {{path/to/input_file}} {{path/to/
output_file1 path/to/output_file2 ...}}
```

ag

The Silver Searcher. Like **ack**, but aims to be faster.

More information: https://manned.org/ag.

- Find files containing `string`, and print the line matches in context:

```
ag string
```

- Find files containing `string` in a specific directory:

```
ag string {{path/to/directory}}
```

- Find files containing `string`, but only list the filenames:

```
ag {{[-l|--files-with-matches]}} string
```

- Find files containing `STRING` case-insensitively, and print only the match, rather than the whole line:

```
ag {{[-i|--ignore-case]}} {{[-o|--only-matching]}} STRING
```

- Find `string` in files with a name matching `file_name`:

```
ag string {{[-G|--file-search-regex]}} file_name
```

- Find files whose contents match a `regex`:

```
ag '{{^ca(t|r)$}}'
```

- Find files with a name matching `string`:

```
ag {{[-g|--filename-pattern]}} string
```

agate

A simple server for the Gemini network protocol.

More information: https://github.com/mbrubeck/agate.

- Run and generate a private key and certificate:

```
agate --content {{path/to/content}}/ --addr {{[::]:1965}} --addr {{0.0.0.0:1965}} --hostname {{example.com}} --lang {{en-US}}
```

- Run server:

```
agate {{path/to/file}}
```

- Display help:

```
agate {{[-h|--help]}}
```

age-keygen

Generate **age** key pairs.

See also: **age** for encrypting/decrypting files.

More information: https://manned.org/age-keygen.

- Generate a key pair, save it to an unencrypted file, and print the public key to `stdout`:

```
age-keygen {{[-o|--output]}} {{path/to/file}}
```

- Convert an identit[y] to a recipient and print the public key to `stdout`:

```
age-keygen -y {{path/to/file}}
```

age

A simple, modern and secure file encryption tool.

See also: **age-keygen** for generating key pairs.

More information: https://github.com/FiloSottile/age.

- Generate an encrypted file that can be decrypted with a passphrase:

```
age --passphrase --output {{path/to/encrypted_file}} {{path/to/unencrypted_file}}
```

- Encrypt a file with one or more public keys entered as literals (repeat the `--recipient` flag to specify multiple public keys):

```
age --recipient {{public_key}} --output {{path/to/encrypted_file}} {{path/to/unencrypted_file}}
```

- Encrypt a file to one or more recipients with their public keys specified in a file (one per line):

```
age --recipients-file {{path/to/recipients_file}} --output {{path/to/encrypted_file}} {{path/to/unencrypted_file}}
```

- Decrypt a file with a passphrase:

```
age --decrypt --output {{path/to/decrypted_file}} {{path/to/encrypted_file}}
```

- Decrypt a file with a private key file:

```
age --decrypt --identity {{path/to/private_key_file}} --output {{path/to/decrypted_file}} {{path/to/encrypted_file}}
```

agg

Create a GIF from an **asciinema** terminal session recording.

More information: https://docs.asciinema.org/manual/agg/usage/.

- Create a GIF:

```
agg {{path/to/demo.cast}} {{path/to/demo.gif}}
```

- Create a GIF that is 80 columns wide and 25 rows in height:

```
agg --cols 80 --rows 25 {{path/to/demo.cast}} {{path/to/demo.gif}}
```

- Create a GIF with a font size of 24 pixels:

```
agg --font-size 24 {{path/to/demo.cast}} {{path/to/demo.gif}}
```

- Display help:

```
agg {{[-h|--help]}}
```

aider

Pair program with the LLM of your choice.

More information: https://github.com/Aider-AI/aider.

- Start a new project or work with an existing code base:

```
aider --model {{model_name}} --api-key {{your_api_key}}
```

- Add new features or test cases to specific files:

```
aider {{path/to/file1 path/to/file2 ...}}
```

- Describe a bug and let `aider` fix it:

```
aider {{path/to/file}} --describe "{{bug_description}}"
```

- Refactor code in a specific file:

```
aider {{path/to/file}} --refactor
```

- Update documentation:

```
aider {{path/to/file}} --update-docs
```

- Display help:

```
aider --help
```

aircrack-ng

Crack WEP and WPA/WPA2 keys from handshake in captured packets.

Part of Aircrack-ng network software suite.

More information: https://www.aircrack-ng.org/doku.php?id=aircrack-ng.

- Crack key from capture file using [w]ordlist:

```
aircrack-ng -w {{path/to/wordlist.txt}} {{path/to/
capture.cap}}
```

- Crack key using multiple CPU threads from capture file using [w]ordlist:

```
aircrack-ng -p {{number}} -w {{path/to/wordlist.txt}} {{path/
to/capture.cap}}
```

- Crack key from capture file using [w]ordlist and the access point's [e]ssid:

```
aircrack-ng -w {{path/to/wordlist.txt}} -e {{essid}} {{path/
to/capture.cap}}
```

- Crack key from capture file using [w]ordlist and the access point's MAC address:

```
aircrack-ng -w {{path/to/wordlist.txt}} --bssid {{mac}}
{{path/to/capture.cap}}
```

airdecap-ng

Decrypt a WEP, WPA, or WPA2 encrypted capture file.

Part of Aircrack-ng network software suite.

More information: https://www.aircrack-ng.org/doku.php?id=airdecap-ng.

- Remove wireless headers from an open network capture file and use the access point's MAC address to filter:

```
airdecap-ng -b {{ap_mac}} {{path/to/capture.cap}}
```

- Decrypt a [w]EP encrypted capture file using the key in hex format:

```
airdecap-ng -w {{hex_key}} {{path/to/capture.cap}}
```

- Decrypt a WPA/WPA2 encrypted capture file using the access point's [e]ssid and [p]assword:

```
airdecap-ng -e {{essid}} -p {{password}} {{path/to/
capture.cap}}
```

- Decrypt a WPA/WPA2 encrypted capture file preserving the headers using the access point's [e]ssid and [p]assword:

```
airdecap-ng -l -e {{essid}} -p {{password}} {{path/to/
capture.cap}}
```

- Decrypt a WPA/WPA2 encrypted capture file using the access point's [e]ssid and [p]assword and use its MAC address to filter:

```
airdecap-ng -b {{ap_mac}} -e {{essid}} -p {{password}}
{{path/to/capture.cap}}
```

aireplay-ng

Inject packets into a wireless network.

Part of Aircrack-ng network software suite.

More information: https://www.aircrack-ng.org/doku.php?id=aireplay-ng.

- Send a specific number of disassociate packets given an access point's MAC address, a client's MAC address and an interface:

```
sudo aireplay-ng --deauth {{count}} --bssid {{ap_mac}} --dmac {{client_mac}} {{interface}}
```

airmon-ng

Activate monitor mode on wireless network devices.

Part of Aircrack-ng network software suite.

More information: https://www.aircrack-ng.org/doku.php?id=airmon-ng.

- List wireless devices and their statuses:

```
sudo airmon-ng
```

- Turn on monitor mode for a specific device:

```
sudo airmon-ng start {{wlan0}}
```

- Kill disturbing processes that use wireless devices:

```
sudo airmon-ng check kill
```

- Turn off monitor mode for a specific network interface:

```
sudo airmon-ng stop {{wlan0mon}}
```

airodump-ng

Capture packets and display information about wireless networks.

Part of Aircrack-ng network software suite.

More information: https://www.aircrack-ng.org/doku.php?id=airodump-ng.

- Capture packets and display information about wireless network(s) on the 2.4GHz band:

```
sudo airodump-ng {{interface}}
```

- Capture packets and display information about wireless network(s) on the 5GHz band:

```
sudo airodump-ng {{interface}} --band a
```

- Capture packets and display information about wireless network(s) on both 2.4GHz and 5GHz bands:

```
sudo airodump-ng {{interface}} --band abg
```

- Capture packets and display information about a wireless network given the MAC address and channel, and save the output to a file:

```
sudo airodump-ng --channel {{channel}} --write {{path/to/file}} --bssid {{mac}} {{interface}}
```

airpaste

Share messages and files on the same network using mDNS.

More information: https://github.com/mafintosh/airpaste.

- Wait for a message and display it when received:

```
airpaste
```

- Send text:

```
echo {{text}} | airpaste
```

- Send a file:

```
airpaste < {{path/to/file}}
```

- Receive a file:

```
airpaste > {{path/to/file}}
```

- Create or join a channel:

```
airpaste {{channel_name}}
```

airshare

Transfer data between two machines in a local network.

More information: https://airshare.rtfd.io/en/latest/cli.html.

- Share files or directories:

```
airshare {{code}} {{path/to/file_or_directory1 path/to/file_or_directory2 ...}}
```

- Receive a file:

```
airshare {{code}}
```

- Host a receiving server (use this to be able to upload files using the web interface):

```
airshare --upload {{code}}
```

- Send files or directories to a receiving server:

```
airshare --upload {{code}} {{path/to/file_or_directory1 path/to/file_or_directory2 ...}}
```

- Send files whose paths have been copied to the clipboard:

```
airshare --file-path {{code}}
```

- Receive a file and copy it to the clipboard:

```
airshare --clip-receive {{code}}
```

ajson

Execute JSONPath on JSON objects.

More information: https://github.com/spyzhov/ajson.

- Read JSON from a file and execute a specified JSONPath expression:

```
ajson '{{$..json[?(@.path)]}}' {{path/to/file.json}}
```

- Read JSON from `stdin` and execute a specified JSONPath expression:

```
cat {{path/to/file.json}} | ajson '{{$..json[?(@.path)]}}'
```

- Read JSON from a URL and evaluate a specified JSONPath expression:

```
ajson '{{avg($..price)}}' '{{https://example.com/api/}}'
```

- Read some simple JSON and calculate a value:

```
echo '{{3}}' | ajson '{{2 * pi * $}}'
```

alacritty

Cross-platform, GPU-accelerated terminal emulator.

More information: https://manned.org/alacritty.

- Start a new Alacritty process and create a window:

```
alacritty
```

- Start the Alacritty daemon (without creating a window):

```
alacritty --daemon
```

- Create a new window using the already running Alacritty process:

```
alacritty msg create-window
```

- Start the shell in a specific directory (also works with `alacritty msg create-window`):

```
alacritty --working-directory {{path/to/directory}}
```

- Execute a command in a new Alacritty window (also works with `alacritty msg create-window`):

```
alacritty {{[-e|--command]}} {{command}}
```

- Use an alternative configuration file (defaults to `$XDG_CONFIG_HOME/alacritty/alacritty.toml`):

```
alacritty --config-file {{path/to/config.toml}}
```

alex

Catch insensitive, inconsiderate writing.

It helps you find gender favouring, polarising, race related, religion inconsiderate, or other unequal phrasing in text.

More information: https://github.com/get-alex/alex.

- Analyze text from `stdin`:

```
echo {{His network looks good}} | alex --stdin
```

- Analyze all files in the current directory:

```
alex
```

- Analyze a specific file:

```
alex {{path/to/file.md}}
```

- Analyze all Markdown files except `example.md`:

```
alex *.md !{{example.md}}
```

alias

Create aliases - words that are replaced by a command string.

Aliases expire with the current shell session unless defined in the shell's configuration file, e.g. **`~/.bashrc`** for Bash or **`~/.zshrc`** for Zsh.

See also: **`unalias`**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-alias.

- List all aliases:

```
alias
```

- Create a generic alias:

```
alias {{word}}="{{command}}"
```

- View the command associated to a given alias:

```
alias {{word}}
```

- Remove an aliased command:

```
unalias {{word}}
```

- Turn `rm` into an interactive command:

```
alias {{rm}}="{{rm --interactive}}"
```

- Create `la` as a shortcut for `ls --all`:

```
alias {{la}}="{{ls --all}}"
```

alr

Ada package manager.

Manage Ada toolchains, dependencies, tools and libraries.

More information: https://alire.ada.dev/docs/#first-steps.

- Create a binary or library project:

```
alr init {{--bin|--lib}} {{project_name}}
```

- Add a dependency to the project:

```
alr add {{crate}}
```

- Run the compiled binary (no need to do `build` before):

```
alr run
```

- Compile the project:

```
alr build {{--release|--development|--validation}}
```

amass enum

Find subdomains of a domain.

More information: https://github.com/owasp-amass/amass/blob/master/doc/user_guide.md#the-enum-subcommand.

- Find (passively) subdomains of a [d]omain:

```
amass enum -d {{domain_name}}
```

- Find subdomains of a [d]omain and actively verify them attempting to resolve the found subdomains:

```
amass enum -active -d {{domain_name}} -p {{80,443,8080}}
```

- Do a brute force search for sub[d]omains:

```
amass enum -brute -d {{domain_name}}
```

- Save the results to a text file:

```
amass enum -o {{output_file}} -d {{domain_name}}
```

- Save terminal output to a file and other detailed output to a directory:

```
amass enum -o {{output_file}} -dir {{path/to/directory}} -d {{domain_name}}
```

- List all available data sources:

```
amass enum -list
```

amass intel

Collect open source intel on an organisation like root domains and ASNs.

More information: https://github.com/owasp-amass/amass/blob/master/doc/user_guide.md#the-intel-subcommand.

- Find root domains in an IP [addr]ess range:

```
amass intel -addr {{192.168.0.1-254}}
```

- Use active recon methods:

```
amass intel -active -addr {{192.168.0.1-254}}
```

- Find root domains related to a [d]omain:

```
amass intel -whois -d {{domain_name}}
```

- Find ASNs belonging to an [org]anisation:

```
amass intel -org {{organisation_name}}
```

- Find root domains belonging to a given Autonomous System Number:

```
amass intel -asn {{asn}}
```

- Save results to a text file:

```
amass intel -o {{output_file}} -whois -d {{domain_name}}
```

- List all available data sources:

```
amass intel -list
```

amass

In-depth Attack Surface Mapping and Asset Discovery tool.

Some subcommands such as **`intel`** have their own usage documentation.

More information: https://github.com/owasp-amass/amass.

- Execute an Amass subcommand:

```
amass {{intel|enum}} {{options}}
```

- Display help:

```
amass -help
```

- Display help on an Amass subcommand:

```
amass {{intel|enum}} -help
```

- Display version:

```
amass -version
```

androguard

Reverse engineer Android applications. Written in Python.

More information: https://github.com/androguard/androguard.

- Display Android app manifest:

```
androguard axml {{path/to/app.apk}}
```

- Display app metadata (version and app ID):

```
androguard apkid {{path/to/app.apk}}
```

- Decompile Java code from an app:

```
androguard decompile {{path/to/app.apk}} --output {{path/to/directory}}
```

ani-cli

A cli to browse and watch anime.

See also: **`animdl`**.

More information: https://manned.org/ani-cli.

- Search anime by name:

```
ani-cli "{{anime_title}}"
```

- Download an episode:

```
ani-cli {{[-d|--download]}} "{{anime_title}}"
```

- Download a range of episodes:

```
ani-cli {{[-d|--download]}} {{[-r|--range]}} "{{1 6}}"
"{{anime_title}}"
```

- Download the entire series (a range of all episodes):

```
ani-cli {{[-d|--download]}} {{[-r|--range]}} "1 -1"
"{{anime_title}}"
```

- Use VLC as the media player:

```
ani-cli {{[-v|-vlc]}} "{{anime_title}}"
```

- Watch a specific episode:

```
ani-cli {{[-e|--episode]}} {{episode_number}}
"{{anime_title}}"
```

- Continue watching anime from history:

```
ani-cli {{[-c|--continue]}}
```

- Update `ani-cli`:

```
ani-cli {{[-U|--update]}}
```

animdl

A highly efficient, powerful, and fast anime scraper.

See also: **ani-cli**.

More information: https://github.com/justfoolingaround/animdl.

- Download a specific anime:

```
animdl download {{anime_title}}
```

- Download a specific anime by specifying an episode range:

```
animdl download {{anime_title}} {{[-r|--range]}}
{{start_episode}}-{{end_episode}}
```

- Download a specific anime by specifying a download directory:

```
animdl download {{anime_title}} {{[-d|--download-dir]}}
{{path/to/download_directory}}
```

- Grab the stream URL for a specific anime:

```
animdl grab {{anime_title}}
```

- Display the upcoming anime schedule for the next week:

```
animdl schedule
```

- Search a specific anime:

```
animdl search {{anime_title}}
```

- Stream a specific anime:

```
animdl stream {{anime_title}}
```

- Stream the latest episode of a specific anime:

```
animdl stream {{anime_title}} {{[-s|--special]}} latest
```

anki

Powerful, intelligent flashcard program.

More information: https://docs.ankiweb.net.

- Launch the GUI:

```
anki
```

- Use a specific profile:

```
anki {{[-p|--profile]}} {{profile_name}}
```

- Use a specific language:

```
anki {{[-l|--lang]}} {{language}}
```

- Use a non-default directory (`~/Anki` for default):

```
anki {{[-b|--base]}} {{path/to/directory}}
```

ansible-doc

Display information on modules installed in Ansible libraries.

Display a terse listing of plugins and their short descriptions.

More information: https://docs.ansible.com/ansible/latest/cli/ansible-doc.html.

- List available action plugins (modules):

```
ansible-doc {{[-l|--list]}}
```

- List available plugins of a specific type:

```
ansible-doc {{[-t|--type]}} {{become|cache|callback|cliconf|
connection|...}} {{[-l|--list]}}
```

- Show information about a specific action plugin (module):

```
ansible-doc {{plugin_name}}
```

- Show information about a plugin with a specific type:

```
ansible-doc {{[-t|--type]}} {{become|cache|callback|cliconf|
connection|...}} {{plugin_name}}
```

- Show the playbook snippet for action plugin (modules):

```
ansible-doc {{[-s|--snippet]}} {{plugin_name}}
```

- Show information about an action plugin (module) as JSON:

```
ansible-doc {{[-j|--json]}} {{plugin_name}}
```

ansible-galaxy

Perform various Ansible Role and Collection related operations.

More information: https://docs.ansible.com/ansible/latest/cli/ansible-galaxy.html.

- List installed roles or collections:

```
ansible-galaxy {{role|collection}} list
```

- Search for a role with various levels of verbosely (`-v` should be specified at the end):

```
ansible-galaxy role search {{keyword}} -v{{vvvvv}}
```

- Install or remove role(s):

```
ansible-galaxy role {{install|remove}} {{role_name1
role_name2 ...}}
```

- Create a new role:

```
ansible-galaxy role init {{role_name}}
```

- Get information about a role:

```
ansible-galaxy role info {{role_name}}
```

- Install or remove collection(s):

```
ansible-galaxy collection {{install|remove}}
{{collection_name1 collection_name2 ...}}
```

- Display help about roles or collections:

```
ansible-galaxy {{role|collection}} {{[-h|--help]}}
```

ansible-inventory

Display or dump an Ansible inventory.

See also: **`ansible`**.

More information: https://docs.ansible.com/ansible/latest/cli/ansible-inventory.html.

- Display the default inventory:

```
ansible-inventory --list
```

- Display a custom inventory:

```
ansible-inventory --list {{[-i|--inventory-file]}} {{path/to/file_or_script_or_directory}}
```

- Display the default inventory in YAML:

```
ansible-inventory --list {{[-y|--yaml]}}
```

- Dump the default inventory to a file:

```
ansible-inventory --list --output {{path/to/file}}
```

ansible-lint

Apply rules and follow best practices with your automation content.

More information: https://ansible.readthedocs.io/projects/lint/.

- Lint a specific playbook and a role directory:

```
ansible-lint {{path/to/playbook_file}} {{path/to/
role_directory}}
```

- Lint a playbook while excluding specific rules:

```
ansible-lint {{[-x|--exclude-rules]}} {{rule1,rule2,...}}
{{path/to/playbook_file}}
```

- Lint a playbook in offline mode and format output as PEP8:

```
ansible-lint {{[-o|--offline]}} {{[-p|--parseable]}} {{path/
to/playbook_file}}
```

- Lint a playbook using a custom rules directory:

```
ansible-lint {{[-r|--rules]}} {{path/to/
custom_rules_directory}} {{path/to/playbook_file}}
```

- Lint all Ansible content recursively in a given directory:

```
ansible-lint {{path/to/project_directory}}
```

ansible-playbook

Execute tasks defined in playbook on remote machines over SSH.

More information: https://docs.ansible.com/ansible/latest/cli/ansible-playbook.html.

- Run tasks in playbook:

```
ansible-playbook {{playbook}}
```

- Run tasks in playbook with custom host inventory:

```
ansible-playbook {{playbook}} {{[-i|--inventory]}} {{inventory_file}}
```

- Run tasks in playbook with extra variables defined via the command-line:

```
ansible-playbook {{playbook}} {{[-e|--extra-vars]}} "{{variable1}}={{value1}} {{variable2}}={{value2}}"
```

- Run tasks in playbook with extra variables defined in a JSON file:

```
ansible-playbook {{playbook}} {{[-e|--extra-vars]}} "@{{variables.json}}"
```

- Run tasks in playbook for the given tags:

```
ansible-playbook {{playbook}} {{[-t|--tags]}} {{tag1,tag2}}
```

- Run tasks in a playbook starting at a specific task:

```
ansible-playbook {{playbook}} --start-at {{task_name}}
```

- Run tasks in a playbook without making any changes (dry-run):

```
ansible-playbook {{playbook}} {{[-C|--check]}} {{[-D|--diff]}}
```

ansible-pull

Pull ansible playbooks from a VCS repo and executes them for the local host.

More information: https://docs.ansible.com/ansible/latest/cli/ansible-pull.html.

- Pull a playbook from a VCS and execute a default local.yml playbook:

```
ansible-pull {{[-U|--url]}} {{repository_url}}
```

- Pull a playbook from a VCS and execute a specific playbook:

```
ansible-pull {{[-U|--url]}} {{repository_url}} {{playbook}}
```

- Pull a playbook from a VCS at a specific branch and execute a specific playbook:

```
ansible-pull {{[-U|--url]}} {{repository_url}} {{[-C|--checkout]}} {{branch}} {{playbook}}
```

- Pull a playbook from a VCS, specify hosts file and execute a specific playbook:

```
ansible-pull {{[-U|--url]}} {{repository_url}} {{[-i|--inventory-file]}} {{hosts_file}} {{playbook}}
```

ansible-vault

Encrypt and decrypt values, data structures and files within Ansible projects.

More information: https://docs.ansible.com/ansible/latest/user_guide/vault.html#id17.

- Create a new encrypted vault file with a prompt for a password:

```
ansible-vault create {{vault_file}}
```

- Create a new encrypted vault file using a vault key file to encrypt it:

```
ansible-vault create --vault-password-file {{password_file}} {{vault_file}}
```

- Encrypt an existing file using an optional password file:

```
ansible-vault encrypt --vault-password-file {{password_file}} {{vault_file}}
```

- Encrypt a string using Ansible's encrypted string format, displaying interactive prompts:

```
ansible-vault encrypt_string
```

- View an encrypted file, using a password file to decrypt:

```
ansible-vault view --vault-password-file {{password_file}} {{vault_file}}
```

- Re-key already encrypted vault file with a new password file:

```
ansible-vault rekey --vault-password-file {{old_password_file}} --new-vault-password-file {{new_password_file}} {{vault_file}}
```

ansible

Manage groups of computers remotely over SSH. (use the **`/etc/ansible/hosts`** file to add new groups/hosts).

Some subcommands such as **`galaxy`** have their own usage documentation.

More information: https://docs.ansible.com/ansible/latest/cli/ansible.html.

- List hosts belonging to a group:

```
ansible {{group}} --list-hosts
```

- Ping a group of hosts by invoking the ping module:

```
ansible {{group}} {{[-m|--module-name]}} ping
```

- Display facts about a group of hosts by invoking the setup module:

```
ansible {{group}} {{[-m|--module-name]}} setup
```

- Execute a command on a group of hosts by invoking command module with arguments:

```
ansible {{group}} {{[-m|--module-name]}} command {{[-a|--
args]}} '{{my_command}}'
```

- Execute a command with administrative privileges:

```
ansible {{group}} {{[-b|--become]}} --ask-become-pass {{[-
m|--module-name]}} command {{[-a|--args]}} '{{my_command}}'
```

- Execute a command using a custom inventory file:

```
ansible {{group}} {{[-i|--inventory]}} {{inventory_file}}
{{[-m|--module-name]}} command {{[-a|--args]}}
'{{my_command}}'
```

- List the groups in an inventory:

```
ansible localhost {{[-m|--module-name]}} debug {{[-a|--
args]}} '{{var=groups.keys()}}'
```

ansiweather

Display the current weather conditions in your terminal.

More information: https://github.com/fcambus/ansiweather.

- Display a [f]orecast for the next seven days using metric [u]nits for a specific [l]ocation:

```
ansiweather -u metric -f 7 -l {{Rzeszow,PL}}
```

- Display a [F]orecast for the next five days showing [s]ymbols and [d]aylight data for your current location:

```
ansiweather -F -s true -d true
```

- Display today's [w]ind and [h]umidity data for your current location:

```
ansiweather -w true -h true
```

ant

Apache Ant: build and manage Java-based projects.

More information: https://ant.apache.org/manual/index.html.

- Build a project with default build file `build.xml`:

  `ant`

- Build a project using build file other than `build.xml`:

  `ant {{[-f|-buildfile]}} {{buildfile.xml}}`

- Print information on possible targets for this project:

  `ant {{[-p|-projecthelp]}}`

- Print debugging information:

  `ant {{[-d|-debug]}}`

- Execute all targets that do not depend on fail target(s):

  `ant {{[-k|-keep-going]}}`

antibody

"The fastest" shell plugin manager.

More information: https://getantibody.github.io.

- Bundle all plugins for static loading:

```
antibody < {{~/.zsh_plugins.txt}} bundle >
{{~/.zsh_plugins.sh}}
```

- Update all bundles:

```
antibody update
```

- List all installed plugins:

```
antibody list
```

anytopnm

Convert an arbitrary type of image file to common image formats.

More information: https://netpbm.sourceforge.net/doc/anytopnm.html.

- Convert an input image to PBM, PGM, or PPM format irrespective of the input type:

```
anytopnm {{path/to/input}} > {{path/to/output.pnm}}
```

- Display version:

```
anytopnm {{[-v|-version]}}
```

apg

Create arbitrarily complex random passwords.

More information: https://manned.org/apg.

- Create random passwords (default password length is 8):

```
apg
```

- Create a password with at least 1 symbol (S), 1 number (N), 1 uppercase (C), 1 lowercase (L):

```
apg -M SNCL
```

- Create a password with 16 characters:

```
apg -m {{16}}
```

- Create a password with maximum length of 16:

```
apg -x {{16}}
```

- Create a password that doesn't appear in a dictionary (the dictionary file has to be provided):

```
apg -r {{path/to/dictionary_file}}
```

apkeep

Download APK files from various sources.

More information: https://github.com/EFForg/apkeep/blob/master/USAGE.

- Download an APK file to the specified directory:

```
apkeep {{[-a|--app]}} {{com.example.application}} {{path/to/directory}}
```

- List all available versions for download:

```
apkeep {{[-a|--app]}} {{com.example.application}} {{[-l|--list-versions]}} {{path/to/directory}}
```

- Specify a store to download from:

```
apkeep {{[-a|--app]}} {{com.example.application}} {{[-d|--download-source]}} {{apk-pure|google-play|f-droid|huawei-app-gallery}} {{path/to/directory}}
```

apkleaks

Expose URIs, endpoints, and secrets from APK files.

Note: APKLeaks utilizes the **`jadx`** disassembler to decompile APK files.

More information: https://github.com/dwisiswant0/apkleaks.

- Scan an APK file for URIs, endpoints, and secrets:

```
apkleaks {{[-f|--file]}} {{path/to/file.apk}}
```

- Scan and save the output to a specific file:

```
apkleaks {{[-f|--file]}} {{path/to/file.apk}} {{[-o|--
output]}} {{path/to/output.txt}}
```

- Pass `jadx` disassembler arguments:

```
apkleaks {{[-f|--file]}} {{path/to/file.apk}} {{[-a|--args]}}
"{{--threads-count 5 --deobf}}"
```

apktool

Reverse engineer APK files.

More information: https://ibotpeaches.github.io/Apktool/.

- Decode an APK file:

```
apktool d {{path/to/file.apk}}
```

- Build an APK file from a directory:

```
apktool b {{path/to/directory}}
```

- Install and store a framework:

```
apktool if {{path/to/framework.apk}}
```

apm

Atom editor Package Manager.

See also: **`atom`**.

More information: https://github.com/atom/apm.

- Install a package from http://atom.io/packages or a theme from http://atom.io/themes:

```
apm install {{package}}
```

- Remove a package/theme:

```
apm remove {{package}}
```

- Upgrade a package/theme:

```
apm upgrade {{package}}
```

apropos

Search the manual pages for names and descriptions.

More information: https://manned.org/apropos.

- Search for a keyword using a regex:

```
apropos {{regex}}
```

- Search without restricting the output to the terminal width (long output):

```
apropos {{[-l|--long]}} {{regex}}
```

- Search for pages that match all the regex given:

```
apropos {{regex_1}} {{[-a|--and]}} {{regex_2}} {{[-a|--and]}} {{regex_3}}
```

ar

Create, modify, and extract from Unix archives. Typically used for static libraries (**`.a`**) and Debian packages (**`.deb`**).

See also: **`tar`**.

More information: https://manned.org/ar.

- E[x]tract all members from an archive:

```
ar x {{path/to/file.a}}
```

- Lis[t] contents in a specific archive:

```
ar t {{path/to/file.ar}}
```

- [r]eplace or add specific files to an archive:

```
ar r {{path/to/file.deb}} {{path/to/debian-binary path/to/
control.tar.gz path/to/data.tar.xz ...}}
```

- In[s]ert an object file index (equivalent to using `ranlib`):

```
ar s {{path/to/file.a}}
```

- Create an archive with specific files and an accompanying object file index:

```
ar rs {{path/to/file.a}} {{path/to/file1.o path/to/
file2.o ...}}
```

arc

Arcanist: a CLI for Phabricator.

More information: https://secure.phabricator.com/book/phabricator/article/arcanist/.

- Send the changes to Differential for review:

```
arc diff
```

- Show pending revision information:

```
arc list
```

- Update Git commit messages after review:

```
arc amend
```

- Push Git changes:

```
arc land
```

arch

Display the name of the system architecture.

See also: **uname**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/arch-invocation.html.

- Display the system's architecture:

```
arch
```

archwiki-rs

Read, search and download pages from the ArchWiki.

More information: https://gitlab.com/lucifayr/archwiki-rs.

- Read a page from the ArchWiki:

```
archwiki-rs read-page {{page_title}}
```

- Read a page from the ArchWiki in the specified format:

```
archwiki-rs read-page {{page_title}} --format {{plain-text|markdown|html}}
```

- Search the ArchWiki for pages containing the provided text:

```
archwiki-rs search "{{search_text}}" --text-search
```

- Download a local copy of all ArchWiki pages into a specific directory:

```
archwiki-rs local-wiki /{{path/to/local_wiki}} --format {{plain-text|markdown|html}}
```

arduino-builder

Compile arduino sketches.

DEPRECATION WARNING: this tool is being phased out in favor of **`arduino`**.

More information: https://github.com/arduino/arduino-builder.

- Compile a sketch:

  ```
  arduino-builder -compile {{path/to/sketch.ino}}
  ```

- Specify the debug level (default: 5):

  ```
  arduino-builder -debug-level {{1..10}}
  ```

- Specify a custom build directory:

  ```
  arduino-builder -build-path {{path/to/build_directory}}
  ```

- Use a build option file, instead of specifying `-hardware`, `-tools`, etc. manually every time:

  ```
  arduino-builder -build-options-file {{path/to/
  build.options.json}}
  ```

- Enable verbose mode:

  ```
  arduino-builder -verbose {{true}}
  ```

arduino

Arduino Studio - Integrated Development Environment for the Arduino platform.

More information: https://github.com/arduino/Arduino/blob/master/build/shared/manpage.adoc.

- Build a sketch:

```
arduino --verify {{path/to/file.ino}}
```

- Build and upload a sketch:

```
arduino --upload {{path/to/file.ino}}
```

- Build and upload a sketch to an Arduino Nano with an Atmega328p CPU, connected on port `/dev/ttyACM0`:

```
arduino --board {{arduino:avr:nano:cpu=atmega328p}} --port {{/dev/ttyACM0}} --upload {{path/to/file.ino}}
```

- Set the preference `name` to a given `value`:

```
arduino --pref {{name}}={{value}}
```

- Build a sketch, put the build results in the build directory, and reuse any previous build results in that directory:

```
arduino --pref build.path={{path/to/build_directory}} --verify {{path/to/file.ino}}
```

- Save any (changed) preferences to `preferences.txt`:

```
arduino --save-prefs
```

- Install the latest SAM board:

```
arduino --install-boards "{{arduino:sam}}"
```

- Install Bridge and Servo libraries:

```
arduino --install-library "{{Bridge:1.0.0,Servo:1.2.0}}"
```

argocd app

Interface to manage applications by Argo CD.

More information: https://argo-cd.readthedocs.io/en/stable/user-guide/commands/argocd_app/.

- List applications:

```
argocd app list --output {{json|yaml|wide}}
```

- Get application details:

```
argocd app get {{app_name}} --output {{json|yaml|wide}}
```

- Deploy application internally (to the same cluster that Argo CD is running in):

```
argocd app create {{app_name}} --repo {{git_repo_url}} --path
{{path/to/repo}} --dest-server https://kubernetes.default.svc
--dest-namespace {{ns}}
```

- Delete an application:

```
argocd app delete {{app_name}}
```

- Enable application auto-sync:

```
argocd app set {{app_name}} --sync-policy auto --auto-prune
--self-heal
```

- Preview app synchronization without affecting cluster:

```
argocd app sync {{app_name}} --dry-run --prune
```

- Show application deployment history:

```
argocd app history {{app_name}} --output {{wide|id}}
```

- Rollback application to a previous deployed version by history ID (deleting unexpected resources):

```
argocd app rollback {{app_name}} {{history_id}} --prune
```

argocd

Interface to control an Argo CD server.

Some subcommands such as **app** have their own usage documentation.

More information: https://argo-cd.readthedocs.io/en/stable/user-guide/commands/argocd/.

- Login to Argo CD server:

```
argocd login --insecure --username {{user}} --password {{password}} {{argocd_server:port}}
```

- List applications:

```
argocd app list
```

argon2

Calculate Argon2 cryptographic hashes.

More information: https://github.com/P-H-C/phc-winner-argon2#command-line-utility.

- Calculate a hash with a password and a salt with the default parameters:

```
echo "{{password}}" | argon2 "{{salt_text}}"
```

- Calculate a hash with the specified algorithm:

```
echo "{{password}}" | argon2 "{{salt_text}}" -{{d|i|id}}
```

- Display the output [e]ncoded hash without additional information:

```
echo "{{password}}" | argon2 "{{salt_text}}" -e
```

- Calculate a hash with given iteration [t]imes, [m]emory usage, and [p]arallelism parameters:

```
echo "{{password}}" | argon2 "{{salt_text}}" -t {{5}} -m {{20}} -p {{7}}
```

argos-translate

An open-source offline translation library and CLI tool written in Python.

More information: https://argos-translate.readthedocs.io/en/latest/source/cli.html.

- Install translation pairs for Spanish to English translation:

```
argospm install translate-es_en
```

- Translate some text from Spanish (`es`) to English (`en`) (Note: Only two letter language codes are supported):

```
argos-translate --from-lang es --to-lang en {{un texto corto}}
```

- Translate a text file from English to Hindi:

```
cat {{path/to/file.txt}} | argos-translate --from-lang en --to-lang hi
```

- List all installed translation pairs:

```
argospm list
```

- Show translation pairs from English that are available to be installed:

```
argospm search --from-lang en
```

- Update installed language package pairs:

```
argospm update
```

- Translate from `ar` to `ru` (Note: This requires the translation pairs `translate-ar_en` and `translate-en_ru` to be installed):

```
argos-translate --from-lang ar --to-lang ru {{صورة تساوي أكثر من ألف كلمة}}
```

aria2

This command is an alias of **`aria2c`**.

- View documentation for the updated command:

```
tldr aria2c
```

aria2c

Fast download utility.

Supports HTTP(S), FTP, SFTP, BitTorrent, and Metalink.

More information: https://aria2.github.io.

- Download a specific URI to a file:

```
aria2c "{{url}}"
```

- Download a file from a URI with a specific output name:

```
aria2c --out {{path/to/file}} "{{url}}"
```

- Download multiple different files in parallel:

```
aria2c --force-sequential {{false}} "{{url1 url2 ...}}"
```

- Download the same file from different mirrors and verify the checksum of the downloaded file:

```
aria2c --checksum {{sha-256}}={{hash}} {{"url1" "url2" ...}}
```

- Download the URIs listed in a file with a specific number of parallel downloads:

```
aria2c --input-file {{path/to/file}} --max-concurrent-downloads {{number_of_downloads}}
```

- Download with multiple connections:

```
aria2c --split {{number_of_connections}} "{{url}}"
```

- FTP download with username and password:

```
aria2c --ftp-user {{username}} --ftp-passwd {{password}} "{{url}}"
```

- Limit download speed in bytes/s:

```
aria2c --max-download-limit {{speed}} "{{url}}"
```

arp-scan

Send ARP packets to hosts (specified as IP addresses or hostnames) to scan the local network.

More information: https://github.com/royhills/arp-scan.

- Scan the current local network:

```
arp-scan {{[-l|--localnet]}}
```

- Scan an IP network with a custom bitmask:

```
arp-scan {{192.168.1.1}}/{{24}}
```

- Scan an IP network within a custom range:

```
arp-scan {{127.0.0.0}}-{{127.0.0.31}}
```

- Scan an IP network with a custom net mask:

```
arp-scan {{10.0.0.0}}:{{255.255.255.0}}
```

arp

Show and manipulate your system's ARP cache.

More information: https://manned.org/arp.8.

- Show the current ARP table:

```
arp
```

- Show [a]lternative BSD style output format with on fixed columns:

```
arp -a
```

- [d]elete a specific entry:

```
sudo arp -d {{address}}
```

- [s]et up a new entry in the ARP table:

```
sudo arp -s {{address}} {{mac_address}}
```

arping

Discover and probe hosts in a network using the ARP protocol.

Useful for MAC address discovery.

More information: https://manned.org/arping.

- Ping a host by ARP request packets:

```
arping {{host_ip}}
```

- Ping a host on a specific [I]nterface:

```
arping -I {{interface}} {{host_ip}}
```

- Ping a host and [f]inish after the first reply:

```
arping -f {{host_ip}}
```

- Ping a host a specific number ([c]ount) of times:

```
arping -c {{count}} {{host_ip}}
```

- Broadcast ARP request packets to update neighbours' ARP caches ([U]nsolicited ARP mode):

```
arping -U {{ip_to_broadcast}}
```

- [D]etect duplicated IP addresses in the network by sending ARP requests with a 3 second timeout:

```
arping -D -w {{3}} {{ip_to_check}}
```

arthas-trace

Trace method invoke chain, and output the time cost for each node in the path.

See also: **`arthas`**, **`arthas-watch`**.

More information: https://arthas.aliyun.com/en/doc/trace.html.

- Trace method invoke chain:

```
trace {{class-pattern}} {{method-pattern}}
```

- Trace method invoke chains and only display invoke information longer than 10 ms:

```
trace {{class-pattern}} {{method-pattern}} '#cost > {{10}}'
```

- Trace the invoke chain of multiple classes or multiple methods:

```
trace -E {{class-pattern1}}|{{class-patter2}} {{method-pattern1}}|{{method-pattern2}}|{{method-pattern3}}
```

- Track method invoke chains, only display invoke information that exceeds 10 ms, and exit after 5 times:

```
trace {{class-pattern}} {{method-pattern}} '#cost > {{10}}' -n 5
```

arthas-watch

Method invoke data observation.

See also: **`arthas`**, **`arthas-trace`**.

More information: https://arthas.aliyun.com/en/doc/watch.html.

- Observe the first parameter and return value of method, and expand the nested attributes of the object to 4 levels:

```
watch {{class-pattern}} {{method-pattern}}
'{{{ params[0],returnObj }}}' -x 4
```

- When the value of the first parameter of the method is 5, the second parameter and return value are output, and the object is expanded 4 layers:

```
watch {{class-pattern}} {{method-pattern}}
'{{{ params[1],returnObj }}}' '{{"5".equals(params[0])}}' -x
4
```

- When the method returns or an exception occurs, observe the count property of the second parameter:

```
watch {{class-pattern}} {{method-pattern}}
'{{{ params[1].count }}}' -e -s
```

arthas

Java diagnostic tool.

See also: **`arthas-watch`**, **`arthas-trace`**.

More information: https://arthas.aliyun.com/en/.

- Start Arthas:

  ```
  java -jar {{path/to/arthas-boot.jar}}
  ```

- Reconnect Arthas (default port used by Arthas is 3658):

  ```
  telnet localhost {{port_number}}
  ```

- Exit the current Arthas client without affecting other clients. equals `exit` `logout` `q` command:

  ```
  {{exit|quit|logout|q}}
  ```

- Terminate the Arthas server, all the Arthas clients connecting to this server will be disconnected:

  ```
  stop
  ```

asar

A file archiver for the Electron platform.

More information: https://github.com/electron/asar.

- Archive a file or directory:

```
asar pack {{path/to/input_file_or_directory}} {{path/to/output_archive.asar}}
```

- Extract an archive:

```
asar extract {{path/to/archive.asar}}
```

- Extract a specific file from an archive:

```
asar extract-file {{path/to/archive.asar}} {{file}}
```

- List the contents of an archive file:

```
asar list {{path/to/archive.asar}}
```

ascii-image-converter

Convert an image into ASCII.

More information: https://github.com/TheZoraiz/ascii-image-converter#cli-usage.

- Convert an image into ASCII:

```
ascii-image-converter {{path/to/image|URL}}
```

- Colorize the output:

```
ascii-image-converter {{[-C|--color]}} {{path/to/image|URL}}
```

- Create a tresholded image using braille (if the image is barely visible, try changing the terminal font):

```
ascii-image-converter {{[-b|--braille]}} {{path/to/image|URL}}
```

- Create a dithered image using braille (if the image is barely visible, try changing the terminal font):

```
ascii-image-converter {{[-b|--braille]}} --dither {{path/to/image|URL}}
```

- Display the image with negative colors:

```
ascii-image-converter {{[-Cn|--color --negative]}} {{path/to/image|URL}}
```

- Use a wider range of characters to display an image (may improve image accuracy):

```
ascii-image-converter {{[-c|--complex]}} {{path/to/image|URL}}
```

asciidoctor

Convert AsciiDoc files to a publishable format.

More information: https://docs.asciidoctor.org.

- Convert a specific `.adoc` file to HTML (the default output format):

```
asciidoctor {{path/to/file.adoc}}
```

- Convert a specific `.adoc` file to HTML and link a CSS stylesheet:

```
asciidoctor {{[-a|--attribute]}} stylesheet={{path/to/
stylesheet.css}} {{path/to/file.adoc}}
```

- Convert a specific `.adoc` file to embeddable HTML, removing everything except the body:

```
asciidoctor {{[-e|--embedded]}} {{path/to/file.adoc}}
```

- Convert a specific `.adoc` file to a PDF using the `asciidoctor-pdf` library:

```
asciidoctor {{[-b|--backend]}} pdf {{[-r|--require]}}
asciidoctor-pdf {{path/to/file.adoc}}
```

asciinema

Record and replay terminal sessions, and optionally share them on https://asciinema.org.

See also: **`terminalizer`**.

More information: https://docs.asciinema.org/manual/cli/.

- Associate the local install of `asciinema` with an asciinema.org account:

```
asciinema {{[a|auth]}}
```

- Make a new recording and save it to a local file (finish it with `<Ctrl d>` or type `exit`):

```
asciinema {{[r|record]}} {{path/to/recording.cast}}
```

- Replay a terminal recording from a local file:

```
asciinema {{[p|play]}} {{path/to/recording.cast}}
```

- Replay a terminal recording hosted on https://asciinema.org:

```
asciinema {{[p|play]}} https://asciinema.org/a/{{cast_id}}
```

- Make a new recording, limiting any idle time to at most 2.5 seconds:

```
asciinema {{[r|record]}} {{[-i|--idle-time-limit]}} 2.5
```

- Print the full output of a locally saved recording:

```
asciinema {{[ca|cat]}} {{path/to/recording.cast}}
```

- Upload a locally saved terminal session to asciinema.org:

```
asciinema {{[u|upload]}} {{path/to/recording.cast}}
```

- Stream the current terminal on a local webpage:

```
asciinema {{[st|stream]}} --local
```

asciitopgm

Convert ASCII graphics into a PGM file.

More information: https://netpbm.sourceforge.net/doc/asciitopgm.html.

- Read ASCII data as input and produce a PGM image with pixel values that are an approximation of the "brightness" of the ASCII characters:

```
asciitopgm {{path/to/input_file}} > {{path/to/
output_file.pgm}}
```

- Display version:

```
asciitopgm {{[-v|-version]}}
```

asdf

Manage versions of different packages.

More information: https://asdf-vm.com/manage/commands.html.

- List all available plugins:

```
asdf plugin list all
```

- Install a plugin:

```
asdf plugin add {{name}}
```

- List all available versions for a package:

```
asdf list all {{name}}
```

- Install a specific version of a package:

```
asdf install {{name}} {{version}}
```

- Set global version for a package:

```
asdf set -u {{name}} {{version}}
```

- Set local version for a package:

```
asdf set {{name}} {{version}}
```

- See the current version used for a package:

```
asdf current {{name}}
```

asnmap

A Go CLI tool for mapping organization network ranges using ASN information.

Note: An API key is required from ProjectDiscovery Cloud Platform for the tool to work.

More information: https://github.com/projectdiscovery/asnmap.

- Lookup CIDR ranges for an ASN:

```
asnmap {{[-a|-asn]}} {{AS5650}} -silent
```

- Lookup CIDR ranges for an IP address:

```
asnmap {{[-i|-ip]}} {{100.19.12.21}} -silent
```

- Lookup CIDR ranges for a domain:

```
asnmap {{[-d|-domain]}} {{example.com}} -silent
```

- Lookup CIDR ranges for an organization:

```
asnmap -org {{GOOGLE}} -silent
```

- Lookup CIDR ranges from a file of targets:

```
asnmap {{[-f|-file]}} {{targets.txt}} -silent
```

- Output results in JSON format:

```
asnmap {{[-d|-domain]}} {{facebook.com}} {{[-j|-json]}} -silent
```

- Output results in CSV format:

```
asnmap {{[-a|-asn]}} {{AS394161}} {{[-c|-csv]}} -silent
```

- Update asnmap to the latest version:

```
asnmap {{[-up|-update]}}
```

aspell

Interactive spell checker.

More information: http://aspell.net/man-html/index.html.

- Spell check a single file:

```
aspell check {{path/to/file}}
```

- List misspelled words from `stdin`:

```
cat {{path/to/file}} | aspell list
```

- Show available dictionary languages:

```
aspell dicts
```

- Run `aspell` with a different language (takes two-letter ISO 639 language code):

```
aspell --lang {{cs}}
```

- List misspelled words from `stdin` and ignore words from personal word list:

```
cat {{path/to/file}} | aspell --personal {{personal-word-list.pws}} list
```

assimp

Client for the Open Asset Import Library.

Supports loading of 40+ 3D file formats, and exporting to several popular 3D formats.

More information: https://manned.org/assimp.

- List all supported import formats:

```
assimp listext
```

- List all supported export formats:

```
assimp listexport
```

- Convert a file to one of the supported output formats, using the default parameters:

```
assimp export {{input_file.stl}} {{output_file.obj}}
```

- Convert a file using custom parameters (the dox_cmd.h file in assimp's source code lists available parameters):

```
assimp export {{input_file.stl}} {{output_file.obj}} {{parameters}}
```

- Display a summary of a 3D file's contents:

```
assimp info {{path/to/file}}
```

- Display help:

```
assimp help
```

- Display help for a specific subcommand:

```
assimp {{subcommand}} --help
```

astronomer

Detect illegitimate stars from bot accounts on GitHub projects.

More information: https://github.com/Ullaakut/astronomer.

- Scan a repository:

```
astronomer {{tldr-pages/tldr-node-client}}
```

- Scan the maximum amount of stars in the repository:

```
astronomer {{tldr-pages/tldr-node-client}} --stars {{50}}
```

- Scan a repository including comparative reports:

```
astronomer {{tldr-pages/tldr-node-client}} --verbose
```

astroterm

A terminal-based star map.

More information: https://github.com/da-luce/astroterm#usage.

- Display real-time positions of stars and planets based on your current location:

```
astroterm
```

- Display constellations, use colour, and render the simulation at the given frame rate:

```
astroterm {{[-C|--constellations]}} {{[-c|--color]}} {{[-f|--fps]}} {{60}}
```

- Use unicode characters instead of the basic ASCII characters, and only render stars brighter than the given magnitude:

```
astroterm {{[-u|--unicode]}} {{[-t|--threshold]}} {{2.0}}
```

- Use a given latitude, longitude and datetime:

```
astroterm {{[-a|--latitude]}} {{90.0}} {{[-o|--longitude]}} {{-180.0}} {{[-d|--datetime]}} {{2025-08-04T12:00:00}}
```

- Use the longitude and latitude of a given city, and set the speed of the simulation to a given factor:

```
astroterm {{[-i|--city]}} {{Singapore}} {{[-s|--speed]}} {{1000.0}}
```

astyle

Source code indenter, formatter, and beautifier for the C, C++, C# and Java programming languages.

Upon running, a copy of the original file is created with an ".orig" appended to the original file name.

More information: https://astyle.sourceforge.net.

- Apply the default style of 4 spaces per indent and no formatting changes:

```
astyle {{source_file}}
```

- Apply the Java style with attached braces:

```
astyle --style=java {{path/to/file}}
```

- Apply the allman style with broken braces:

```
astyle --style=allman {{path/to/file}}
```

- Apply a custom indent using spaces. Choose between 2 and 20 spaces:

```
astyle --indent=spaces={{number_of_spaces}} {{path/to/file}}
```

- Apply a custom indent using tabs. Choose between 2 and 20 tabs:

```
astyle --indent=tab={{number_of_tabs}} {{path/to/file}}
```

at

Execute commands once at a later time.

Results will be sent to the users mail.

More information: https://manned.org/at.

- Create commands interactively and execute them in 5 minutes (press `<Ctrl d>` when done):

```
at now + 5 minutes
```

- Create commands interactively and execute them at a specific time:

```
at {{hh:mm}}
```

- Execute a command from `stdin` at 10:00 AM today:

```
echo "{{command}}" | at 1000
```

- Execute commands from a given file next Tuesday:

```
at -f {{path/to/file}} 9:30 PM Tue
```

- List all queued jobs for the current user (same as `atq`):

```
at -l
```

- View a specied job:

```
at -c {{job_number}}
```

atktopbm

Convert a Andrew Toolkit raster object to a PBM image.

See also: **pbmtoatk**.

More information: https://netpbm.sourceforge.net/doc/atktopbm.html.

- Convert a Andrew Toolkit raster object to a PBM image:

```
atktopbm {{path/to/image.atk}} > {{path/to/output.pbm}}
```

atom

A cross-platform pluggable text editor.

Plugins are managed by **`apm`**.

Note: Atom has been sunsetted and is no longer actively maintained.

More information: https://atom.io/.

- Open a file or directory:

  ```
  atom {{path/to/file_or_directory}}
  ```

- Open a file or directory in a [n]ew window:

  ```
  atom -n {{path/to/file_or_directory}}
  ```

- Open a file or directory in an existing window:

  ```
  atom --add {{path/to/file_or_directory}}
  ```

- Open Atom in safe mode (does not load any additional packages):

  ```
  atom --safe
  ```

- Prevent Atom from forking into the background, keeping Atom attached to the terminal:

  ```
  atom --foreground
  ```

- Wait for Atom window to close before returning (useful for Git commit editor):

  ```
  atom --wait
  ```

atool

A script for managing file archives of various types.

`atool` uses external archiver programs but provides a consistent command-line interface for listing, extracting, creating, and managing archives.

More information: https://manned.org/atool.

- List files in an archive:

```
atool {{[-l|--list]}} {{path/to/archive.zip}}
```

- Extract an archive (safely creates a subdirectory if needed):

```
atool {{[-x|--extract]}} {{archive.tar.gz}}
```

- Extract an archive to a specific directory:

```
atool {{[-X|--extract-to]}} {{path/to/output_directory}}
{{archive.rar}}
```

- Display a specific file's content from an archive to `stdout` (like `cat`):

```
atool {{[-c|--cat]}} {{archive.tar}} {{path/to/
file_in_archive.txt}}
```

- Create a new archive from specified files and/or directories:

```
atool {{[-a|--add]}} {{new_archive.zip}} {{path/to/file1
path/to/file2 ...}}
```

- List files in an archive and send the output through a pager:

```
atool {{[-l|--list]}} {{[-p|--pager]}}
{{large_archive.tar.bz2}}
```

- Extract multiple archives at once (each to its own subdirectory if needed):

```
atool {{[-x|--extract]}} {{[-e|--each]}} {{archive1.zip}}
{{archive2.tar.gz}} {{*.rar}}
```

- Repack an archive from one format to another (e.g., .tar.gz to .tar.7z):

```
atool {{[-r|--repack]}} {{old_archive.tar.gz}}
{{new_archive.tar.7z}}
```

atoum

A simple, modern and intuitive unit testing framework for PHP.

More information: https://atoum.readthedocs.io/en/latest/option_cli.html.

- Initialize a configuration file:

  ```
  atoum --init
  ```

- Run all tests:

  ```
  atoum
  ```

- Run tests using the specified configuration file:

  ```
  atoum {{[-c|--configuration]}} {{path/to/file}}
  ```

- Run a specific test file:

  ```
  atoum {{[-f|--files]}} {{path/to/file}}
  ```

- Run a specific directory of tests:

  ```
  atoum {{[-d|--directories]}} {{path/to/directory}}
  ```

- Run all tests under a specific namespace:

  ```
  atoum {{[-ns|--namespaces]}} {{namespace}}
  ```

- Run all tests with a specific tag:

  ```
  atoum {{[-t|--tags]}} {{tag}}
  ```

- Load a custom bootstrap file before running tests:

  ```
  atoum {{[-bf|--bootstrap-file]}} {{path/to/file}}
  ```

atq

Show jobs scheduled by **`at`** or **`batch`** commands.

More information: https://manned.org/atq.

- Show the current user's scheduled jobs:

```
atq
```

- Show jobs from the `a` [q]ueue (queues have single-character names):

```
atq -q {{a}}
```

- Show jobs of all users (run as superuser):

```
sudo atq
```

atrm

Remove jobs scheduled by **`at`** or **`batch`** commands.

To find job numbers use **`atq`**.

More information: https://manned.org/atrm.

- Remove job number 10:

```
atrm {{10}}
```

- Remove many jobs, separated by spaces:

```
atrm {{15}} {{17}} {{22}}
```

atuin

Store your shell history in a searchable database.

Optionally sync your encrypted history between machines.

More information: https://docs.atuin.sh/.

- Install atuin into your shell:

```
eval "$(atuin init {{bash|zsh|fish}})"
```

- Import history from the shell default history file:

```
atuin import auto
```

- Search shell history for a specific command:

```
atuin search {{command}}
```

- Register an account on the default sync server using the specified username, email and password:

```
atuin register {{[-u|--username]}} {{username}} {{[-e|--email]}} {{email}} {{[-p|--password]}} {{password}}
```

- Login to the default sync server:

```
atuin login {{[-u|--username]}} {{username}} {{[-p|--password]}} {{password}}
```

- Sync history with the sync server:

```
atuin sync
```

audacious

An open-source audio player. Indirectly based on XMMS.

See also: **`audtool`**, **`clementine`**, **`mpc`**, **`ncmpcpp`**.

More information: https://audacious-media-player.org.

- Launch the GUI:

```
audacious
```

- Start a new instance and play an audio:

```
audacious --new-instance {{path/to/audio}}
```

- Enqueue a specific directory of audio files:

```
audacious --enqueue {{path/to/directory}}
```

- Start or stop playback:

```
audacious --play-pause
```

- Skip forwards ([fwd]) or backwards ([rew]) in the playlist:

```
audacious --{{fwd|rew}}
```

- Stop playback:

```
audacious --stop
```

- Start in CLI mode (headless):

```
audacious --headless
```

- Exit as soon as playback stops or there is nothing to playback:

```
audacious --quit-after-play
```

auditd

This responds to requests from the audit utility and notifications from the kernel.

It should not be invoked manually.

More information: https://manned.org/auditd.

- Start the daemon:

```
auditd
```

- Start the daemon in debug mode:

```
auditd -d
```

- Start the daemon on-demand from launchd:

```
auditd -l
```

audtool

Control Audacious using commands.

See also: **audacious**.

More information: https://manned.org/audtool.

- Play/pause audio playback:

```
audtool playback-playpause
```

- Print artist, album, and song name of currently playing song:

```
audtool current-song
```

- Set volume of audio playback:

```
audtool set-volume {{100}}
```

- Skip to the next song:

```
audtool playlist-advance
```

- Print the bitrate of the current song in kilobits:

```
audtool current-song-bitrate-kbps
```

- Open Audacious in full-screen if hidden:

```
audtool mainwin-show
```

- Display help:

```
audtool help
```

- Display settings:

```
audtool preferences-show
```

autoconf

Generate configuration scripts to automatically configure software source code packages.

More information: https://manned.org/autoconf.

- Generate a configuration script from `configure.ac` (if present) or `configure.in` and save this script to `configure`:

```
autoconf
```

- Generate a configuration script from the specified template; output to `stdout`:

```
autoconf {{template-file}}
```

- Generate a configuration script from the specified template (even if the input file has not changed) and write the output to a file:

```
autoconf {{[-f|--force]}} {{[-o|--output]}} {{outfile}} {{template-file}}
```

autoflake

Remove unused imports and variables from Python code.

More information: https://github.com/myint/autoflake.

- Remove unused variables from a single file and display the diff:

```
autoflake --remove-unused-variables {{path/to/file.py}}
```

- Remove unused imports from multiple files and display the diffs:

```
autoflake --remove-all-unused-imports {{path/to/file1.py
path/to/file2.py ...}}
```

- Remove unused variables from a file, overwriting the file:

```
autoflake --remove-unused-variables --in-place {{path/to/
file.py}}
```

- Remove unused variables recursively from all files in a directory, overwriting each file:

```
autoflake --remove-unused-variables --in-place --recursive
{{path/to/directory}}
```

autojump

Quickly jump among the directories you visit the most.

Aliases like **`j`** or **`jc`** are provided for even less typing.

See also: **`bashmarks`**.

More information: https://github.com/wting/autojump.

- Add the `autojump` aliases to your shell:

```
source /usr/share/autojump/autojump.{{bash|fish|zsh}}
```

- Jump to a directory that contains the given pattern:

```
j {{pattern}}
```

- Jump to a sub-directory (child) of the current directory that contains the given pattern:

```
jc {{pattern}}
```

- Open a directory that contains the given pattern in the operating system file manager:

```
jo {{pattern}}
```

- Remove non-existing directories from the `autojump` database:

```
j --purge
```

- Show the entries in the `autojump` database:

```
j {{[-s|--stat]}}
```

automake

Automated Makefile generation for software projects using GNU standards.

More information: https://www.gnu.org/software/automake/manual/automake.html#automake-Invocation.

- Run automake to regenerate Makefiles after editing `Makefile.am`:

```
automake
```

- Generate `Makefile.in` for a non-GNU project (foreign mode):

```
automake --foreign
```

- Add verbose output for debugging:

```
automake {{[-v|--verbose]}}
```

- Add missing standard files (INSTALL, COPYING, depcomp, etc.):

```
automake {{[-a|--add-missing]}}
```

- Display help:

```
automake --help
```

autopep8

Format Python code according to the PEP 8 style guide.

More information: https://github.com/hhatto/autopep8.

- Format a file to `stdout`, with a custom maximum line length:

```
autopep8 {{path/to/file.py}} --max-line-length {{length}}
```

- Format a file, displaying a diff of the changes:

```
autopep8 --diff {{path/to/file}}
```

- Format a file in-place and save the changes:

```
autopep8 --in-place {{path/to/file.py}}
```

- Recursively format all files in a directory in-place and save changes:

```
autopep8 --in-place --recursive {{path/to/directory}}
```

autossh

Run, monitor and restart SSH connections.

Auto-reconnects to keep port forwarding tunnels up. Accepts all SSH flags.

More information: https://manned.org/autossh.

- Start an SSH session, restarting when the [M]onitoring port fails to return data:

```
autossh -M {{monitor_port}} "{{ssh_command}}"
```

- Forward a [L]ocal port to a remote one, restarting when necessary:

```
autossh -M {{monitor_port}} -L {{local_port}}:localhost:
{{remote_port}} {{user}}@{{host}}
```

- Fork `autossh` into the background before executing SSH and do [N]ot open a remote shell:

```
autossh -f -M {{monitor_port}} -N "{{ssh_command}}"
```

- Run in the background, with no monitoring port, and instead send SSH keep-alive packets every 10 seconds to detect failure:

```
autossh -f -M 0 -N -o "ServerAliveInterval 10" -o
"ServerAliveCountMax 3" "{{ssh_command}}"
```

- Run in the background, with no monitoring port and no remote shell, exiting if the port forward fails:

```
autossh -f -M 0 -N -o "ServerAliveInterval 10" -o
"ServerAliveCountMax 3" -o ExitOnForwardFailure=yes -L
{{local_port}}:localhost:{{remote_port}} {{user}}@{{host}}
```

- Run in the background, logging `autossh` debug output and SSH verbose output to files:

```
AUTOSSH_DEBUG=1 AUTOSSH_LOGFILE={{path/to/
autossh_log_file.log}} autossh -f -M {{monitor_port}} -v -E
{{path/to/ssh_log_file.log}} {{ssh_command}}
```

avo

The official interface for Avo.

More information: https://www.avo.app/docs/implementation/cli.

- Initialize a workspace in the current directory:

```
avo init
```

- Log into the Avo platform:

```
avo login
```

- Switch to an existing Avo branch:

```
avo checkout {{branch_name}}
```

- Pull analytics wrappers for the current path:

```
avo pull
```

- Display the status of the Avo implementation:

```
avo status
```

- Resolve Git conflicts in Avo files:

```
avo conflict
```

- Open the current Avo workspace in the default web browser:

```
avo edit
```

- Display help for a subcommand:

```
avo {{subcommand}} --help
```

avrdude

Driver program for Atmel AVR microcontrollers programming.

More information: https://www.nongnu.org/avrdude/user-manual/avrdude_3.html#Option-Descriptions.

- [r]ead the flash ROM of a AVR microcontroller with a specific [p]art ID:

```
avrdude -p {{part_no}} -c {{programmer_id}} -U flash:r:{{file.hex}}:i
```

- [w]rite to the flash ROM AVR microcontroller:

```
avrdude -p {{part_no}} -c {{programmer}} -U flash:w:{{file.hex}}
```

- List available AVR devices:

```
avrdude -p \?
```

- List available AVR programmers:

```
avrdude -c \?
```

awk

A versatile programming language for working on files.

Note: Different implementations of AWK often make this a symlink of their binary.

See also: **gawk**.

More information: https://github.com/onetrueawk/awk.

- Print the fifth column (a.k.a. field) in a space-separated file:

```
awk '{print $5}' {{path/to/file}}
```

- Print the second column of the lines containing "foo" in a space-separated file:

```
awk '/{{foo}}/ {print $2}' {{path/to/file}}
```

- Print the last column of each line in a file, using a comma (instead of space) as a field separator:

```
awk -F ',' '{print $NF}' {{path/to/file}}
```

- Sum the values in the first column of a file and print the total:

```
awk '{s+=$1} END {print s}' {{path/to/file}}
```

- Print every third line starting from the first line:

```
awk 'NR%3==1' {{path/to/file}}
```

- Print different values based on conditions:

```
awk '{if ($1 == "foo") print "Exact match foo"; else if ($1 ~ "bar") print "Partial match bar"; else print "Baz"}' {{path/to/file}}
```

- Print all the lines which the 10th column value is between a min and a max:

```
awk '($10 >= {{min_value}} && $10 <= {{max_value}})'
```

- Print table of users with UID >=1000 with header and formatted output, using colon as separator (`%-20s` mean: 20 left-align string characters, `%6s` means: 6 right-align string characters):

```
awk 'BEGIN {FS=":";printf "%-20s %6s %25s\n", "Name", "UID", "Shell"} $4 >= 1000 {printf "%-20s %6d %25s\n", $1, $4, $7}' /etc/passwd
```

aws accessanalyzer

Analyze and review resource policies to identify potential security risks.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/accessanalyzer/index.html.

- Create a new Access Analyzer:

```
aws accessanalyzer create-analyzer --analyzer-name
{{analyzer_name}} --type {{type}} --tags {{tags}}
```

- Delete an existing Access Analyzer:

```
aws accessanalyzer delete-analyzer --analyzer-arn
{{analyzer_arn}}
```

- Get details of a specific Access Analyzer:

```
aws accessanalyzer get-analyzer --analyzer-arn
{{analyzer_arn}}
```

- List all Access Analyzers:

```
aws accessanalyzer list-analyzers
```

- Update settings of an Access Analyzer:

```
aws accessanalyzer update-analyzer --analyzer-arn
{{analyzer_arn}} --tags {{new_tags}}
```

- Create a new Access Analyzer archive rule:

```
aws accessanalyzer create-archive-rule --analyzer-arn
{{analyzer_arn}} --rule-name {{rule_name}} --filter
{{filter}}
```

- Delete an Access Analyzer archive rule:

```
aws accessanalyzer delete-archive-rule --analyzer-arn
{{analyzer_arn}} --rule-name {{rule_name}}
```

- List all Access Analyzer archive rules:

```
aws accessanalyzer list-archive-rules --analyzer-arn
{{analyzer_arn}}
```

aws acm-pca

AWS Certificate Manager Private Certificate Authority.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/acm-pca/index.html.

- Create a private certificate authority:

```
aws acm-pca create-certificate-authority --certificate-authority-configuration {{ca_config}} --idempotency-token {{token}} --permanent-deletion-time-in-days {{number}}
```

- Describe a private certificate authority:

```
aws acm-pca describe-certificate-authority --certificate-authority-arn {{ca_arn}}
```

- List private certificate authorities:

```
aws acm-pca list-certificate-authorities
```

- Update a certificate authority:

```
aws acm-pca update-certificate-authority --certificate-authority-arn {{ca_arn}} --certificate-authority-configuration {{ca_config}} --status {{status}}
```

- Delete a private certificate authority:

```
aws acm-pca delete-certificate-authority --certificate-authority-arn {{ca_arn}}
```

- Issue a certificate:

```
aws acm-pca issue-certificate --certificate-authority-arn {{ca_arn}} --certificate-signing-request {{cert_signing_request}} --signing-algorithm {{algorithm}} --validity {{validity}}
```

- Revoke a certificate:

```
aws acm-pca revoke-certificate --certificate-authority-arn {{ca_arn}} --certificate-serial {{serial}} --reason {{reason}}
```

- Get certificate details:

```
aws acm-pca get-certificate --certificate-authority-arn
{{ca_arn}} --certificate-arn {{cert_arn}}
```

aws acm

AWS Certificate Manager.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/acm/index.html.

- Import a certificate:

```
aws acm import-certificate --certificate-arn {{certificate_arn}} --certificate {{certificate}} --private-key {{private_key}} --certificate-chain {{certificate_chain}}
```

- List certificates:

```
aws acm list-certificates
```

- Describe a certificate:

```
aws acm describe-certificate --certificate-arn {{certificate_arn}}
```

- Request a certificate:

```
aws acm request-certificate --domain-name {{domain_name}} --validation-method {{validation_method}}
```

- Delete a certificate:

```
aws acm delete-certificate --certificate-arn {{certificate_arn}}
```

- List certificate validations:

```
aws acm list-certificates --certificate-statuses {{status}}
```

- Get certificate details:

```
aws acm get-certificate --certificate-arn {{certificate_arn}}
```

- Update certificate options:

```
aws acm update-certificate-options --certificate-arn {{certificate_arn}} --options {{options}}
```

aws amplify

Development platform for building secure, scalable mobile and web applications.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/amplify/index.html.

- Create a new Amplify app:

```
aws amplify create-app --name {{app_name}} --description {{description}} --repository {{repo_url}} --platform {{platform}} --environment-variables {{env_vars}} --tags {{tags}}
```

- Delete an existing Amplify app:

```
aws amplify delete-app --app-id {{app_id}}
```

- Get details of a specific Amplify app:

```
aws amplify get-app --app-id {{app_id}}
```

- List all Amplify apps:

```
aws amplify list-apps
```

- Update settings of an Amplify app:

```
aws amplify update-app --app-id {{app_id}} --name {{new_name}} --description {{new_description}} --repository {{new_repo_url}} --environment-variables {{new_env_vars}} --tags {{new_tags}}
```

- Add a new backend environment to an Amplify app:

```
aws amplify create-backend-environment --app-id {{app_id}} --environment-name {{env_name}} --deployment-artifacts {{artifacts}}
```

- Remove a backend environment from an Amplify app:

```
aws amplify delete-backend-environment --app-id {{app_id}} --environment-name {{env_name}}
```

- List all backend environments in an Amplify app:

```
aws amplify list-backend-environments --app-id {{app_id}}
```

aws backup

Unified backup service designed to protect Amazon Web Services services and their associated data.

More information: https://docs.aws.amazon.com/cli/latest/reference/backup/index.html.

- Return BackupPlan details for a specific BackupPlanId:

```
aws backup get-backup-plan --backup-plan-id {{id}}
```

- Create a backup plan using a specific backup plan name and backup rules:

```
aws backup create-backup-plan --backup-plan {{plan}}
```

- Delete a specific backup plan:

```
aws backup delete-backup-plan --backup-plan-id {{id}}
```

- List all active backup plans for the current account:

```
aws backup list-backup-plans
```

- Display details about your report jobs:

```
aws backup list-report-jobs
```

aws batch

Run batch computing workloads through the AWS Batch service.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/batch/index.html.

- List running batch jobs:

```
aws batch list-jobs --job-queue {{queue_name}}
```

- Create compute environment:

```
aws batch create-compute-environment --compute-environment-name {{compute_environment_name}} --type {{type}}
```

- Create batch job queue:

```
aws batch create-job-queue --job-queue-name {{queue_name}} --priority {{priority}} --compute-environment-order {{compute_environment}}
```

- Submit job:

```
aws batch submit-job --job-name {{job_name}} --job-queue {{job_queue}} --job-definition {{job_definition}}
```

- Describe the list of batch jobs:

```
aws batch describe-jobs --jobs {{jobs}}
```

- Cancel job:

```
aws batch cancel-job --job-id {{job_id}} --reason {{reason}}
```

aws ce

Run cost management operations through the AWS Cost Explorer service.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ce/index.html.

- Create anomaly monitor:

```
aws ce create-anomaly-monitor --monitor {{monitor_name}} --monitor-type {{monitor_type}}
```

- Create anomaly subscription:

```
aws ce create-anomaly-subscription --subscription {{subscription_name}} --monitor-arn {{monitor_arn}} --subscribers {{subscribers}}
```

- Get anomalies:

```
aws ce get-anomalies --monitor-arn {{monitor_arn}} --start-time {{start_time}} --end-time {{end_time}}
```

- Get cost and usage:

```
aws ce get-cost-and-usage --time-period {{start_date}}/{{end_date}} --granularity {{granularity}} --metrics {{metrics}}
```

- Get cost forecast:

```
aws ce get-cost-forecast --time-period {{start_date}}/{{end_date}} --granularity {{granularity}} --metric {{metric}}
```

- Get reservation utilization:

```
aws ce get-reservation-utilization --time-period {{start_date}}/{{end_date}} --granularity {{granularity}}
```

- List cost category definitions:

```
aws ce list-cost-category-definitions
```

- Tag resource:

```
aws ce tag-resource --resource-arn {{resource_arn}} --tags {{tags}}
```

aws cloud9

Manage Cloud9 - a collection of tools to code, build, run, test, debug, and release software in the cloud.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/cloud9/index.html.

- List all Cloud9 development environment identifiers:

```
aws cloud9 list-environments
```

- Create a Cloud9 development environment:

```
aws cloud9 create-environment-ec2 --name {{name}} --instance-type {{instance_type}}
```

- Display information about Cloud9 development environments:

```
aws cloud9 describe-environments --environment-ids {{environment_ids}}
```

- Add an environment member to a Cloud9 development environment:

```
aws cloud9 create-environment-membership --environment-id {{environment_id}} --user-arn {{user_arn}} --permissions {{permissions}}
```

- Display status information for a Cloud9 development environment:

```
aws cloud9 describe-environment-status --environment-id {{environment_id}}
```

- Delete a Cloud9 environment:

```
aws cloud9 delete-environment --environment-id {{environment_id}}
```

- Delete an environment member from a development environment:

```
aws cloud9 delete-environment-membership --environment-id {{environment_id}} --user-arn {{user_arn}}
```

aws cloudformation

Model, provision, and manage AWS and third-party resources by treating infrastructure as code.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/cloudformation/index.html.

- Create a stack from a template file:

```
aws cloudformation create-stack --stack-name {{stack-name}} --region {{region}} --template-body {{file://path/to/file.yml}} --profile {{profile}}
```

- Delete a stack:

```
aws cloudformation delete-stack --stack-name {{stack-name}} --profile {{profile}}
```

- List all stacks:

```
aws cloudformation list-stacks --profile {{profile}}
```

- List all running stacks:

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE --profile {{profile}}
```

- Check the status of a stack:

```
aws cloudformation describe-stacks --stack-name {{stack-id}} --profile {{profile}}
```

- Initiate drift detection for a stack:

```
aws cloudformation detect-stack-drift --stack-name {{stack-id}} --profile {{profile}}
```

- Check the drift status output of a stack using `StackDriftDetectionId` from the previous command output:

```
aws cloudformation describe-stack-resource-drifts --stack-name {{stack-drift-detection-id}} --profile {{profile}}
```

aws cloudwatch

Monitor AWS resources to gain system-wide visibility into resource utilization, application performance, and operational health.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/cloudwatch/index.html.

- List dashboards for your account:

```
aws cloudwatch list-dashboards
```

- Display details for the specified dashboard:

```
aws cloudwatch get-dashboard --dashboard-name
{{dashboard_name}}
```

- List metrics:

```
aws cloudwatch list-metrics
```

- List alarms:

```
aws cloudwatch describe-alarms
```

- Create or update an alarm and associate it with a metric:

```
aws cloudwatch put-metric-alarm --alarm-name {{alarm_name}}
--evaluation-periods {{evaluation_periods}} --comparison-
operator {{comparison_operator}}
```

- Delete the specified alarms:

```
aws cloudwatch delete-alarms --alarm_names {{alarm_names}}
```

- Delete the specified dashboards:

```
aws cloudwatch delete-dashboards --dashboard-names
{{dashboard_names}}
```

aws codeartifact

Manage CodeArtifact repositories, domains, packages, package versions and assets.

CodeArtifact is an artifact repository compatible with popular package managers and build tools like Maven, Gradle, npm, Yarn, Twine, pip, NuGet, and SwiftPM.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/codeartifact/index.html.

- List available domains for your AWS account:

```
aws codeartifact list-domains
```

- Generate credentials for a specific package manager:

```
aws codeartifact login --tool {{npm|pip|twine}} --domain {{your_domain}} --repository {{repository_name}}
```

- Get the endpoint URL of a CodeArtifact repository:

```
aws codeartifact get-repository-endpoint --domain {{your_domain}} --repository {{repository_name}} --format {{npm|pypi|maven|nuget|generic}}
```

- Display help:

```
aws codeartifact help
```

- Display help for a specific subcommand:

```
aws codeartifact {{subcommand}} help
```

aws codecommit

A managed source control service that hosts private Git repositories.

More information: https://docs.aws.amazon.com/cli/latest/reference/codecommit/.

- Display help:

```
aws codecommit help
```

- Display help for a specific command:

```
aws codecommit {{command}} help
```

aws cognito-idp

Configure an Amazon Cognito user pool and its users and groups and authenticate them.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/cognito-idp/index.html.

- Create a new Cognito user pool:

```
aws cognito-idp create-user-pool --pool-name {{name}}
```

- List all user pools:

```
aws cognito-idp list-user-pools --max-results {{10}}
```

- Delete a specific user pool:

```
aws cognito-idp delete-user-pool --user-pool-id
{{user_pool_id}}
```

- Create a user in a specific pool:

```
aws cognito-idp admin-create-user --username {{username}} --
user-pool-id {{user_pool_id}}
```

- List the users of a specific pool:

```
aws cognito-idp list-users --user-pool-id {{user_pool_id}}
```

- Delete a user from a specific pool:

```
aws cognito-idp admin-delete-user --username {{username}} --
user-pool-id {{user_pool_id}}
```

aws configure

Manage configuration for the AWS CLI.

More information: https://docs.aws.amazon.com/cli/latest/reference/configure/.

- Configure AWS CLI interactively (creates a new configuration or updates the default):

```
aws configure
```

- Configure a named profile for AWS CLI interactively (creates a new profile or updates an existing one):

```
aws configure --profile {{profile_name}}
```

- Display the value from a specific configuration variable:

```
aws configure get {{name}}
```

- Display the value for a configuration variable in a specific profile:

```
aws configure get {{name}} --profile {{profile_name}}
```

- Set the value of a specific configuration variable:

```
aws configure set {{name}} {{value}}
```

- Set the value of a configuration variable in a specific profile:

```
aws configure set {{name}} {{value}} --profile
{{profile_name}}
```

- List the configuration entries:

```
aws configure list
```

- List the configuration entries for a specific profile:

```
aws configure list --profile {{profile_name}}
```

aws cur

Create, query, and delete AWS usage report definitions.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/cur/index.html.

- Create an AWS cost and usage report definition from a JSON file:

```
aws cur put-report-definition --report-definition file://{{path/to/report_definition.json}}
```

- List usage report definitions defined for the logged in account:

```
aws cur describe-report-definitions
```

- Delete a usage report definition:

```
aws cur --region {{aws_region}} delete-report-definition --report-name {{report}}
```

aws dynamodb

Manipulate an AWS Dynamodb database, a fast NoSQL database with predictable performance and seamless scalability.

More information: https://docs.aws.amazon.com/cli/latest/reference/dynamodb/.

- Create a table:

```
aws dynamodb create-table --table-name {{table_name}} --attribute-definitions {{AttributeName=S,AttributeType=S}} --key-schema {{AttributeName=S,KeyType=HASH}} --provisioned-throughput {{ReadCapacityUnits=5,WriteCapacityUnits=5}}
```

- List all tables in the DynamoDB:

```
aws dynamodb list-tables
```

- Get details about a specific table:

```
aws dynamodb describe-table --table-name {{table_name}}
```

- Add an item to a table:

```
aws dynamodb put-item --table-name {{table_name}} --item '{{{"AttributeName": {"S": "value"}}}}'
```

- Retrieve an item from a table:

```
aws dynamodb get-item --table-name {{table_name}} --key '{{{"ID": {"N": "1"}}}}'
```

- Update an item in the table:

```
aws dynamodb update-item --table-name {{table_name}} --key '{{{"ID": {"N": "1"}}}}' --update-expression "{{SET Name = :n}}" --expression-attribute-values '{{{":n": {"S": "Jane"}}}}'
```

- Scan items in the table:

```
aws dynamodb scan --table-name {{table_name}}
```

- Delete an item from the table:

```
aws dynamodb delete-item --table-name {{table_name}} --key '{{{"ID": {"N": "1"}}}}'
```

aws ec2

Manage AWS EC2 instances and volumes.

AWS EC2 provides secure and resizable computing capacity in the AWS cloud for faster development and deployment of applications.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/index.html.

- Display information about a specific instance:

```
aws ec2 describe-instances --instance-ids {{instance_id}}
```

- Display information about all instances:

```
aws ec2 describe-instances
```

- Display information about all EC2 volumes:

```
aws ec2 describe-volumes
```

- Delete an EC2 volume:

```
aws ec2 delete-volume --volume-id {{volume_id}}
```

- Create a snapshot from an EC2 volume:

```
aws ec2 create-snapshot --volume-id {{volume_id}}
```

- List available AMIs (Amazon Machine Images):

```
aws ec2 describe-images
```

- Show list of all available EC2 commands:

```
aws ec2 help
```

- Display help for specific EC2 subcommand:

```
aws ec2 {{subcommand}} help
```

aws ecr

Push, pull, and manage container images.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecr/index.html.

- Authenticate Docker with the default registry (username is AWS):

```
aws ecr get-login-password --region {{region}} | {{docker login}} --username AWS --password-stdin {{aws_account_id}}.dkr.ecr.{{region}}.amazonaws.com
```

- Create a repository:

```
aws ecr create-repository --repository-name {{repository}} --image-scanning-configuration scanOnPush={{true|false}} --region {{region}}
```

- Tag a local image for ECR:

```
docker tag {{container_name}}:{{tag}} {{aws_account_id}}.dkr.ecr.{{region}}.amazonaws.com/{{container_name}}:{{tag}}
```

- Push an image to a repository:

```
docker push {{aws_account_id}}.dkr.ecr.{{region}}.amazonaws.com/{{container_name}}:{{tag}}
```

- Pull an image from a repository:

```
docker pull {{aws_account_id}}.dkr.ecr.{{region}}.amazonaws.com/{{container_name}}:{{tag}}
```

- Delete an image from a repository:

```
aws ecr batch-delete-image --repository-name {{repository}} --image-ids imageTag={{latest}}
```

- Delete a repository:

```
aws ecr delete-repository --repository-name {{repository}} --force
```

- List images within a repository:

```
aws ecr list-images --repository-name {{repository}}
```

aws eks

Manage Amazon Elastic Kubernetes Service (EKS) addons, clusters, and node groups.

Amazon EKS is a service for easily running Kubernetes on AWS.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/eks/index.html.

- Create an EKS Cluster:

```
aws eks create-cluster --name {{cluster_name}} --role-arn {{eks_service_role_arn}} --resources-vpc-config subnetIds={{subnet_ids}},securityGroupIds={{security_group_ids}}
```

- Update kubeconfig to connect to the EKS Cluster:

```
aws eks update-kubeconfig --name {{cluster_name}}
```

- List available EKS clusters:

```
aws eks list-clusters
```

- Describe EKS cluster details:

```
aws eks describe-cluster --name {{cluster_name}}
```

- Delete an EKS Cluster:

```
aws eks delete-cluster --name {{cluster_name}}
```

- List nodegroups in an EKS cluster:

```
aws eks list-nodegroups --cluster-name {{cluster_name}}
```

- Describe nodegroup details:

```
aws eks describe-nodegroup --cluster-name {{cluster_name}} --nodegroup-name {{nodegroup_name}}
```

aws glue

CLI for AWS Glue.

Define the public endpoint for the AWS Glue service.

More information: https://docs.aws.amazon.com/cli/latest/reference/glue/.

- List jobs:

```
aws glue list-jobs
```

- Start a job:

```
aws glue start-job-run --job-name {{job_name}}
```

- Start running a workflow:

```
aws glue start-workflow-run --name {{workflow_name}}
```

- List triggers:

```
aws glue list-triggers
```

- Start a trigger:

```
aws glue start-trigger --name {{trigger_name}}
```

- Create a dev endpoint:

```
aws glue create-dev-endpoint --endpoint-name {{name}} --role-
arn {{role_arn_used_by_endpoint}}
```

aws-google-auth

Acquire AWS temporary (STS) credentials using Google Apps as a federated (Single Sign-On) provider.

More information: https://github.com/cevoaustralia/aws-google-auth.

- Log in with Google SSO using the specified username IDP and SP identifiers and set the credentials duration to one hour:

```
aws-google-auth {{[-u|--username]}} {{example@example.com}} {{[-I|--idp-id]}} {{$GOOGLE_IDP_ID}} {{[-S|--sp-id]}} {{$GOOGLE_SP_ID}} {{[-d|--duration]}} {{3600}}
```

- Log in asking which role to use (in case of several available SAML roles):

```
aws-google-auth {{[-u|--username]}} {{example@example.com}} {{[-I|--idp-id]}} {{$GOOGLE_IDP_ID}} {{[-S|--sp-id]}} {{$GOOGLE_SP_ID}} {{[-d|--duration]}} {{3600}} {{[-a|--ask-role]}}
```

- Resolve aliases for AWS accounts:

```
aws-google-auth {{[-u|--username]}} {{example@example.com}} {{[-I|--idp-id]}} {{$GOOGLE_IDP_ID}} {{[-S|--sp-id]}} {{$GOOGLE_SP_ID}} {{[-d|--duration]}} {{3600}} {{[-a|--ask-role]}} --resolve-aliases
```

- Display help:

```
aws-google-auth {{[-h|--help]}}
```

aws help

Display help information about the AWS CLI.

More information: https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-help.html.

- Display help:

```
aws help
```

- List all available topics:

```
aws help topics
```

- Display help about a specific topic:

```
aws help {{topic_name}}
```

aws history

Print the command-line history for AWS CLI commands (the record of history of AWS CLI commands must be enabled).

More information: https://docs.aws.amazon.com/cli/latest/reference/history/.

- List commands history with command IDs:

```
aws history list
```

- Display events related to a specific command given a command ID:

```
aws history show {{command_id}}
```

aws iam

Interact with Identity and Access Management (IAM), a web service for securely controlling access to AWS services.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/index.html.

- List users:

```
aws iam list-users
```

- List policies:

```
aws iam list-policies
```

- List groups:

```
aws iam list-groups
```

- Get users in a group:

```
aws iam get-group --group-name {{group_name}}
```

- Describe an IAM policy:

```
aws iam get-policy --policy-arn arn:aws:iam::aws:policy/{{policy_name}}
```

- List access keys:

```
aws iam list-access-keys
```

- List access keys for a specific user:

```
aws iam list-access-keys --user-name {{user_name}}
```

- Display help:

```
aws iam help
```

aws kafka

Manage an Amazon MSK (Managed Streaming for Apache Kafka) cluster.

See also: **aws**.

More information: https://docs.aws.amazon.com/cli/latest/reference/kafka/index.html.

- Create a new MSK cluster:

```
aws kafka create-cluster --cluster-name {{cluster_name}} --broker-node-group-info instanceType={{instance_type}},clientSubnets={{subnet_id1 subnet_id2 ...}} --kafka-version {{version}} --number-of-broker-nodes {{number}}
```

- Describe a MSK cluster:

```
aws kafka describe-cluster --cluster-arn {{cluster_arn}}
```

- List all MSK clusters in the current region:

```
aws kafka list-clusters
```

- Create a new MSK configuration:

```
aws kafka create-configuration --name {{configuration_name}} --server-properties file://{{path/to/configuration_file.txt}}
```

- Describe a MSK configuration:

```
aws kafka describe-configuration --arn {{configuration_arn}}
```

- List all MSK configurations in the current region:

```
aws kafka list-configurations
```

- Update the MSK cluster configuration:

```
aws kafka update-cluster-configuration --cluster-arn {{cluster_arn}} --configuration-info arn={{configuration_arn}},revision={{configuration_revision}}
```

- Delete the MSK cluster:

```
aws kafka delete-cluster --cluster-arn {{cluster_arn}}
```

aws kendra

CLI for AWS Kendra.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/kendra/index.html.

- Create an index:

```
aws kendra create-index --name {{name}} --role-arn {{role_arn}}
```

- List indexes:

```
aws kendra list-indexes
```

- Describe an index:

```
aws kendra describe-index --id {{index_id}}
```

- List data sources:

```
aws kendra list-data-sources
```

- Describe a data source:

```
aws kendra describe-data-source --id {{data_source_id}}
```

- List search queries:

```
aws kendra list-query-suggestions --index-id {{index_id}} --query-text {{query_text}}
```

aws kinesis

Interact with the Amazon Kinesis Data Streams, a service that scales elastically for real-time processing of streaming big data.

More information: https://docs.aws.amazon.com/cli/latest/reference/kinesis/index.html#cli-aws-kinesis.

- Show all streams in the account:

```
aws kinesis list-streams
```

- Write one record to a Kinesis stream:

```
aws kinesis put-record --stream-name {{name}} --partition-key {{key}} --data {{base64_encoded_message}}
```

- Write a record to a Kinesis stream with inline base64 encoding:

```
aws kinesis put-record --stream-name {{name}} --partition-key {{key}} --data "$( echo "{{my raw message}}" | base64 )"
```

- List the shards available on a stream:

```
aws kinesis list-shards --stream-name {{name}}
```

- Get a shard iterator for reading from the oldest message in a stream's shard:

```
aws kinesis get-shard-iterator --shard-iterator-type TRIM_HORIZON --stream-name {{name}} --shard-id {{id}}
```

- Read records from a shard, using a shard iterator:

```
aws kinesis get-records --shard-iterator {{iterator}}
```

aws lambda

Use AWS Lambda, a compute service for running code without provisioning or managing servers.

More information: https://docs.aws.amazon.com/cli/latest/reference/lambda/.

- Run a function:

```
aws lambda invoke --function-name {{name}} {{path/to/
response.json}}
```

- Run a function with an input payload in JSON format:

```
aws lambda invoke --function-name {{name}} --payload {{json}}
{{path/to/response.json}}
```

- List functions:

```
aws lambda list-functions
```

- Display the configuration of a function:

```
aws lambda get-function-configuration --function-name
{{name}}
```

- List function aliases:

```
aws lambda list-aliases --function-name {{name}}
```

- Display the reserved concurrency configuration for a function:

```
aws lambda get-function-concurrency --function-name {{name}}
```

- List which AWS services can invoke the function:

```
aws lambda get-policy --function-name {{name}}
```

aws lightsail

Manage Amazon Lightsail resources.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/lightsail/index.html.

- List all virtual private servers, or instances:

```
aws lightsail get-instances
```

- List all bundles (instance plans):

```
aws lightsail list-bundles
```

- List all available instance images, or blueprints:

```
aws lightsail list-blueprints
```

- Create an instance:

```
aws lightsail create-instances --instance-names {{name}} --availability-zone {{region}} --bundle-id {{nano_2_0}} --blueprint-id {{blueprint_id}}
```

- Print the state of a specific instance:

```
aws lightsail get-instance-state --instance-name {{name}}
```

- Stop a specific instance:

```
aws lightsail stop-instance --instance-name {{name}}
```

- Delete a specific instance:

```
aws lightsail delete-instance --instance-name {{name}}
```

aws pricing

Query services, products, and pricing information from Amazon Web Services.

More information: https://docs.aws.amazon.com/cli/latest/reference/pricing/.

- List service codes of a specific region:

```
aws pricing describe-services --region {{us-east-1}}
```

- List attributes for a given service code in a specific region:

```
aws pricing describe-services --service-code {{AmazonEC2}} --
region {{us-east-1}}
```

- Print pricing information for a service code in a specific region:

```
aws pricing get-products --service-code {{AmazonEC2}} --
region {{us-east-1}}
```

- List values for a specific attribute for a service code in a specific region:

```
aws pricing get-attribute-values --service-code {{AmazonEC2}}
--attribute-name {{instanceType}} --region {{us-east-1}}
```

- Print pricing information for a service code using filters for instance type and location:

```
aws pricing get-products --service-code {{AmazonEC2}} --
filters
"{{Type=TERM_MATCH,Field=instanceType,Value=m5.xlarge}}"
"{{Type=TERM_MATCH,Field=location,Value=US East (N.
Virginia)}}" --region {{us-east-1}}
```

aws quicksight

Create, delete, list, search and update AWS QuickSight entities.

More information: https://docs.aws.amazon.com/cli/latest/reference/quicksight/.

- List datasets:

```
aws quicksight list-data-sets --aws-account-id
{{aws_account_id}}
```

- List users:

```
aws quicksight list-users --aws-account-id {{aws_account_id}}
--namespace default
```

- List groups:

```
aws quicksight list-groups --aws-account-id
{{aws_account_id}} --namespace default
```

- List dashboards:

```
aws quicksight list-dashboards --aws-account-id
{{aws_account_id}}
```

- Display detailed information about a dataset:

```
aws quicksight describe-data-set --aws-account-id
{{aws_account_id}} --data-set-id {{data_set_id}}
```

- Display who has access to the dataset and what kind of actions they can perform on the dataset:

```
aws quicksight describe-data-set-permissions --aws-account-id
{{aws_account_id}} --data-set-id {{data_set_id}}
```

aws rds

Use AWS Relational Database Service, a web service for setting up, operating and scaling relational databases.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/rds/index.html.

- Display help for a specific RDS subcommand:

```
aws rds {{subcommand}} help
```

- Stop instance:

```
aws rds stop-db-instance --db-instance-identifier
{{instance_identifier}}
```

- Start instance:

```
aws rds start-db-instance --db-instance-identifier
{{instance_identifier}}
```

- Modify an RDS instance:

```
aws rds modify-db-instance --db-instance-identifier
{{instance_identifier}} {{parameters}} --apply-immediately
```

- Apply updates to an RDS instance:

```
aws rds apply-pending-maintenance-action --resource-
identifier {{database_arn}} --apply-action {{system-update}}
--opt-in-type {{immediate}}
```

- Change an instance identifier:

```
aws rds modify-db-instance --db-instance-identifier
{{old_instance_identifier}} --new-db-instance-identifier
{{new_instance_identifier}}
```

- Reboot an instance:

```
aws rds reboot-db-instance --db-instance-identifier
{{instance_identifier}}
```

- Delete an instance:

```
aws rds delete-db-instance --db-instance-identifier
{{instance_identifier}} --final-db-snapshot-identifier
{{snapshot_identifier}} --delete-automated-backups
```

aws route53

CLI for AWS Route53 - Route 53 is a highly available and scalable Domain Name System (DNS) web service.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/route53/index.html.

- List all hosted zones, private and public:

```
aws route53 list-hosted-zones
```

- Show all records in a zone:

```
aws route53 list-resource-record-sets --hosted-zone-id {{zone_id}}
```

- Create a new, public zone using a request identifier to retry the operation safely:

```
aws route53 create-hosted-zone --name {{name}} --caller-reference {{request_identifier}}
```

- Delete a zone (if the zone has non-defaults SOA and NS records the command will fail):

```
aws route53 delete-hosted-zone --id {{zone_id}}
```

- Test DNS resolving by Amazon servers of a given zone:

```
aws route53 test-dns-answer --hosted-zone-id {{zone_id}} --record-name {{name}} --record-type {{type}}
```

aws s3 cp

Copy local files or S3 objects to another location locally or in S3.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/cp.html.

- Copy a file from local to a specific bucket:

```
aws s3 cp {{path/to/file}} s3://{{bucket_name}}/{{path/to/remote_file}}
```

- Copy a specific S3 object into another bucket:

```
aws s3 cp s3://{{bucket_name1}}/{{path/to/file}} s3://{{bucket_name2}}/{{path/to/target}}
```

- Copy a specific S3 object into another bucket keeping the original name:

```
aws s3 cp s3://{{bucket_name1}}/{{path/to/file}} s3://{{bucket_name2}}
```

- Copy S3 objects to a local directory recursively:

```
aws s3 cp s3://{{bucket_name}} . --recursive
```

- Display help:

```
aws s3 cp help
```

aws s3 ls

List AWS S3 buckets, folders (prefixes), and files (objects).

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/ls.html.

- List all buckets:

```
aws s3 ls
```

- List files and folders in the root of a bucket (`s3://` is optional):

```
aws s3 ls s3://{{bucket_name}}
```

- List files and folders directly inside a directory:

```
aws s3 ls {{bucket_name}}/{{path/to/directory}}/
```

- List all files in a bucket:

```
aws s3 ls --recursive {{bucket_name}}
```

- List all files in a path with a given prefix:

```
aws s3 ls --recursive {{bucket_name}}/{{path/to/directory}}/{{prefix}}
```

- Display help:

```
aws s3 ls help
```

aws s3 mb

Create S3 buckets.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/mb.html.

- Create an S3 bucket:

```
aws s3 mb s3://{{bucket_name}}
```

- Create an S3 bucket in a specific region:

```
aws s3 mb s3://{{bucket_name}} --region {{region}}
```

- Display help:

```
aws s3 mb help
```

aws s3 mv

Move local files or S3 objects to another location locally or in S3.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/mv.html.

- Move a file from local to a specified bucket:

```
aws s3 mv {{path/to/local_file}} s3://{{bucket_name}}/{{path/to/remote_file}}
```

- Move a specific S3 object into another bucket:

```
aws s3 mv s3://{{bucket_name1}}/{{path/to/file}} s3://{{bucket_name2}}/{{path/to/target}}
```

- Move a specific S3 object into another bucket keeping the original name:

```
aws s3 mv s3://{{bucket_name1}}/{{path/to/file}} s3://{{bucket_name2}}
```

- Display help:

```
aws s3 mv help
```

aws s3 presign

Generate pre-signed URLs for Amazon S3 objects.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/presign.html.

- Generate a pre-signed URL for a specific S3 object that is valid for one hour:

```
aws s3 presign s3://{{bucket_name}}/{{path/to/file}}
```

- Generate a pre-signed URL valid for a specific lifetime:

```
aws s3 presign s3://{{bucket_name}}/{{path/to/file}} --expires-in {{duration_in_seconds}}
```

- Display help:

```
aws s3 presign help
```

aws s3 rb

Delete an empty S3 bucket.

More information: https://docs.aws.amazon.com/cli/latest/reference/s3/rb.html.

- Delete an empty S3 bucket:

```
aws s3 rb s3://{{bucket_name}}
```

- Force delete an S3 bucket and its non-versioned objects (will crash if versioned objects are present):

```
aws s3 rb s3://{{bucket_name}} --force
```

aws s3 rm

Delete S3 objects.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/rm.html.

- Delete a specific S3 object:

```
aws s3 rm s3://{{bucket_name}}/{{path/to/file}}
```

- Preview the deletion of a specific S3 object without deleting it (dry-run):

```
aws s3 rm s3://{{bucket_name}}/{{path/to/file}} --dryrun
```

- Delete an object from a specific S3 access point:

```
aws s3 rm s3://arn:aws:s3:{{region}}:{{account_id}}:
{{access_point}}/{{access_point_name}}/{{object_key}}
```

- Remove all objects from a bucket (empty the bucket):

```
aws s3 rm s3://{{bucket_name}} --recursive
```

- Display help:

```
aws s3 rm help
```

aws s3 sync

Recursively sync files and directories between your local system and an S3 bucket, or between S3 buckets.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/sync.html.

- Sync files in a directory from local to a bucket:

```
aws s3 sync {{path/to/directory}} s3://{{bucket_name}}/{{path/to/remote_location}}
```

- Sync files in a directory from a bucket to local:

```
aws s3 sync s3://{{bucket_name}}/{{path/to/remote_location}} {{path/to/directory}}
```

- Sync objects between two buckets:

```
aws s3 sync s3://{{bucket_source_name}}/{{path/to/remote_location}} s3://{{bucket_target_name}}/{{path/to/remote_location}}
```

- Sync local files to S3 while excluding specific files or directories:

```
aws s3 sync {{path/to/directory}} s3://{{bucket_name}}/{{path/to/remote_location}} --exclude {{path/to/file}} --exclude {{path/to/directory}}/*
```

- Sync objects between buckets and delete destination files not in source:

```
aws s3 sync s3://{{bucket_source_name}}/{{path/to/remote_location}} s3://{{bucket_target_name}}/{{path/to/remote_location}} --delete
```

- Sync to S3 with advanced options (set ACL and storage class):

```
aws s3 sync {{path/to/local_directory}} s3://{{bucket_name}}/{{path/to/remote_location}} --acl {{private|public-read}} --storage-class {{STANDARD_IA|GLACIER}}
```

- Sync files to S3 and skip unchanged ones (compare size and modification time):

```
aws s3 sync {{path/to/directory}} s3://{{bucket_name}}/{{path/to/remote_location}} --size-only
```

- Display help:

```
aws s3 sync help
```

aws s3 website

Set the website configuration for a bucket.

See also: **`aws s3`**.

More information: https://docs.aws.amazon.com/cli/latest/reference/s3/website.html.

- Configure a bucket as a static website:

```
aws s3 website {{s3://bucket-name}} --index-document {{index.html}}
```

- Configure an error page for the website:

```
aws s3 website {{s3://bucket-name}} --index-document {{index.html}} --error-document {{error.html}}
```

aws s3

CLI for AWS S3 - provides storage through web services interfaces.

Some subcommands such as **cp** have their own usage documentation.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/index.html.

- Show files in a bucket:

```
aws s3 ls {{bucket_name}}
```

- Sync files in a directory from local to bucket:

```
aws s3 sync {{path/to/directory}} s3://{{bucket_name}}
```

- Sync files and directories from bucket to local:

```
aws s3 sync s3://{{bucket_name}} {{path/to/target}}
```

- Sync files in a directory with exclusions:

```
aws s3 sync {{path/to/directory}} s3://{{bucket_name}} --exclude {{path/to/file}} --exclude {{path/to/directory}}/*
```

- Remove file from bucket:

```
aws s3 rm s3://{{bucket}}/{{path/to/file}}
```

- Preview changes only:

```
aws s3 {{any_command}} --dryrun
```

aws s3api

Create and delete Amazon S3 buckets and edit bucket properties.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3api/index.html.

- Create bucket in a specific region:

```
aws s3api create-bucket --bucket {{bucket_name}} --region {{region}} --create-bucket-configuration LocationConstraint={{region}}
```

- Delete a bucket:

```
aws s3api delete-bucket --bucket {{bucket_name}}
```

- List buckets:

```
aws s3api list-buckets
```

- List the objects inside of a bucket and only show each object's key and size:

```
aws s3api list-objects --bucket {{bucket_name}} --query '{{Contents[].{Key: Key, Size: Size}}}'
```

- Add an object to a bucket:

```
aws s3api put-object --bucket {{bucket_name}} --key {{object_key}} --body {{path/to/file}}
```

- Download object from a bucket (The output file is always the last argument):

```
aws s3api get-object --bucket {{bucket_name}} --key {{object_key}} {{path/to/output_file}}
```

- Apply an Amazon S3 bucket policy to a specified bucket:

```
aws s3api put-bucket-policy --bucket {{bucket_name}} --policy file://{{path/to/bucket_policy.json}}
```

- Download the Amazon S3 bucket policy from a specified bucket:

```
aws s3api get-bucket-policy --bucket {{bucket_name}} --query Policy --output {{json|table|text|yaml|yaml-stream}} > {{path/to/bucket_policy}}
```

aws secretsmanager

Store, manage, and retrieve secrets.

More information: https://docs.aws.amazon.com/cli/latest/reference/secretsmanager/.

- Show secrets stored by the secrets manager in the current account:

```
aws secretsmanager list-secrets
```

- List all secrets but only show the secret names and ARNs (easy to view):

```
aws secretsmanager list-secrets --query 'SecretList[*].{Name: Name, ARN: ARN}'
```

- Create a secret:

```
aws secretsmanager create-secret --name {{name}} --description "{{secret_description}}" --secret-string '{{secret}}'
```

- Delete a secret (append `--force-delete-without-recovery` to delete immediately without any recovery period):

```
aws secretsmanager delete-secret --secret-id {{name|arn}}
```

- View details of a secret except for secret text:

```
aws secretsmanager describe-secret --secret-id {{name|arn}}
```

- Retrieve the value of a secret (to get the latest version of the secret omit `--version-stage`):

```
aws secretsmanager get-secret-value --secret-id {{name|arn}} --version-stage {{version_of_secret}}
```

- Rotate the secret immediately using a Lambda function:

```
aws secretsmanager rotate-secret --secret-id {{name|arn}} --rotation-lambda-arn {{arn_of_lambda_function}}
```

- Rotate the secret automatically every 30 days using a Lambda function:

```
aws secretsmanager rotate-secret --secret-id {{name|arn}} --rotation-lambda-arn {{arn_of_lambda_function}} --rotation-rules AutomaticallyAfterDays={{30}}
```

aws ses

CLI for AWS Simple Email Service.

High-scale inbound and outbound cloud email service.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ses/index.html.

- Create a new receipt rule set:

```
aws ses create-receipt-rule-set --rule-set-name {{rule_set_name}} --generate-cli-skeleton
```

- Describe the active receipt rule set:

```
aws ses describe-active-receipt-rule-set --generate-cli-skeletion
```

- Describe a specific receipt rule:

```
aws ses describe-receipt-rule --rule-set-name {{rule_set_name}} --rule-name {{rule_name}} --generate-cli-skeleton
```

- List all receipt rule sets:

```
aws ses list-receipt-rule-sets --starting-token {{token_string}} --max-items {{integer}} --generate-cli-skeleton
```

- Delete a specific receipt rule set (the currently active rule set cannot be deleted):

```
aws ses delete-receipt-rule-set --rule-set-name {{rule_set_name}} --generate-cli-skeleton
```

- Delete a specific receipt rule:

```
aws ses delete-receipt-rule --rule-set-name {{rule_set_name}} --rule-name {{rule_name}} --generate-cli-skeleton
```

- Send an email:

```
aws ses send-email --from {{from_address}} --destination
"ToAddresses={{addresses}}" --message
"Subject={Data={{subject_text}},Charset=utf8},Body={Text={Data={{body_
```

- Display help for a specific SES subcommand:

```
aws ses {{subcommand}} help
```

aws sns

Create topics and subscriptions, send and receive messages, and monitor events and logs for the Amazon Simple Notification Service.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/sns/index.html.

- List all objects of a specific type:

```
aws sns list-{{origination-numbers|phone-numbers-opted-out|platform-applications|sms-sandbox-phone-numbers|subscriptions|topics}}
```

- Create a topic with a specific name and show its Amazon Resource Name (ARN):

```
aws sns create-topic --name {{name}}
```

- Subscribe an email address to the topic with a specific ARN and show the subscription ARN:

```
aws sns subscribe --topic-arn {{topic_ARN}} --protocol email --notification-endpoint {{email}}
```

- Publish a message to a specific topic or phone number and show the message ID:

```
aws sns publish {{--topic-arn "arn:aws:sns:us-west-2:123456789012:topic-name"||--phone-number +1-555-555-0100}} --message file://{{path/to/file}}
```

- Delete the subscription with a specific ARN from its topic:

```
aws sns unsubscribe --subscription-arn {{subscription_ARN}}
```

- Create a platform endpoint:

```
aws sns create-platform-endpoint --platform-application-arn {{platform_application_ARN}} --token {{token}}
```

- Add a statement to a topic's access control policy:

```
aws sns add-permission --topic-arn {{topic_ARN}} --label {{topic_label}} --aws-account-id {{account_id}} --action-name
```

```
{{AddPermission|CreatePlatformApplication|DeleteEndpoint|
GetDataProtectionPolicy|GetEndpointAttributes|Subscribe|...}}
```

- Add a tag to the topic with a specific ARN:

```
aws sns tag-resource --resource-arn {{topic_ARN}} --tags
{{Key=tag1_key Key=tag2_key,Value=tag2_value ...}}
```

aws sqs

Create, delete, and send messages to queues for the AWS SQS service.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/sqs/index.html.

- List all availables queues:

```
aws sqs list-queues
```

- Display the URL of a specific queue:

```
aws sqs get-queue-url --queue-name {{queue_name}}
```

- Create a queue with specific attributes from a file in JSON format:

```
aws sqs create-queue --queue-name {{queue_name}} --attributes
{{file://path/to/attributes_file.json}}
```

- Send a specific message to a queue:

```
aws sqs send-message --queue-url https://sqs.
{{region}}.amazonaws.com/{{queue_name}} --message-body
"{{message_body}}" --delay-seconds {{delay}} --message-
attributes {{file://path/to/attributes_file.json}}
```

- Delete the specified message from a queue:

```
aws sqs delete-message --queue-url {{https://queue_url}} --
receipt-handle {{receipt_handle}}
```

- Delete a specific queue:

```
aws sqs delete-queue --queue-url https://sqs.
{{region}}.amazonaws.com/{{queue_name}}
```

- Delete all messages from the specified queue:

```
aws sqs purge-queue --queue-url https://sqs.
{{region}}.amazonaws.com/{{queue_name}}
```

- Enable a specific AWS account to send messages to queue:

```
aws sqs add-permission --queue-url https://sqs.
{{region}}.amazonaws.com/{{queue_name}} --label
{{permission_name}} --aws-account-ids {{account_id}} --
actions SendMessage
```

aws sso

Manage access to AWS resources using Single Sign-On (SSO) credentials.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/sso/index.html.

- Start SSO session and refresh access tokens. Requires setup using `aws configure sso`:

```
aws sso login
```

- End SSO session and clear cached access tokens:

```
aws sso logout
```

- List all AWS accounts accessible to the user:

```
aws sso list-accounts
```

- List all roles accessible to the user for a given AWS account:

```
aws sso list-account-roles --account-id {{account}} --access-token {{token}}
```

- Retrieve short-term credentials for a specific account:

```
aws get-role-credentials --account-id {{account}} --role-name {{role}} --access-token {{token}}
```

aws sts

Security Token Service (STS) allows to request temporary credentials for (IAM) users or federated users.

More information: https://docs.aws.amazon.com/cli/latest/reference/sts/.

- Get temporary security credentials to access specific AWS resources:

  ```
  aws sts assume-role --role-arn {{aws_role_arn}}
  ```

- Get an IAM user or role whose credentials are used to call the operation:

  ```
  aws sts get-caller-identity
  ```

aws-vault

A vault for securely storing and accessing AWS credentials in development environments.

More information: https://github.com/99designs/aws-vault.

- Add credentials to the secure keystore:

  ```
  aws-vault add {{profile}}
  ```

- Execute a command with AWS credentials in the environment:

  ```
  aws-vault exec {{profile}} -- {{aws s3 ls}}
  ```

- Open a browser window and login to the AWS Console:

  ```
  aws-vault login {{profile}}
  ```

- List profiles, along with their credentials and sessions:

  ```
  aws-vault list
  ```

- Rotate AWS credentials:

  ```
  aws-vault rotate {{profile}}
  ```

- Remove credentials from the secure keystore:

  ```
  aws-vault remove {{profile}}
  ```

aws workmail

Manage Amazon WorkMail.

More information: https://awscli.amazonaws.com/v2/documentation/api/latest/reference/workmail/index.html.

- List all WorkMail organizations:

```
aws workmail list-organizations
```

- List all users of a specific organization:

```
aws workmail list-users --organization-id {{organization_id}}
```

- Create a WorkMail user in a specific organization:

```
aws workmail create-user --name {{username}} --display-name
{{name}} --password {{password}} --organization-id
{{organization_id}}
```

- Register and enable a group/user to WorkMail:

```
aws workmail register-to-work-mail --entity-id {{entity_id}}
--email {{email}} --organization-id {{organization_id}}
```

- Create a WorkMail group in a specific organization:

```
aws workmail create-group --name {{group_name}} --
organization-id {{organization_id}}
```

- Associate a member to a specific group:

```
aws workmail associate-member-to-group --group-id
{{group_id}} --member-id {{member_id}} --organization-id
{{organization_id}}
```

- Deregister and disable a user/group from WorkMail:

```
aws workmail deregister-from-work-mail --entity-id
{{entity_id}} --organization-id {{organization_id}}
```

- Delete a user from an organization:

```
aws workmail delete-user --user-id {{user_id}} --
organization-id {{organization_id}}
```

aws

The official CLI tool for Amazon Web Services.

Some subcommands such as **s3** have their own usage documentation.

More information: https://aws.amazon.com/cli.

- Configure the AWS Command-line:

```
aws configure wizard
```

- Configure the AWS Command-line using SSO:

```
aws configure sso
```

- Get the caller identity (used to troubleshoot permissions):

```
aws sts get-caller-identity
```

- List AWS resources in a region and output in YAML:

```
aws dynamodb list-tables --region {{us-east-1}} --output yaml
```

- Use auto prompt to help with a command:

```
aws iam create-user --cli-auto-prompt
```

- Get an interactive wizard for an AWS resource:

```
aws dynamodb wizard {{new_table}}
```

- Generate a JSON CLI Skeleton (useful for infrastructure as code):

```
aws dynamodb update-table --generate-cli-skeleton
```

- Display help for a specific command:

```
aws {{command}} help
```

awslogs

Queries groups, streams and events from Amazon CloudWatch logs.

More information: https://github.com/jorgebastida/awslogs.

- List log groups:

```
awslogs groups
```

- List existing streams for the specified group:

```
awslogs streams {{/var/log/syslog}}
```

- Get logs for any streams in the specified group between 1 and 2 hours ago:

```
awslogs get {{/var/log/syslog}} {{[-s|--start]}} '{{2h ago}}' {{[-e|--end]}} '{{1h ago}}'
```

- Get logs that match a specific CloudWatch Logs Filter pattern:

```
awslogs get {{/aws/lambda/my_lambda_group}} --filter-pattern '{{ERROR}}'
```

- Watch logs for any streams in the specified group:

```
awslogs get {{/var/log/syslog}} ALL --watch
```

axel

Download accelerator.

Supports HTTP, HTTPS, and FTP.

More information: https://manned.org/axel.

- Download a URL to a file:

```
axel {{url}}
```

- Download and specify an output file:

```
axel {{url}} {{[-o|--output]}} {{path/to/file}}
```

- Download with a specific number connections:

```
axel {{[-n|--num-connections]}} {{connections_num}} {{url}}
```

- Search for mirrors:

```
axel {{[-S|--search]}} {{mirrors_num}} {{url}}
```

- Limit download speed (bytes per second):

```
axel {{[-s|--max-speed]}} {{speed}} {{url}}
```

az account

Manage Azure subscription information.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/account.

- List all subscriptions for the logged in account:

  ```
  az account list
  ```

- Set a `subscription` to be the currently active subscription:

  ```
  az account set {{[-s|--subscription]}} {{subscription_id}}
  ```

- List supported regions for the currently active subscription:

  ```
  az account list-locations
  ```

- Print an access token to be used with `MS Graph API`:

  ```
  az account get-access-token --resource-type {{ms-graph}}
  ```

- Print details of the currently active subscription in a specific format:

  ```
  az account show {{[-o|--output]}} {{json|tsv|table|yaml}}
  ```

az acr

Manage private registries with Azure Container Registries.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/acr.

- Create a managed container registry:

```
az acr create {{[-n|--name]}} {{registry_name}} {{[-g|--resource-group]}} {{resource_group}} --sku {{sku}}
```

- Login to a registry:

```
az acr login {{[-n|--name]}} {{registry_name}}
```

- Tag a local image for ACR:

```
docker tag {{image_name}} {{registry_name}}.azurecr.io/{{image_name}}:{{tag}}
```

- Push an image to a registry:

```
docker push {{registry_name}}.azurecr.io/{{image_name}}:{{tag}}
```

- Pull an image from a registry:

```
docker pull {{registry_name}}.azurecr.io/{{image_name}}:{{tag}}
```

- Delete an image from a registry:

```
az acr repository delete {{[-n|--name]}} {{registry_name}} --repository {{image_name}}:{{tag}}
```

- Delete a managed container registry:

```
az acr delete {{[-n|--name]}} {{registry_name}} {{[-g|--resource-group]}} {{resource_group}} {{[-y|--yes]}}
```

- List images within a registry:

```
az acr repository list {{[-n|--name]}} {{registry_name}} --output table
```

az advisor

Manage Azure subscription information.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/advisor.

- List Azure Advisor configuration for the entire subscription:

  ```
  az advisor configuration list
  ```

- Show Azure Advisor configuration for the given subscription or resource group:

  ```
  az advisor configuration show {{[-g|--resource-group]}} {{resource_group}}
  ```

- List Azure Advisor recommendations:

  ```
  az advisor recommendation list
  ```

- Enable Azure Advisor recommendations:

  ```
  az advisor recommendation enable {{[-g|--resource-group]}} {{resource_group}}
  ```

- Disable Azure Advisor recommendations:

  ```
  az advisor recommendation disable {{[-g|--resource-group]}} {{resource_group}}
  ```

az aks

Manage Azure Kubernetes Service (AKS) clusters.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/aks.

- List AKS clusters:

```
az aks list {{[-g|--resource-group]}} {{resource_group}}
```

- Create a new AKS cluster:

```
az aks create {{[-g|--resource-group]}} {{resource_group}}
{{[-n|--name]}} {{name}} {{[-c|--node-count]}} {{count}} --
node-vm-size {{size}}
```

- Delete an AKS cluster:

```
az aks delete {{[-g|--resource-group]}} {{resource_group}}
{{[-n|--name]}} {{name}}
```

- Get the access credentials for an AKS cluster:

```
az aks get-credentials {{[-g|--resource-group]}}
{{resource_group}} {{[-n|--name]}} {{name}}
```

- Get the upgrade versions available for an AKS cluster:

```
az aks get-upgrades {{[-g|--resource-group]}}
{{resource_group}} {{[-n|--name]}} {{name}}
```

az apim

Manage Azure API Management services.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/apim.

- List API Management services within a resource group:

```
az apim list {{[-g|--resource-group]}} {{resource_group}}
```

- Create an API Management service instance:

```
az apim create {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}} --publisher-email {{email}} --publisher-name {{name}}
```

- Delete an API Management service:

```
az apim delete {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}}
```

- Show details of an API Management service instance:

```
az apim show {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}}
```

- Update an API Management service instance:

```
az apim update {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}}
```

az appconfig

Manage App configurations on Azure.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/appconfig.

- Create an App Configuration:

```
az appconfig create {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{group_name}} {{[-l|--location]}} {{location}}
```

- Delete a specific App Configuration:

```
az appconfig delete {{[-g|--resource-group]}} {{rg_name}} {{[-n|--name]}} {{appconfig_name}}
```

- List all App Configurations under the current subscription:

```
az appconfig list
```

- List all App Configurations under a specific resource group:

```
az appconfig list {{[-g|--resource-group]}} {{rg_name}}
```

- Show properties of an App Configuration:

```
az appconfig show {{[-n|--name]}} {{appconfig_name}}
```

- Update a specific App Configuration:

```
az appconfig update {{[-g|--resource-group]}} {{rg_name}} {{[-n|--name]}} {{appconfig_name}}
```

az bicep

Bicep CLI command group.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/bicep.

- Install Bicep CLI:

```
az bicep install
```

- Build a Bicep file:

```
az bicep build {{[-f|--file]}} {{path/to/file.bicep}}
```

- Attempt to decompile an ARM template file to a Bicep file:

```
az bicep decompile {{[-f|--file]}} {{path/to/
template_file.json}}
```

- Upgrade Bicep CLI to the latest version:

```
az bicep upgrade
```

- Display the installed version of Bicep CLI:

```
az bicep version
```

- List all available versions of Bicep CLI:

```
az bicep list-versions
```

- Uninstall Bicep CLI:

```
az bicep uninstall
```

az config

Manage Azure CLI configuration.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/config.

- Print all configurations:

```
az config get
```

- Print configurations for a specific section:

```
az config get {{section_name}}
```

- Set a configuration:

```
az config set {{configuration_name}}={{value}}
```

- Unset a configuration:

```
az config unset {{configuration_name}}
```

az container

Manage Azure Container Instances.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/container.

- Create a container in a container group:

```
az container create {{[-g|--resource-group]}}
{{resource_group}} {{[-n|--name]}} {{name}} --image
{{image_name}} {{[-os|--os-type]}} {{windows|linux}} --cpu
{{number_of_CPU_cores}} --memory {{memory_in_GB}}
```

- Execute a command from within a running container of a container group:

```
az container exec {{[-g|--resource-group]}}
{{resource_group}} {{[-n|--name]}} {{container_group_name}}
--exec-command "{{command}}"
```

- Examine the logs of a container in a container group:

```
az container logs {{[-n|--name]}} {{name}} {{[-g|--resource-
group]}} {{resource_group}}
```

- Get the details of a container group:

```
az container show {{[-n|--name]}} {{name}} {{[-g|--resource-
group]}} {{resource_group}}
```

- Start all containers in a container group:

```
az container start {{[-n|--name]}} {{name}} {{[-g|--resource-
group]}} {{resource_group}}
```

- Stop all containers in a container group:

```
az container stop {{[-n|--name]}} {{name}} {{[-g|--resource-
group]}} {{resource_group}}
```

- Delete a container group:

```
az container delete {{[-n|--name]}} {{name}} {{[-g|--
resource-group]}} {{resource_group}}
```

az devops

Manage Azure DevOps organizations.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/devops.

- Set the Personal Access Token (PAT) to login to a particular organization:

```
az devops login {{[--org|--organization]}}
{{organization_url}}
```

- Open a project in the browser:

```
az devops project show {{[-p|--project]}} {{project_name}} --
open
```

- List members of a specific team working on a particular project:

```
az devops team list-member {{[-p|--project]}}
{{project_name}} --team {{team_name}}
```

- Check the Azure DevOps CLI current configuration:

```
az devops configure {{[-l|--list]}}
```

- Configure the Azure DevOps CLI behavior by setting a default project and a default organization:

```
az devops configure {{[-d|--defaults]}}
project={{project_name}} organization={{organization_url}}
```

az disk

Manage Azure Managed Disks.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/disk.

- Create a managed disk:

```
az disk create {{[-g|--resource-group]}} {{resource_group}}
{{[-n|--name]}} {{disk_name}} {{[-z|--size-gb]}}
{{size_in_gb}}
```

- List managed disks in a resource group:

```
az disk list {{[-g|--resource-group]}} {{resource_group}}
```

- Delete a managed disk:

```
az disk delete {{[-g|--resource-group]}} {{resource_group}}
{{[-n|--name]}} {{disk_name}}
```

- Grant read or write access to a managed disk (for export):

```
az disk grant-access {{[-g|--resource-group]}}
{{resource_group}} {{[-n|--name]}} {{disk_name}} {{[--
access|--access-level]}} {{Read|Write}} --duration-in-seconds
{{seconds}}
```

- Update disk size:

```
az disk update {{[-g|--resource-group]}} {{resource_group}}
{{[-n|--name]}} {{disk_name}} {{[-z|--size-gb]}}
{{new_size_in_gb}}
```

az feedback

Send feedback to the Azure CLI Team.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/reference-index#az-feedback.

- Send feedback to the Azure CLI Team:

```
az feedback
```

az group

Manage resource groups and template deployments.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/group.

- Create a new resource group:

```
az group create {{[-n|--name]}} {{name}} {{[-l|--location]}} {{location}}
```

- Check if a resource group exists:

```
az group exists {{[-n|--name]}} {{name}}
```

- Delete a resource group:

```
az group delete {{[-n|--name]}} {{name}}
```

- Wait until a condition of the resource group is met:

```
az group wait {{[-n|--name]}} {{name}} --{{created|deleted|exists|updated}}
```

az image

Manage custom Virtual Machine Images in Azure.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/image.

- List the custom images under a resource group:

```
az image list {{[-g|--resource-group]}} {{resource_group}}
```

- Create a custom image from managed disks or snapshots:

```
az image create {{[-g|--resource-group]}} {{resource_group}}
{{[-n|--name]}} {{name}} --os-type {{windows|linux}} --source
{{os_disk_source}}
```

- Delete a custom image:

```
az image delete {{[-n|--name]}} {{name}} {{[-g|--resource-
group]}} {{resource_group}}
```

- Show details of a custom image:

```
az image show {{[-n|--name]}} {{name}} {{[-g|--resource-
group]}} {{resource_group}}
```

- Update custom images:

```
az image update {{[-n|--name]}} {{name}} {{[-g|--resource-
group]}} {{resource_group}} --set {{property=value}}
```

az lock

Manage Azure locks.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/lock.

- Create a read-only subscription level lock:

```
az lock create {{[-n|--name]}} {{lock_name}} {{[-t|--lock-type]}} ReadOnly
```

- Create a read-only resource group level lock:

```
az lock create {{[-n|--name]}} {{lock_name}} {{[-g|--resource-group]}} {{group_name}} {{[-t|--lock-type]}} ReadOnly
```

- Delete a subscription level lock:

```
az lock delete {{[-n|--name]}} {{lock_name}}
```

- Delete a resource group level lock:

```
az lock delete {{[-n|--name]}} {{lock_name}} {{[-g|--resource-group]}} {{group_name}}
```

- List out all locks on the subscription level:

```
az lock list
```

- Show a subscription level lock with a specific name:

```
az lock show {{[-n|--name]}} {{lock_name}}
```

az logicapp

Manage Logic Apps in Azure Cloud Services.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/logicapp.

- Create a logic app:

```
az logicapp create {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}} {{[-s|--storage-account]}} {{storage_account}}
```

- Delete a logic app:

```
az logicapp delete {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}}
```

- List logic apps:

```
az logicapp list {{[-g|--resource-group]}} {{resource_group}}
```

- Restart a logic app:

```
az logicapp restart {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}}
```

- Start a logic app:

```
az logicapp start {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}}
```

- Stop a logic app:

```
az logicapp stop {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}}
```

az login

Log in to Azure.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/reference-index#az-login.

- Log in interactively:

```
az login
```

- Log in with a service principal using a client secret:

```
az login --service-principal {{[-u|--username]}} {{http://
azure-cli-service-principal}} {{[-p|--password]}} {{secret}}
--tenant {{someone.onmicrosoft.com}}
```

- Log in with a service principal using a client certificate:

```
az login --service-principal {{[-u|--username]}} {{http://
azure-cli-service-principal}} {{[-p|--password]}} {{path/to/
cert.pem}} {{[-t|--tenant]}} {{someone.onmicrosoft.com}}
```

- Log in using a VM's system assigned identity:

```
az login {{[-i|--identity]}}
```

- Log in using a VM's user assigned identity:

```
az login {{[-i|--identity]}} {{[-u|--username]}} /
subscriptions/{{subscription_id}}/resourcegroups/{{my_rg}}/
providers/Microsoft.ManagedIdentity/userAssignedIdentities/
{{my_id}}
```

az logout

Log out from an Azure subscription.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/reference-index#az-logout.

- Log out from the active account:

```
az logout
```

- Log out a specific username:

```
az logout --username {{alias@example.com}}
```

az network

Manage Azure Network resources.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/network.

- List network resources in a region that are used against a subscription quota:

```
az network list-usages
```

- List all virtual networks in a subscription:

```
az network vnet list
```

- Create a virtual network:

```
az network vnet create --address-prefixes {{10.0.0.0/16}}
{{[-n|--name]}} {{vnet}} {{[-g|--resource-group]}}
{{group_name}} --subnet-name {{subnet}} --subnet-prefixes
{{10.0.0.0/24}}
```

- Enable accelerated networking for a network interface card:

```
az network nic update --accelerated-networking true {{[-n|--
name]}} {{nic}} {{[-g|--resource-group]}} {{resource_group}}
```

az pipelines

Manage Azure Pipelines resources.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/pipelines.

- Create a new Azure Pipeline (YAML based):

```
az pipelines create {{[--org|--organization]}}
{{organization_url}} {{[-p|--project]}} {{project_name}} --
name {{pipeline_name}} --description {{description}} --
repository {{repository_name}} --branch {{branch_name}}
```

- Delete a specific pipeline:

```
az pipelines delete {{[--org|--organization]}}
{{organization_url}} {{[-p|--project]}} {{project_name}} --id
{{pipeline_id}}
```

- List pipelines:

```
az pipelines list {{[--org|--organization]}}
{{organization_url}} {{[-p|--project]}} {{project_name}}
```

- Enqueue a specific pipeline to run:

```
az pipelines run {{[--org|--organization]}}
{{organization_url}} {{[-p|--project]}} {{project_name}} --
name {{pipeline_name}}
```

- Get the details of a specific pipeline:

```
az pipelines show {{[--org|--organization]}}
{{organization_url}} {{[-p|--project]}} {{project_name}} --
name {{pipeline_name}}
```

- Update a specific pipeline:

```
az pipelines update {{[--org|--organization]}}
{{organization_url}} {{[-p|--project]}} {{project_name}} --
name {{pipeline_name}} --new-name {{pipeline_new_name}} --
new-folder-path {{user1/production_pipelines}}
```

- List all agents in a pool:

```
az pipelines agent list {{[--org|--organization]}}
{{organization_url}} --pool-id {{agent_pool}}
```

az provider

Manage resource providers.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/provider.

- Register a provider:

```
az provider register {{[-n|--namespace]}}
{{Microsoft.PolicyInsights}}
```

- Unregister a provider:

```
az provider unregister {{[-n|--namespace]}}
{{Microsoft.Automation}}
```

- List all providers for a subscription:

```
az provider list
```

- Show information about a specific provider:

```
az provider show {{[-n|--namespace]}} {{Microsoft.Storage}}
```

- List all resource types for a specific provider:

```
az provider list --query "[?
namespace=='{{Microsoft.Network}}'].resourceTypes[].resourceType"
```

az quantum

Manage Azure Quantum workspaces and submit quantum jobs to providers (preview, requires quantum extension).

More information: https://learn.microsoft.com/cli/azure/quantum.

- Create a new Azure Quantum workspace:

```
az quantum workspace create {{[-g|--resource-group]}} {{ResourceGroup}} {{[-l|--location]}} {{Location}} {{[-w|--workspace-name]}} {{Workspace}} {{[-a|--storage-account]}} {{MyStorageAccountName}}
```

- List all Azure Quantum workspaces:

```
az quantum workspace list
```

- Set a default Azure Quantum workspace:

```
az quantum workspace set {{[-g|--resource-group]}} {{ResourceGroup}} {{[-w|--workspace-name]}} {{Workspace}}
```

- Submit a QIR quantum job to a target:

```
az quantum job submit {{[-g|--resource-group]}} {{ResourceGroup}} {{[-w|--workspace-name]}} {{Workspace}} {{[-l|--location]}} {{Location}} {{[-t|--target-id]}} {{Id}} --job-name {{Job}} --job-input-file {{QirBitcode.bc}} --job-input-format {{qir.v1}}
```

- List all jobs in a Quantum Workspace:

```
az quantum job list {{[-g|--resource-group]}} {{ResourceGroup}} {{[-l|--location]}} {{Location}} {{[-w|--workspace-name]}} {{Workspace}}
```

- Get the output of a quantum job:

```
az quantum job output {{[-g|--resource-group]}} {{ResourceGroup}} {{[-w|--workspace-name]}} {{Workspace}} --job-id {{Job}}
```

- List available provider offerings in a location:

```
az quantum offerings list {{[-l|--location]}} {{Location}}
```

- Set a default target for job submissions:

```
az quantum target set {{[-t|--target-id]}} {{Id}}
```

az redis

Manage Redis caches.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/redis.

- Create a new Redis cache instance:

```
az redis create --location {{location}} {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}} --sku {{Basic|Premium|Standard}} --vm-size {{c0|c1|c2|c3|c4|c5|c6|p1|p2|p3|p4|p5}}
```

- Update a Redis cache:

```
az redis update {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}} --sku {{Basic|Premium|Standard}} --vm-size {{c0|c1|c2|c3|c4|c5|c6|p1|p2|p3|p4|p5}}
```

- Export data stored in a Redis cache:

```
az redis export --container {{container}} --file-format {{file-format}} {{[-n|--name]}} {{name}} --prefix {{prefix}} {{[-g|--resource-group]}} {{resource_group}}
```

- Delete a Redis cache:

```
az redis delete {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}} {{[-y|--yes]}}
```

az repos

Manage Azure DevOps repos.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/repos.

- List all repos in a specific project:

```
az repos list {{[-p|--project]}} {{project_name}}
```

- Add policy on a specific branch of a specific repository to restrict basic merge:

```
az repos policy merge-strategy create --repository-id
{{repository_id_in_repos_list}} --branch {{branch_name}} --
blocking --enabled --allow-no-fast-forward false --allow-
rebase true --allow-rebase-merge true --allow-squash true
```

- Add build validation on a specific repository, using an existing build pipeline, to be triggered automatically on source update:

```
az repos policy build create --repository-id
{{repository_id}} --build-definition-id {{build_pipeline_id}}
--branch main --blocking --enabled --queue-on-source-update-
only true --display-name {{name}} --valid-duration
{{minutes}}
```

- List all active Pull Requests on a specific repository within a specific project:

```
az repos pr list {{[-p|--project]}} {{project_name}} {{[-r|--
repository]}} {{repository_name}} --status active
```

az serial-console

Connect to the serial console of a Virtual Machine.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/serial-console.

- Connect to a serial console:

```
az serial-console connect {{[-g|--resource-group]}}
{{Resource_Group_Name}} {{[-n|--name]}}
{{Virtual_Machine_Name}}
```

- Terminate the connection:

```
<Ctrl ]>
```

az sshkey

Manage SSH public keys with virtual machines.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/sshkey.

- Create a new SSH key:

```
az sshkey create --name {{name}} {{[-g|--resource-group]}}
{{resource_group}}
```

- Upload an existing SSH key:

```
az sshkey create --name {{name}} {{[-g|--resource-group]}}
{{resource_group}} --public-key "{{@path/to/key.pub}}"
```

- List all SSH public keys:

```
az sshkey list
```

- Show information about an SSH public key:

```
az sshkey show --name {{name}} {{[-g|--resource-group]}}
{{resource_group}}
```

az storage account

Manage storage accounts in Azure.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/storage/account.

- Create an storage account:

```
az storage account create {{[-n|--name]}}
{{storage_account_name}} {{[-g|--resource-group]}}
{{azure_resource_group}} --location {{azure_location}} --sku
{{storage_account_sku}}
```

- Generate a shared access signature for a specific storage account:

```
az storage account generate-sas --account-name
{{storage_account_name}} {{[-n|--name]}} {{account_name}} --
permissions {{sas_permissions}} --expiry {{expiry_date}} --
services {{storage_services}} --resource-types
{{resource_types}}
```

- List storage accounts:

```
az storage account list {{[-g|--resource-group]}}
{{azure_resource_group}}
```

- Delete a specific storage account:

```
az storage account delete {{[-n|--name]}}
{{storage_account_name}} {{[-g|--resource-group]}}
{{azure_resource_group}}
```

az storage blob

Manage blob storage containers and objects in Azure.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/storage/blob.

- Download a blob to a file path specifying a source container:

```
az storage blob download --account-name {{account_name}} --
account-key {{account_key}} {{[-c|--container-name]}}
{{container_name}} {{[-n|--name]}} {{blob_name}} {{[-f|--
file]}} {{path/to/file}}
```

- Download blobs from a blob container recursively:

```
az storage blob download-batch --account-name
{{account_name}} --account-key {{account_key}} {{[-s|--
source]}} {{container_name}} --pattern {{filename_regex}}
{{[-d|--destination]}} {{path/to/destination}}
```

- Upload a local file to blob storage:

```
az storage blob upload --account-name {{account_name}} --
account-key {{account_key}} {{[-c|--container-name]}}
{{container_name}} {{[-n|--name]}} {{blob_name}} {{[-f|--
file]}} {{path/to/file}}
```

- Delete a blob object:

```
az storage blob delete --account-name {{account_name}} --
account-key {{account_key}} {{[-c|--container-name]}}
{{container_name}} {{[-n|--name]}} {{blob_name}}
```

- Generate a shared access signature for a blob:

```
az storage blob generate-sas --account-name {{account_name}}
--account-key {{account_key}} {{[-c|--container-name]}}
{{container_name}} {{[-n|--name]}} {{blob_name}} --
permissions {{permission_set}} --expiry {{Y-m-d'T'H:M'Z'}} --
https-only
```

az storage container

Manage blob storage containers in Azure.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/storage/container.

- Create a container in a storage account:

```
az storage container create --account-name
{{storage_account_name}} {{[-n|--name]}} {{container_name}}
--public-access {{access_level}} --fail-on-exist
```

- Generate a shared access signature for the container:

```
az storage container generate-sas --account-name
{{storage_account_name}} {{[-n|--name]}} {{container_name}}
--permissions {{sas_permissions}} --expiry {{expiry_date}} --
https-only
```

- List containers in a storage account:

```
az storage container list --account-name
{{storage_account_name}} --prefix {{filter_prefix}}
```

- Mark the specified container for deletion:

```
az storage container delete --account-name
{{storage_account_name}} {{[-n|--name]}} {{container_name}}
--fail-not-exist
```

az storage entity

Manage Azure Table storage entities.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/storage/entity.

- Insert an entity into a table:

```
az storage entity insert {{[-e|--entity]}}
{{space_separated_key_value_pairs}} {{[-t|--table-name]}}
{{table_name}} --account-name {{storage_account_name}} --
account-key {{storage_account_key}}
```

- Delete an existing entity from a table:

```
az storage entity delete --partition-key {{partition_key}} --
row-key {{row_key}} {{[-t|--table-name]}} {{table_name}} --
account-name {{storage_account_name}} --account-key
{{storage_account_key}}
```

- Update an existing entity by merging its properties:

```
az storage entity merge {{[-e|--entity]}}
{{space_separated_key_value_pairs}} {{[-t|--table-name]}}
{{table_name}} --account-name {{storage_account_name}} --
account-key {{storage_account_key}}
```

- List entities which satisfy a query:

```
az storage entity query --filter {{query_filter}} {{[-t|--
table-name]}} {{table_name}} --account-name
{{storage_account_name}} --account-key
{{storage_account_key}}
```

- Get an entity from the specified table:

```
az storage entity show --partition-key {{partition_key}} --
row-key {{row_key}} {{[-t|--table-name]}} {{table_name}} --
account-name {{storage_account_name}} --account-key
{{storage_account_key}}
```

az storage queue

Manage storage queues in Azure.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/storage/queue.

- Create a queue:

```
az storage queue create --account-name
{{storage_account_name}} {{[-n|--name]}} {{queue_name}} --
metadata {{queue_metadata}}
```

- Generate a shared access signature for the queue:

```
az storage queue generate-sas --account-name
{{storage_account_name}} {{[-n|--name]}} {{queue_name}} --
permissions {{queue_permissions}} --expiry {{expiry_date}} --
https-only
```

- List queues in a storage account:

```
az storage queue list --prefix {{filter_prefix}} --account-
name {{storage_account_name}}
```

- Delete the specified queue and any messages it contains:

```
az storage queue delete --account-name
{{storage_account_name}} {{[-n|--name]}} {{queue_name}} --
fail-not-exist
```

az storage table

Manage NoSQL key-value storage in Azure.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/storage/table.

- Create a new table in the storage account:

```
az storage table create --account-name
{{storage_account_name}} {{[-n|--name]}} {{table_name}} --
fail-on-exist
```

- Generate a shared access signature for the table:

```
az storage table generate-sas --account-name
{{storage_account_name}} {{[-n|--name]}} {{table_name}} --
permissions {{sas_permissions}} --expiry {{expiry_date}} --
https-only
```

- List tables in a storage account:

```
az storage table list --account-name {{storage_account_name}}
```

- Delete the specified table and any data it contains:

```
az storage table delete --account-name
{{storage_account_name}} {{[-n|--name]}} {{table_name}} --
fail-not-exist
```

az storage

Manage Azure Cloud Storage resources.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/storage.

- Create a storage account specifying a location:

```
az storage account create {{[-g|--resource-group]}} {{group_name}} {{[-n|--name]}} {{account_name}} {{[-l|--location]}} {{location}} --sku {{account_sku}}
```

- List all storage accounts in a resource group:

```
az storage account list {{[-g|--resource-group]}} {{group_name}}
```

- List the access keys for a storage account:

```
az storage account keys list {{[-g|--resource-group]}} {{group_name}} {{[-n|--name]}} {{account_name}}
```

- Delete a storage account:

```
az storage account delete {{[-g|--resource-group]}} {{group_name}} {{[-n|--name]}} {{account_name}}
```

- Update the minimum tls version setting for a storage account:

```
az storage account update --min-tls-version {{TLS1_0|TLS1_1|TLS1_2}} {{[-g|--resource-group]}} {{group_name}} {{[-n|--name]}} {{account_name}}
```

az tag

Manage tags on a resource.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/tag.

- Create a tag value:

```
az tag add-value {{[-n|--name]}} {{tag_name}} --value
{{tag_value}}
```

- Create a tag in the subscription:

```
az tag create {{[-n|--name]}} {{tag_name}}
```

- Delete a tag from the subscription:

```
az tag delete {{[-n|--name]}} {{tag_name}}
```

- List all tags on a subscription:

```
az tag list --resource-id /subscriptions/{{subscription_id}}
```

- Delete a tag value for a specific tag name:

```
az tag remove-value {{[-n|--name]}} {{tag_name}} --value
{{tag_value}}
```

az term

Manage marketplace agreement with marketplaceordering.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/term.

- Print marketplace terms:

```
az term show --product "{{product_identifier}}" --plan
"{{plan_identifier}}" --publisher "{{publisher_identifier}}"
```

- Accept marketplace terms:

```
az term accept --product "{{product_identifier}}" --plan
"{{plan_identifier}}" --publisher "{{publisher_identifier}}"
```

az upgrade

Upgrade Azure CLI and Extensions.

Part of **azure-cli** (also known as **az**).

More information: https://learn.microsoft.com/cli/azure/reference-index?view=azure-cli-latest#az-upgrade.

- Upgrade Azure CLI:

```
az upgrade
```

- Upgrade Azure CLI and Extensions:

```
az upgrade --all
```

- Upgrade Azure CLI and Extensions without prompting for confirmation:

```
az upgrade --all {{[-y|--yes]}}
```

az version

Shows the current version of Azure CLI modules and extensions.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/reference-index?view=azure-cli-latest#az-version.

- Show the current version of Azure CLI modules and extensions in JSON format:

  ```
  az version
  ```

- Show the current version of Azure CLI modules and extensions in a given format:

  ```
  az version {{[-o|--output]}} {{json|table|tsv}}
  ```

az vm

Manage virtual machines in Azure.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/vm.

- Display a table of available Virtual Machines:

```
az vm list --output table
```

- Create a virtual machine using the default Ubuntu image and generate SSH keys:

```
az vm create {{[-g|--resource-group]}} {{rg}} {{[-n|--name]}} {{vm_name}} --image {{UbuntuLTS}} --admin-user {{azureuser}} --generate-ssh-keys
```

- Stop a Virtual Machine:

```
az vm stop {{[-g|--resource-group]}} {{rg}} {{[-n|--name]}} {{vm_name}}
```

- Deallocate a Virtual Machine:

```
az vm deallocate {{[-g|--resource-group]}} {{rg}} {{[-n|--name]}} {{vm_name}}
```

- Start a Virtual Machine:

```
az vm start {{[-g|--resource-group]}} {{rg}} {{[-n|--name]}} {{vm_name}}
```

- Restart a Virtual Machine:

```
az vm restart {{[-g|--resource-group]}} {{rg}} {{[-n|--name]}} {{vm_name}}
```

- List VM images available in the Azure Marketplace:

```
az vm image list
```

az webapp

Manage Web Applications hosted in Azure Cloud Services.

Part of **`azure-cli`** (also known as **`az`**).

More information: https://learn.microsoft.com/cli/azure/webapp.

- List available runtimes for a web application:

```
az webapp list-runtimes {{[-os|--os-type]}} {{windows|linux}}
```

- Create a web application:

```
az webapp up {{[-n|--name]}} {{name}} {{[-l|--location]}} {{location}} {{[-r|--runtime]}} {{runtime}}
```

- List all web applications:

```
az webapp list
```

- Delete a specific web application:

```
az webapp delete {{[-n|--name]}} {{name}} {{[-g|--resource-group]}} {{resource_group}}
```

az

The official CLI tool for Microsoft Azure.

Some subcommands such as **`login`** have their own usage documentation.

More information: https://learn.microsoft.com/cli/azure/reference-index.

- Log in to Azure:

```
az login
```

- Manage azure subscription information:

```
az account
```

- List all Azure Managed Disks:

```
az disk list
```

- List all Azure virtual machines:

```
az vm list
```

- Manage Azure Kubernetes Services:

```
az aks
```

- Manage Azure Network resources:

```
az network
```

- Start in interactive mode:

```
az interactive
```

- Display help:

```
az --help
```

azure-cli

This command is an alias of **az**.

- View documentation for the original command:

```
tldr az
```

azurite

Azure Storage API compatible server (emulator) in local environment.

More information: https://www.npmjs.com/package/azurite.

- Use an existing location as workspace path:

  ```
  azurite {{[-l|--location]}} {{path/to/directory}}
  ```

- Disable access log displayed in console:

  ```
  azurite {{[-s|--silent]}}
  ```

- Enable debug log by providing a file path as log destination:

  ```
  azurite {{[-d|--debug]}} {{path/to/debug.log}}
  ```

- Customize the listening address of Blob/Queue/Table service:

  ```
  azurite {{--blobHost|--queueHost|--tableHost}} {{0.0.0.0}}
  ```

- Customize the listening port of Blob/Queue/Table service:

  ```
  azurite {{--blobPort|--queuePort|--tablePort}} {{8888}}
  ```

b2

Access all features of Backblaze B2 Cloud Storage easily.

More information: https://www.backblaze.com/docs/cloud-storage-command-line-tools.

- Access your account:

```
b2 authorize_account {{key_id}}
```

- List the existing buckets in your account:

```
b2 list_buckets
```

- Create a bucket, provide the bucket name, and access type (e.g. allPublic or allPrivate):

```
b2 create_bucket {{bucket_name}} {{allPublic|allPrivate}}
```

- Upload a file. Choose a file, bucket, and a folder:

```
b2 upload_file {{bucket_name}} {{path/to/file}}
{{folder_name}}
```

- Upload a source directory to a Backblaze B2 bucket destination:

```
b2 sync {{path/to/source_file}} {{bucket_name}}
```

- Copy a file from one bucket to another bucket:

```
b2 copy-file-by-id {{path/to/source_file_id}}
{{destination_bucket_name}} {{path/to/b2_file}}
```

- Show the files in your bucket:

```
b2 ls {{bucket_name}}
```

- Remove a "folder" or a set of files matching a pattern:

```
b2 rm {{path/to/folder|pattern}}
```

b2sum

Calculate BLAKE2 cryptographic checksums.

More information: https://www.gnu.org/software/coreutils/manual/html_node/b2sum-invocation.html.

- Calculate the BLAKE2 checksum for one or more files:

```
b2sum {{path/to/file1 path/to/file2 ...}}
```

- Calculate and save the list of BLAKE2 checksums to a file:

```
b2sum {{path/to/file1 path/to/file2 ...}} > {{path/to/file.b2}}
```

- Calculate a BLAKE2 checksum from `stdin`:

```
{{command}} | b2sum
```

- Read a file of BLAKE2 checksums and filenames and verify all files have matching checksums:

```
b2sum {{[-c|--check]}} {{path/to/file.b2}}
```

- Only show a message for missing files or when verification fails:

```
b2sum {{[-c|--check]}} --quiet {{path/to/file.b2}}
```

- Only show a message when verification fails, ignoring missing files:

```
b2sum --ignore-missing {{[-c|--check]}} --quiet {{path/to/file.b2}}
```

- Check a known BLAKE2 checksum of a file:

```
echo {{known_blake2_checksum_of_the_file}} {{path/to/file}} | b2sum {{[-c|--check]}}
```

b3sum

Calculate BLAKE3 cryptographic checksums.

More information: https://github.com/BLAKE3-team/BLAKE3/tree/master/b3sum.

- Calculate the BLAKE3 checksum for one or more files:

```
b3sum {{path/to/file1 path/to/file2 ...}}
```

- Calculate and save the list of BLAKE3 checksums to a file:

```
b3sum {{path/to/file1 path/to/file2 ...}} > {{path/to/file.b3}}
```

- Calculate a BLAKE3 checksum from `stdin`:

```
{{command}} | b3sum
```

- Read a file of BLAKE3 checksums and filenames and verify all files have matching checksums:

```
b3sum {{[-c|--check]}} {{path/to/file.b3}}
```

- Only show a message for missing files or when verification fails:

```
b3sum {{[-c|--check]}} --quiet {{path/to/file.b3}}
```

- Check a known BLAKE3 checksum of a file:

```
echo {{known_blake3_checksum_of_the_file}} {{path/to/file}} | b3sum {{[-c|--check]}}
```

babel

A transpiler which converts code from JavaScript ES6/ES7 syntax to ES5 syntax.

More information: https://babeljs.io/docs/babel-cli.

- Transpile a specified input file and output to `stdout`:

```
babel {{path/to/file}}
```

- Transpile a specified input file and output to a specific file:

```
babel {{path/to/input_file}} --out-file {{path/to/
output_file}}
```

- Transpile the input file every time it is changed:

```
babel {{path/to/input_file}} --watch
```

- Transpile a whole directory of files:

```
babel {{path/to/input_directory}}
```

- Ignore specified comma-separated files in a directory:

```
babel {{path/to/input_directory}} --ignore
{{ignored_file1,ignored_file2,...}}
```

- Transpile and output as minified JavaScript:

```
babel {{path/to/input_file}} --minified
```

- Choose a set of presets for output formatting:

```
babel {{path/to/input_file}} --presets
{{preset1,preset2,...}}
```

- Display help:

```
babel --help
```

babeld

Routing daemon for Babel which uses firewall-style filters.

More information: https://www.irif.fr/~jch/software/babel/babeld.html.

- Start the daemon with one or more [c]onfiguration files (read in order):

```
babeld -c {{path/to/ports.conf}} -c {{path/to/filters.conf}} -c {{path/to/interfaces.conf}}
```

- [D]eamonize after startup:

```
babeld -D
```

- Specify a [C]onfiguration command:

```
babeld -C '{{redistribute metric 256}}'
```

- Specify on which interfaces to operate:

```
babeld {{eth0}} {{eth1}} {{wlan0}}
```

bacon

A background code checker for Rust.

More information: https://github.com/Canop/bacon.

- Run `cargo check` whenever a change is detected in the current directory:

  ```
  bacon
  ```

- Run `cargo test` whenever a change is detected in the given directory:

  ```
  bacon test {{path/to/directory}}
  ```

- Run `cargo check` against all targets whenever a change is detected in the current directory:

  ```
  bacon check-all
  ```

- Run a specific job whenever a change is detected in the current directory:

  ```
  bacon {{run|test|clippy|doc|...}}
  ```

- List all currently available jobs:

  ```
  bacon --list-jobs
  ```

- Initialize a `bacon.toml` configuration file in the current directory:

  ```
  bacon --init
  ```

badblocks

Search a device for bad blocks.

Some usages of badblocks can cause destructive actions, such as erasing all data on a disk, including the partition table.

More information: https://manned.org/badblocks.

- Search a disk for bad blocks by using a non-destructive read-only test:

```
sudo badblocks {{/dev/sdX}}
```

- Search an unmounted disk for bad blocks with a [n]on-destructive read-write test:

```
sudo badblocks -n {{/dev/sdX}}
```

- Search an unmounted disk for bad blocks with a destructive [w]rite test:

```
sudo badblocks -w {{/dev/sdX}}
```

- Use the destructive [w]rite test and [s]how [v]erbose progress:

```
sudo badblocks -svw {{/dev/sdX}}
```

- In destructive mode, [o]utput found blocks to a file:

```
sudo badblocks -o {{path/to/file}} -w {{/dev/sdX}}
```

- Use the destructive mode with improved speed using 4K [b]lock size and 64K block [c]ount:

```
sudo badblocks -w -b {{4096}} -c {{65536}} {{/dev/sdX}}
```

balena

Interact with the balenaCloud, openBalena and the balena API.

More information: https://www.balena.io/docs/reference/cli/.

- Log in to the balenaCloud account:

  ```
  balena login
  ```

- Create a balenaCloud or openBalena application:

  ```
  balena app create {{app_name}}
  ```

- List all balenaCloud or openBalena applications within the account:

  ```
  balena apps
  ```

- List all devices associated with the balenaCloud or openBalena account:

  ```
  balena devices
  ```

- Flash a balenaOS image to a local drive:

  ```
  balena local flash {{path/to/balenaos.img}} --drive
  {{drive_location}}
  ```

bandwhich

Display the current network utilization by process, connection or remote IP/hostname.

More information: https://github.com/imsnif/bandwhich#usage.

- Show the remote addresses table only:

```
bandwhich {{[-a|--addresses]}}
```

- Show DNS queries:

```
bandwhich {{[-s|--show-dns]}}
```

- Show total (cumulative) usage:

```
bandwhich {{[-t|--total-utilization]}}
```

- Show the network utilization for a specific network interface:

```
bandwhich {{[-i|--interface]}} {{eth0}}
```

- Show DNS queries with a given DNS server:

```
bandwhich {{[-s|--show-dns]}} {{[-d|--dns-server]}} {{dns_server_ip}}
```

banner

Print the argument as a large ASCII art.

More information: https://manned.org/banner.

- Print the text message as a large banner (quotes are optional):

```
banner "{{Hello World}}"
```

- Use a banner width of 50 characters:

```
banner {{[-w|--width]}} 50 "{{Hello World}}"
```

- Read text from `stdin`:

```
banner
```

base32

Encode or decode file or **`stdin`** to/from Base32, to **`stdout`**.

More information: https://manned.org/base32.

- Encode a file:

```
base32 {{path/to/file}}
```

- Wrap encoded output at a specific width (`0` disables wrapping):

```
base32 {{[-w|--wrap]}} {{0|76|...}} {{path/to/file}}
```

- Decode a file:

```
base32 {{[-d|--decode]}} {{path/to/file}}
```

- Encode from `stdin`:

```
{{command}} | base32
```

- Decode from `stdin`:

```
{{command}} | base32 {{[-d|--decode]}}
```

base64

Encode or decode file or **`stdin`** to/from base64, to **`stdout`**.

More information: https://manned.org/base64.

- Encode a file:

```
base64 {{path/to/file}}
```

- Wrap encoded output at a specific width (`0` disables wrapping):

```
base64 {{[-w|--wrap]}} {{0|76|...}} {{path/to/file}}
```

- Decode a file:

```
base64 {{[-d|--decode]}} {{path/to/file}}
```

- Encode from `stdin`:

```
{{command}} | base64
```

- Decode from `stdin`:

```
{{command}} | base64 {{[-d|--decode]}}
```

basename

Remove leading directory portions from a path.

More information: https://www.gnu.org/software/coreutils/manual/html_node/basename-invocation.html.

- Show only the file name from a path:

```
basename {{path/to/file}}
```

- Show only the rightmost directory name from a path:

```
basename {{path/to/directory}}
```

- Show only the file name from a path, with a suffix removed:

```
basename {{path/to/file}} {{suffix}}
```

basenc

Encode or decode file or **`stdin`** using a specified encoding, to **`stdout`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/basenc-invocation.html.

- Encode a file with base64 encoding:

```
basenc --base64 {{path/to/file}}
```

- Decode a file with base64 encoding:

```
basenc {{[-d|--decode]}} --base64 {{path/to/file}}
```

- Encode from `stdin` with base32 encoding with 42 columns:

```
{{command}} | basenc --base32 {{[-w|--wrap]}} 42
```

- Encode from `stdin` with base32 encoding:

```
{{command}} | basenc --base32
```

bash-it

A collection of community contributed Bash commands and scripts for Bash 3.2+.

More information: https://bash-it.readthedocs.io/en/latest/.

- Update Bash-it to the latest stable/development version:

```
bash-it update {{stable|dev}}
```

- Reload Bash profile (set `BASH_IT_AUTOMATIC_RELOAD_AFTER_CONFIG_CHANGE` to non-empty value for an automatic reload):

```
bash-it reload
```

- Restart Bash:

```
bash-it restart
```

- Reload Bash profile with enabled error and warning logging:

```
bash-it doctor
```

- Reload Bash profile with enabled error/warning/entire logging:

```
bash-it doctor {{errors|warnings|all}}
```

- Search for Bash-it aliases/plugins/completions:

```
bash-it search {{alias|plugin|completion}}
```

- Search for Bash-it aliases/plugins/completions and enable/disable all found items:

```
bash-it search --{{enable|disable}} {{alias|plugin|
completion}}
```

bash

Bourne-Again SHell, an **sh**-compatible command-line interpreter.

See also: **zsh**, **histexpand** (history expansion).

More information: https://www.gnu.org/software/bash/manual/bash.html#Invoking-Bash.

- Start an interactive shell session:

```
bash
```

- Start an interactive shell session without loading startup configs:

```
bash --norc
```

- Execute specific [c]ommands:

```
bash -c "{{echo 'bash is executed'}}"
```

- Execute a specific script:

```
bash {{path/to/script.sh}}
```

- E[x]ecute a specific script, printing each command before executing it:

```
bash -x {{path/to/script.sh}}
```

- Execute a specific script and stop at the first [e]rror:

```
bash -e {{path/to/script.sh}}
```

- Execute specific commands from `stdin`:

```
{{echo "echo 'bash is executed'"}} | bash
```

- Start a [r]estricted shell session:

```
bash -r
```

bashmarks

Save and jump to commonly used directories using 1 character commands.

See also: **`autojump`**.

More information: https://github.com/huyng/bashmarks.

- Add the bashmark aliases to your shell:

```
source /usr/share/bashmarks/bashmarks.sh
```

- List available bookmarks:

```
l
```

- Save the current directory as "bookmark_name":

```
s {{bookmark_name}}
```

- Go to a bookmarked directory:

```
g {{bookmark_name}}
```

- Print a bookmarked directory's contents:

```
p {{bookmark_name}}
```

- Delete a bookmark:

```
d {{bookmark_name}}
```

bastet

Clone of the game Tetris in the terminal.

More information: https://fph.altervista.org/prog/bastet.html.

- Start a Tetris game:

  `bastet`

- Navigate the piece horizontally:

  `{{<ArrowLeft>|<ArrowRight>}}`

- Rotate the piece clockwise or counterclockwise:

  `{{<Space>|<ArrowUp>}}`

- Soft drop the piece:

  `<ArrowDown>`

- Hard drop the piece:

  `<Enter>`

- Pause the game:

  `<p>`

- Quit the game:

  `<Ctrl c>`

bat

Print and concatenate files.

A **`cat`** clone with syntax highlighting and Git integration.

More information: https://github.com/sharkdp/bat.

- Pretty print the contents of one or more files to `stdout`:

```
bat {{path/to/file1 path/to/file2 ...}}
```

- Concatenate several files into the target file:

```
bat {{path/to/file1 path/to/file2 ...}} > {{path/to/target_file}}
```

- Remove decorations and disable paging (`--style plain` can be replaced with `-p`, or both options with `-pp`):

```
bat --style plain --pager never {{path/to/file}}
```

- Highlight a specific line or a range of lines with a different background color:

```
bat {{[-H|--highlight-line]}} {{10|5:10|:10|10:|10:+5}} {{path/to/file}}
```

- Show non-printable characters like space, tab or newline:

```
bat {{[-A|--show-all]}} {{path/to/file}}
```

- Remove all decorations except line numbers in the output:

```
bat {{[-n|--number]}} {{path/to/file}}
```

- Syntax highlight a JSON file by explicitly setting the language:

```
bat {{[-l|--language]}} json {{path/to/file.json}}
```

- Display all supported languages:

```
bat {{[-L|--list-languages]}}
```

batch

Execute commands at a later time when the system load levels permit.

Results will be sent to the user's mail.

See also: **`at`**, **`atq`**, **`atrm`** **`mail`**.

More information: https://manned.org/batch.

- Execute commands from `stdin` (press `<Ctrl d>` when done):

  ```
  batch
  ```

- Execute a command from `stdin`:

  ```
  echo "{{./make_db_backup.sh}}" | batch
  ```

bats

Bash Automated Testing System: a TAP (https://testanything.org/) compliant testing framework for Bash.

More information: https://bats-core.readthedocs.io/en/stable/usage.html.

- Run a BATS test script and output results in the TAP (Test Anything Protocol) format:

```
bats {{[-t|--tap]}} {{path/to/test.bats}}
```

- Count test cases of a test script without running any tests:

```
bats {{[-c|--count]}} {{path/to/test.bats}}
```

- Run BATS test cases recursively (files with a `.bats` extension):

```
bats {{[-r|--recursive]}} {{path/to/directory}}
```

- Output results in a specific format:

```
bats {{[-F|--formatter]}} {{pretty|tap|tap13|junit}} {{path/to/test.bats}}
```

- Add timing information to tests:

```
bats {{[-T|--timing]}} {{path/to/test.bats}}
```

- Run specific number of jobs in parallel (requires GNU `parallel` to be installed):

```
bats {{[-j|--jobs]}} {{number}} {{path/to/test.bats}}
```

bazel

Open-source build and test tool similar to Make, Maven, and Gradle.

More information: https://bazel.build/reference/command-line-reference.

- Build the specified target in the workspace:

```
bazel build {{target}}
```

- Remove output files and stop the server if running:

```
bazel clean
```

- Stop the bazel server:

```
bazel shutdown
```

- Display runtime info about the bazel server:

```
bazel info
```

- Display help:

```
bazel help
```

- Display version:

```
bazel version
```

bb

Native Clojure interpreter for scripting.

More information: https://book.babashka.org/#usage.

- Evaluate an expression:

```
bb {{[-e|--eval]}} "(+ 1 2 3)"
```

- Evaluate a script file:

```
bb {{[-f|--file]}} {{path/to/script.clj}}
```

- Bind [i]nput to a sequence of lines from `stdin`:

```
printf "first\nsecond" | bb -i "(map clojure.string/
capitalize *input*)"
```

- Bind [I]nput to a sequence of EDN (Extensible Data Notation) values from `stdin`:

```
echo "{:key 'val}" | bb -I "(:key (first *input*))"
```

bc

An arbitrary precision calculator language.

See also: **`dc`**, **`qalc`**.

More information: https://manned.org/bc.

- Start an interactive session:

```
bc
```

- Start an interactive session with the standard math library enabled:

```
bc {{[-i|--interactive]}} {{[-l|--mathlib]}}
```

- Calculate an expression:

```
echo '{{5 / 3}}' | bc
```

- Execute a script:

```
bc {{path/to/script.bc}}
```

- Calculate an expression with the specified scale:

```
echo 'scale = {{10}}; {{5 / 3}}' | bc
```

- Calculate a sine/cosine/arctangent/natural logarithm/exponential function using `mathlib`:

```
echo '{{s|c|a|l|e}}({{1}})' | bc {{[-l|--mathlib]}}
```

- Execute an inline factorial script:

```
echo "define factorial(n) { if (n <= 1) return 1; return
n*factorial(n-1); }; factorial({{10}})" | bc
```

bcomps

Decompose graphs into their biconnected components.

Graphviz filters: **`acyclic`**, **`bcomps`**, **`comps`**, **`edgepaint`**, **`gvcolor`**, **`gvpack`**, **`mingle`**, **`nop`**, **`sccmap`**, **`tred`**, & **`unflatten`**.

More information: https://graphviz.org/pdf/bcomps.1.pdf.

- Decompose one or more graphs into their biconnected components:

```
bcomps {{path/to/input1.gv path/to/input2.gv ...}} > {{path/to/output.gv}}
```

- Print the number of blocks and cutvertices in one or more graphs:

```
bcomps -v -s {{path/to/input1.gv path/to/input2.gv ...}}
```

- Write each block and block-cutvertex tree to multiple numbered filenames based on `output.gv`:

```
bcomps -x -o {{path/to/output.gv}} {{path/to/input1.gv path/to/input2.gv ...}}
```

- Display help:

```
bcomps -?
```

bdfr

Bulk downloader for Reddit.

More information: https://github.com/Serene-Arc/bulk-downloader-for-reddit.

- Download videos/images from the specified links to URL or ID's of posts:

```
bdfr download {{path/to/output_directory}} {{[-l|--link]}}
{{post_url}}
```

- Download the maximum possible number (roughly 1000) of videos/images from a specified user:

```
bdfr download {{path/to/output_directory}} {{[-u|--user]}}
{{reddit_user}} --submitted
```

- Download submission data (text, upvotes, comments, etc.) limited to 10 submissions for each subreddit (30 total):

```
bdfr archive {{path/to/output_directory}} {{[-s|--
subreddit]}} '{{Python, all, mindustry}}' {{[-L|--limit]}} 10
```

- Download videos/images from the subreddit r/Python sorted by top (default is hot) using time filter all, limited to 10 submissions:

```
bdfr download {{path/to/output_directory}} {{[-s|--
subreddit]}} Python {{[-S|--sort]}} top {{[-t|--time]}} all
{{[-L|--limit]}} 10
```

- Download the maximum possible number of both submission data and videos/images from subreddit r/Python skipping over submissions with mp4 or gif file extensions and creating hard links for duplicate files:

```
bdfr clone {{path/to/output_directory}} {{[-s|--subreddit]}}
Python --skip mp4 --skip gif --make-hard-links
```

- Download saved posts of the authenticated user, naming each file according to a specified format. Avoid downloading duplicates and posts already present in the output directory:

```
bdfr download {{path/to/output_directory}} {{[-u|--user]}} me
--saved --authenticate --file-scheme '{{ {POSTID}_{TITLE}
_{UPVOTES} }}' --no-dupes --search-existing
```

beanstalkd

A simple and generic work-queue server.

More information: https://manned.org/beanstalkd.

- Start the server, listening on port 11300:

```
beanstalkd
```

- [l]isten on a specific [p]ort and address:

```
beanstalkd -l {{ip_address}} -p {{port_number}}
```

- Persist work queues by saving them to disk:

```
beanstalkd -b {{path/to/persistence_directory}}
```

- Sync to the persistence directory every 500 milliseconds:

```
beanstalkd -b {{path/to/persistence_directory}} -f {{500}}
```

bear

A tool to generate compilation databases for **`clang`** tooling.

More information: https://github.com/rizsotto/Bear.

- Generate `compile_commands.json` by running a build command:

```
bear -- {{make}}
```

- Generate compilation database with a custom output file name:

```
bear --output {{path/to/compile_commands.json}} -- {{make}}
```

- Append results to an existing `compile_commands.json` file:

```
bear --append -- {{make}}
```

- Run in verbose mode to get detailed output:

```
bear --verbose -- {{make}}
```

- Force `bear` to use the preload method for command interception:

```
bear --force-preload -- {{make}}
```

bedtools

A swiss-army knife of tools for genomic-analysis tasks.

Used to intersect, group, convert and count data in BAM, BED, GFF/GTF, VCF format.

More information: https://bedtools.readthedocs.io/en/latest/content/overview.html#summary-of-available-tools.

- Intersect file [a] and file(s) [b] regarding the sequences' [s]trand and save the result to a specific file:

```
bedtools intersect -a {{path/to/file_A}} -b {{path/to/file_B1 path/to/file_B2 ...}} -s > {{path/to/output_file}}
```

- Intersect two files with a [l]eft [o]uter [j]oin, i.e. report each feature from `file1` and NULL if no overlap with `file2`:

```
bedtools intersect -a {{path/to/file1}} -b {{path/to/file2}} -loj > {{path/to/output_file}}
```

- Using more efficient algorithm to intersect two pre-sorted files:

```
bedtools intersect -a {{path/to/file1}} -b {{path/to/file2}} -sorted > {{path/to/output_file}}
```

- [g]roup a file based on the first three and the fifth [c]olumn and apply the sum [o]peration on the sixth column:

```
bedtools groupby -i {{path/to/file}} -c 1-3,5 -g 6 -o sum
```

- Convert bam-formatted [i]nput file to a bed-formatted one:

```
bedtools bamtobed -i {{path/to/file.bam}} > {{path/to/file.bed}}
```

- Find for all features in `file1.bed` the closest one in `file2.bed` and write their [d]istance in an extra column (input files must be sorted):

```
bedtools closest -a {{path/to/file1.bed}} -b {{path/to/file2.bed}} -d
```

behat

A PHP framework for Behaviour-Driven Development.

More information: https://docs.behat.org/en/latest/user_guide/command_line_tool.html.

- Initialize a new Behat project:

  ```
  behat --init
  ```

- Run all tests:

  ```
  behat
  ```

- Run all tests from the specified suite:

  ```
  behat --suite {{suite_name}}
  ```

- Run tests with a specific output formatter:

  ```
  behat --format {{pretty|progress}}
  ```

- Run tests and output results to a file:

  ```
  behat --out {{path/to/file}}
  ```

- List the definitions in your test suites:

  ```
  behat --definitions
  ```

berks

Chef cookbook dependency manager.

More information: https://docs.chef.io/berkshelf.html.

- Install cookbook dependencies into a local repo:

```
berks install
```

- Update a specific cookbook and its dependencies:

```
berks update {{cookbook}}
```

- Upload a cookbook to the Chef server:

```
berks upload {{cookbook}}
```

- View the dependencies of a cookbook:

```
berks contingent {{cookbook}}
```

betty

Use natural language to execute commands.

More information: https://github.com/pickhardt/betty.

- Ask Betty something:

```
betty {{what time is it}}
```

- Download a file:

```
betty download {{https://example.com/file.ext}} to {{path/to/output_file.ext}}
```

- Compress a file or directory to one of the support archive formats:

```
betty {{zip}} {{path/to/file_or_directory}}
```

- Extract an archive into the current directory:

```
betty {{unzip}} {{archive.tar.gz}}
```

- Extract an archive into a specific directory:

```
betty unarchive {{archive.tar.gz}} to {{path/to/directory}}
```

- Play Spotify:

```
betty play {{Spotify}}
```

- Drive Betty to madness:

```
betty go crazy
```

- Display version:

```
betty version
```

bfg

Remove large files or passwords from Git history like git-filter-branch.

Note: If your repository is connected to a remote, you will need to force push to it.

More information: https://rtyley.github.io/bfg-repo-cleaner/.

- Remove a file with sensitive data but leave the latest commit untouched:

```
bfg --delete-files {{file_with_sensitive_data}}
```

- Remove all text mentioned in the specified file wherever it can be found in the repository's history:

```
bfg --replace-text {{path/to/file.txt}}
```

bfs

Breadth-first search for your files.

More information: https://manned.org/bfs.

- Find files by extension:

```
bfs {{root_path}} -name '{{*.ext}}'
```

- Find files matching multiple path/name patterns:

```
bfs {{root_path}} -path '{{**/path/**/*.ext}}' -or -name
'{{*pattern*}}'
```

- Find directories matching a given name, in case-insensitive mode:

```
bfs {{root_path}} -type d -iname '{{*lib*}}'
```

- Find files matching a given pattern, excluding specific paths:

```
bfs {{root_path}} -name '{{*.py}}' -not -path '{{*/site-
packages/*}}'
```

- Find files matching a given size range, limiting the recursive depth to "1":

```
bfs {{root_path}} -maxdepth 1 -size {{+500k}} -size {{-10M}}
```

- Run a command for each file (use `{}` within the command to access the filename):

```
bfs {{root_path}} -name '{{*.ext}}' -exec {{wc -l}} {} \;
```

- Find all files modified today and pass the results to a single command as arguments:

```
bfs {{root_path}} -daystart -mtime {{-1}} -exec {{tar -cvf
archive.tar}} {} \+
```

- Find empty files (0 byte) or directories and delete them verbosely:

```
bfs {{root_path}} -type {{f|d}} -empty -delete -print
```

bg

Resume suspended jobs (e.g. using **`<Ctrl z>`**), and keeps them running in the background.

See also: **`jobs`**, **`fg`**, **`disown`**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-bg.

- Resume the most recently suspended job and run it in the background:

```
bg
```

- Resume a specific job and run it in the background (run `jobs` to find the job number):

```
bg %{{job_number}}
```

bgpgrep

Filter and print BGP data within MRT dumps.

Can read files compressed with **`gzip`**, **`bzip2`** and **`xz`**.

More information: https://codeberg.org/1414codeforge/ubgpsuite.

- List all routes:

```
bgpgrep {{master6.mrt}}
```

- List routes received from a specific peer, determined by the peer's AS number:

```
bgpgrep {{master4.mrt}} -peer {{64498}}
```

- List routes received from a specific peer, determined by the peer's IP address:

```
bgpgrep {{master4.mrt.bz2}} -peer {{2001:db8:dead:cafe:acd::19e}}
```

- List routes which have certain ASNs in their AS path:

```
bgpgrep {{master6.mrt.bz2}} -aspath '{{64498 64510}}'
```

- List routes that lead to a specific address:

```
bgpgrep {{master6.mrt.bz2}} -supernet '{{2001:db8:dead:cafe:aef::5}}'
```

- List routes that have communities from a specific AS:

```
bgpgrep {{master4.mrt}} -communities \( '{{64497}}:*' \)
```

biff

A simple utility for doing datetime arithmetic, parsing and formatting.

More information: https://github.com/burntsushi/biff.

- Print the current time in a format of your choosing:

```
biff time fmt {{[-f|--format]}} rfc3339 now
```

- Print multiple relative times in one command:

```
biff time fmt {{[-f|--format]}} '%c' now -1d 'next sat' 'last monday' '9pm last mon'
```

- Print the current time in another time zone, and round it the nearest 15 minute increment:

```
biff time in Asia/Bangkok now | biff time round {{[-i|--increment]}} 15 {{[-s|--smallest]}} minute
```

- Convert a time between two different time zone:

```
TZ='Japan' biff time in America/New_York 02:30
```

- Print a past or future time relative to current time:

```
biff time add {{-1d|1d|1w|-1m|1y|...}} now
```

- Add a complex duration to the current time:

```
biff time add '1 week, 12 hours ago' now
```

- Find the duration since a date in the past and round it to the desired precision:

```
biff span since 2025-01-20T12:00 {{[-l|--largest]}} year
```

- Find timestamps in a log file and reformat them into your local time in place:

```
biff tag lines /tmp/access.log | biff time in system | biff time fmt {{[-f|--format]}} '%c' | head {{[-n|--lines]}} 3 | biff untag {{[-s|--substitute]}}
```

bind

Bash builtin to manage bash hotkeys and variables.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-bind.

- List all bound commands and their hotkeys:

```
bind {{-p|-P}}
```

- Query a command for its hotkey:

```
bind -q {{command}}
```

- Bind a key:

```
bind -x '"{{key_sequence}}":{{command}}'
```

- List user defined bindings:

```
bind -X
```

- Display help:

```
help bind
```

bindkey

Add keybindings to Z-Shell.

More information: https://zsh.sourceforge.io/Guide/zshguide04.html.

- Bind a hotkey to a specific command:

```
bindkey "{{^k}}" {{kill-line}}
```

- Bind a hotkey to a specific key [s]equence:

```
bindkey -s '^o' 'cd ..\n'
```

- [l]ist keymaps:

```
bindkey -l
```

- View the hotkey in a key[M]ap:

```
bindkey -M main
```

bing-rewards

Automate daily Bing Rewards points.

More information: https://github.com/jack-mil/bing-rewards.

- Complete both desktop and mobile searches:

```
bing-rewards
```

- Run 10 searches with mobile user-agent:

```
bing-rewards {{[-m|--mobile]}} {{[-c|--count]}} 10
```

- Complete mobile and desktop searches using specified Chrome profile "Profile 1":

```
bing-rewards --profile "Profile 1"
```

- Run searches sequentially across multiple Chrome profiles:

```
bing-rewards --profile "Default" "Profile 1" "Profile 2"
```

- Display help:

```
bing-rewards {{[-h|--help]}}
```

binwalk

Firmware Analysis Tool.

More information: https://github.com/ReFirmLabs/binwalk.

- Scan a binary file:

```
binwalk {{path/to/binary}}
```

- Extract files from a binary, specifying the output directory:

```
binwalk {{[-e|--extract]}} {{[-C|--directory]}}
{{output_directory}} {{path/to/binary}}
```

- Recursively extract files from a binary limiting the recursion depth to 2:

```
binwalk {{[-e|--extract]}} {{[-M|--matryoshka]}} {{[-d|--
depth]}} {{2}} {{path/to/binary}}
```

- Extract files from a binary with the specified file signature:

```
binwalk {{[-D|--dd]}} '{{png image:png}}' {{path/to/binary}}
```

- Analyze the entropy of a binary, saving the plot with the same name as the binary and `.png` extension appended:

```
binwalk {{[-E|--entropy]}} {{[-J|--save]}} {{path/to/binary}}
```

- Combine entropy, signature and opcodes analysis in a single command:

```
binwalk {{[-E|--entropy]}} {{[-B|--signature]}} {{[-A|--
opcodes]}} {{path/to/binary}}
```

bioradtopgm

Convert a Biorad confocal file into a PGM file.

More information: https://netpbm.sourceforge.net/doc/bioradtopgm.html.

- Read a Biorad confocal file and store the n'th image contained in it to as a PGM file:

```
bioradtopgm -{{n}} {{path/to/file.pic}} > {{path/to/file.pgm}}
```

- Read a Biorad confocal file and print the number of images it contains:

```
bioradtopgm {{path/to/file.pic}}
```

- Display version:

```
bioradtopgm {{[-v|-version]}}
```

bird

BIRD Internet Routing Daemon.

Routing daemon with support for BGP, OSPF, Babel and others.

More information: https://bird.network.cz/?get_doc&v=30&f=bird-1.html#ss1.3.

- Start Bird with a specific configuration file:

```
bird -c {{path/to/bird.conf}}
```

- Start Bird as a specific user and group:

```
bird -u {{username}} -g {{group}}
```

birdc

BIRD remote control.

Retrieve information like routes from bird and perform configurations during runtime.

More information: https://bird.network.cz/?get_doc&v=30&f=bird-4.html.

- Open the remote control console:

```
birdc
```

- Reload the configuration without restarting BIRD:

```
birdc configure
```

- Show the current status of BIRD:

```
birdc show status
```

- Show all configured protocols:

```
birdc show protocols
```

- Show all details about a protocol:

```
birdc show protocols {{upstream1}} all
```

- Show all routes that contain a specific AS number:

```
birdc "show route where bgp_path ~ [{{4242120045}}]"
```

- Show all best routes:

```
birdc show route primary
```

- Show all details of all routes from a given prefix:

```
birdc show route for {{fd00:/8}} all
```

bison

GNU parser generator.

More information: https://manned.org/bison.

- Compile a bison definition file:

```
bison {{path/to/file.y}}
```

- Compile in debug mode, which causes the resulting parser to write additional information to `stdout`:

```
bison {{[-t|--debug]}} {{path/to/file.y}}
```

- Specify the output filename:

```
bison {{[-o|--output]}} {{path/to/output.c}} {{path/to/file.y}}
```

- Be verbose when compiling:

```
bison {{[-v|--verbose]}}
```

bitcoin-cli

Client to interact with the Bitcoin Core daemon via RPC calls.

Uses the configuration defined in **`bitcoin.conf`**.

More information: https://en.bitcoin.it/wiki/Running_Bitcoin#Command-line_arguments.

- Send a transaction to a given address:

```
bitcoin-cli sendtoaddress "{{address}}" {{amount}}
```

- Generate one or more blocks:

```
bitcoin-cli generate {{num_blocks}}
```

- Print high-level information about the wallet:

```
bitcoin-cli getwalletinfo
```

- List all outputs from previous transactions available to fund outgoing transactions:

```
bitcoin-cli listunspent
```

- Export the wallet information to a text file:

```
bitcoin-cli dumpwallet "{{path/to/file}}"
```

- Get blockchain information:

```
bitcoin-cli getblockchaininfo
```

- Get network information:

```
bitcoin-cli getnetworkinfo
```

- Stop the Bitcoin Core daemon:

```
bitcoin-cli stop
```

bitcoind

Bitcoin Core daemon.

Uses the configuration defined in **`bitcoin.conf`**.

More information: https://manned.org/bitcoind.

- Start the Bitcoin Core daemon (in the foreground):

```
bitcoind
```

- Start the Bitcoin Core daemon in the background (use `bitcoin-cli stop` to stop):

```
bitcoind -daemon
```

- Start the Bitcoin Core daemon on a specific network:

```
bitcoind -chain={{main|test|signet|regtest}}
```

- Start the Bitcoin Core daemon using specific configuration file and data directory:

```
bitcoind -conf={{path/to/bitcoin.conf}} -datadir={{path/to/directory}}
```

black

Format Python code automatically.

More information: https://black.readthedocs.io/en/stable/usage_and_configuration/the_basics.html.

- Auto-format a file or entire directory:

```
black {{path/to/file_or_directory}}
```

- Format the code passed in as a string:

```
black {{[-c|--code]}} "{{code}}"
```

- Show whether a file or a directory would have changes made to them if they were to be formatted:

```
black --check {{path/to/file_or_directory}}
```

- Show changes that would be made to a file or a directory without performing them (dry-run):

```
black --diff {{path/to/file_or_directory}}
```

- Auto-format a file or directory, emitting exclusively error messages to `stderr`:

```
black {{[-q|--quiet]}} {{path/to/file_or_directory}}
```

- Auto-format a file or directory without replacing single quotes with double quotes (adoption helper, avoid using this for new projects):

```
black {{[-S|--skip-string-normalization]}} {{path/to/file_or_directory}}
```

blackfire

Monitor, profile and test a PHP application.

More information: https://blackfire.io.

- Initialize and configure the Blackfire client:

```
blackfire config
```

- Launch the Blackfire agent:

```
blackfire agent
```

- Launch the Blackfire agent on a specific socket:

```
blackfire agent --socket="{{tcp://127.0.0.1:8307}}"
```

- Run the profiler on a specific program:

```
blackfire run {{php path/to/file.php}}
```

- Run the profiler and collect 10 samples:

```
blackfire --samples 10 run {{php path/to/file.php}}
```

- Run the profiler and output results as JSON:

```
blackfire --json run {{php path/to/file.php}}
```

- Upload a profiler file to the Blackfire web service:

```
blackfire upload {{path/to/file}}
```

- View the status of profiles on the Blackfire web service:

```
blackfire status
```

blahaj

A lolcat-like output colorizer that also prints flags and colorful sharks.

More information: https://codeberg.org/GeopJr/BLAHAJ.

- Get a list of possible flags/colors:

```
blahaj --flags
```

- Print a shark (blahaj) with default trans colors:

```
blahaj {{[-s|--shark]}}
```

- Print a random flag with a 2x size multiplier:

```
blahaj {{[-f|--flag]}} {{[-r|--random]}} {{[-m|--
multiplier]}} 2
```

- Print the result of a text-producing command with lesbian colors:

```
{{cowsay "Hello, world"}} | blahaj {{[-c|--colors]}} lesbian
```

- Print text and color by individual character:

```
echo "{{Hello, world}}" | blahaj {{[-i|--individual]}}
```

- Print contents of a text document, coloring the background instead of text, by word:

```
blahaj {{[-w|--words]}} {{[-b|--background]}} {{path/to/
file}}
```

blender

Command-line interface to the Blender 3D computer graphics application.

Arguments are executed in the order they are given.

More information: https://docs.blender.org/manual/en/latest/advanced/command_line/arguments.html.

- Render all frames of an animation in the background, without loading the UI (output is saved to `/tmp`):

```
blender {{[-b|--background]}} {{path/to/file.blend}} {{[-a|--render-anim]}}
```

- Render an animation using a specific image naming pattern, in a path relative (`//`) to the .blend file:

```
blender {{[-b|--background]}} {{path/to/file.blend}} {{[-o|--render-output]}} //{{render/frame_###.png}} {{[-a|--render-anim]}}
```

- Render the 10th frame of an animation as a single image, saved to an existing directory (absolute path):

```
blender {{[-b|--background]}} {{path/to/file.blend}} {{[-o|--render-output]}} /{{path/to/output_directory}} {{[-f|--render-frame]}} {{10}}
```

- Render the second last frame in an animation as a JPEG image, saved to an existing directory (relative path):

```
blender {{[-b|--background]}} {{path/to/file.blend}} {{[-o|--render-output]}} //{{output_directory}} {{[-f|--render-frame]}} {{JPEG}} {{[-f|--render-frame]}} {{-2}}
```

- Render the animation of a specific scene, starting at frame 10 and ending at frame 500:

```
blender {{[-b|--background]}} {{path/to/file.blend}} {{[-S|--scene]}} {{scene_name}} {{[-s|--frame-start]}} {{10}} {{[-e|--frame-end]}} {{500}} {{[-a|--render-anim]}}
```

- Render an animation at a specific resolution, by passing a Python expression:

```
blender {{[-b|--background]}} {{path/to/file.blend}} --python-expr '{{import bpy;
```

```
bpy.data.scenes[0].render.resolution_percentage = 25}}' {{[-
a|--render-anim]}}
```

- Start an interactive Blender session in the terminal with a Python console (do `import bpy` after starting):

  ```
  blender {{[-b|--background]}} --python-console
  ```

blockout2

Tetris like game in 3D.

More information: https://www.blockout.net/blockout2/.

- Start a new game:

  ```
  blockout2
  ```

- Navigate the current piece on a 2D plane:

  ```
  {{<ArrowUp>|<ArrowDown>|<ArrowLeft>|<ArrowRight>}}
  ```

- Rotate the piece on its axis:

  ```
  {{<q>|<w>|<e>|<a>|<s>|<d>}}
  ```

- Hard drop the current piece:

  ```
  <Space>
  ```

- Pause/unpause the game:

  ```
  <p>
  ```

bloodhound-python

A Python ingestor for BloodHound, used to enumerate Active Directory relationships.

More information: https://github.com/dirkjanm/BloodHound.py.

- Collect all data using default collection methods (includes groups, sessions, and trusts):

```
bloodhound-python --username {{username}} --password {{password}} --domain {{domain}}
```

- Collect data using Kerberos authentication without requiring a plaintext password:

```
bloodhound-python --collectionmethod {{All}} --kerberos --domain {{domain}}
```

- Authenticate using NTLM hashes instead of a password:

```
bloodhound-python --collectionmethod {{All}} --username {{username}} --hashes {{LM:NTLM}} --domain {{domain}}
```

- Specify a custom name server for DNS queries:

```
bloodhound-python --collectionmethod {{All}} --username {{username}} --password {{password}} --domain {{domain}} --nameserver {{nameserver}}
```

- Save the output files as a compressed ZIP archive:

```
bloodhound-python --collectionmethod {{All}} --username {{username}} --password {{password}} --domain {{domain}} --zip
```

bmaptool

Create or copy block maps intelligently (designed to be faster than **cp** or **dd**).

More information: https://manned.org/bmaptool.

- Output a blockmap file from image file:

```
bmaptool create {{[-o|--output]}} {{blockmap.bmap}} {{source.img}}
```

- Copy an image file into sdb:

```
bmaptool copy --bmap {{blockmap.bmap}} {{source.img}} {{/dev/sdb}}
```

- Copy a compressed image file into sdb:

```
bmaptool copy --bmap {{blockmap.bmap}} {{source.img.gz}} {{/dev/sdb}}
```

- Copy an image file into sdb without using a blockmap:

```
bmaptool copy --nobmap {{source.img}} {{/dev/sdb}}
```

bmptopnm

Convert a BMP file into a PBM, PGM, or PNM image.

More information: https://netpbm.sourceforge.net/doc/bmptopnm.html.

- Generate the PBM, PGM, or PNM image as output, for Windows or OS/2 BMP file as input:

```
bmptopnm {{path/to/file.bmp}}
```

- Report contents of the BMP header to `stderr`:

```
bmptopnm {{[-verb|-verbose]}} {{path/to/file.bmp}}
```

- Display version:

```
bmptopnm {{[-v|-version]}}
```

bmptoppm

This command has been superseded by **bmptopnm**.

More information: https://netpbm.sourceforge.net/doc/bmptoppm.html.

- View documentation for the current command:

```
tldr bmptopnm
```

bob

Manage and switch between Neovim versions.

More information: https://github.com/MordechaiHadad/bob.

- Install and switch to the specified version of Neovim:

```
bob use {{nightly|stable|latest|version_string|commit_hash}}
```

- List installed and currently used versions of Neovim:

```
bob list
```

- Uninstall the specified version of Neovim:

```
bob uninstall {{nightly|stable|latest|version_string|
commit_hash}}
```

- Uninstall Neovim and erase any changes `bob` has made:

```
bob erase
```

- Roll back to a previous nightly version:

```
bob rollback
```

boot

Build tooling for the Clojure programming language.

More information: https://github.com/boot-clj/boot.

- Start a REPL session either with the project or standalone:

  ```
  boot repl
  ```

- Build a single `uberjar`:

  ```
  boot jar
  ```

- Generate scaffolding for a new project based on a template:

  ```
  boot --dependencies boot/new new --template {{template_name}}
  --name {{project_name}}
  ```

- Build for development (if using the boot/new template):

  ```
  boot dev
  ```

- Build for production (if using the boot/new template):

  ```
  boot prod
  ```

- Display help for a specific task:

  ```
  boot {{task}} --help
  ```

borg

Deduplicating backup tool.

Create local or remote backups that are mountable as filesystems.

More information: https://borgbackup.readthedocs.io/en/stable/usage/general.html.

- Initialize a (local) repository:

```
borg init {{path/to/repo_directory}}
```

- Backup a directory into the repository, creating an archive called "Monday":

```
borg create --progress {{path/to/repo_directory}}::{{Monday}} {{path/to/source_directory}}
```

- List all archives in a repository:

```
borg list {{path/to/repo_directory}}
```

- Extract a specific directory from the "Monday" archive in a remote repository, excluding all `*.ext` files:

```
borg extract {{user}}@{{host}}:{{path/to/repo_directory}}::{{Monday}} {{path/to/target_directory}} --exclude '{{*.ext}}'
```

- Prune a repository by deleting all archives older than 7 days, listing changes:

```
borg prune --keep-within {{7d}} --list {{path/to/repo_directory}}
```

- Mount a repository as a FUSE filesystem:

```
borg mount {{path/to/repo_directory}}::{{Monday}} {{path/to/mountpoint}}
```

- Display help on creating archives:

```
borg create --help
```

bosh

Deploy and manage the BOSH director.

More information: https://bosh.io/docs/cli-v2/.

- Create local alias for director in a specific environment:

```
bosh alias-env {{environment_name}} {{[-e|--environment]}} {{ip_address|URL}} --ca-cert {{ca_certificate}}
```

- List environments:

```
bosh environments
```

- Log in to the director:

```
bosh login {{[-e|--environment]}} {{environment}}
```

- List deployments:

```
bosh {{[-e|--environment]}} {{environment}} deployments
```

- List environment virtual machines in a deployment:

```
bosh {{[-e|--environment]}} {{environment}} vms {{[-d|--deployment]}} {{deployment}}
```

- SSH into virtual machine:

```
bosh {{[-e|--environment]}} {{environment}} ssh {{virtual_machine}} {{[-d|--deployment]}} {{deployment}}
```

- Upload stemcell:

```
bosh {{[-e|--environment]}} {{environment}} upload-stemcell {{stemcell_file|url}}
```

- Show current cloud config:

```
bosh {{[-e|--environment]}} {{environment}} cloud-config
```

bower

A package manager optimized for front-end web development.

A package can be a GitHub user/repo shorthand, a Git endpoint, a URL or a registered package.

More information: https://bower.io/#getting-started.

- Install a project's dependencies, listed in its bower.json:

  ```
  bower install
  ```

- Install one or more packages to the bower_components directory:

  ```
  bower install {{package}} {{package}}
  ```

- Uninstall packages locally from the bower_components directory:

  ```
  bower uninstall {{package}} {{package}}
  ```

- List local packages and possible updates:

  ```
  bower list
  ```

- Create a `bower.json` file for your package:

  ```
  bower init
  ```

- Install a specific dependency version, and add it to `bower.json`:

  ```
  bower install {{local_name}}={{package}}#{{version}} --save
  ```

- Display help for a specific command:

  ```
  bower help {{command}}
  ```

box

A PHP application for building and managing Phars.

More information: https://github.com/box-project/box.

- Compile a new Phar file:

```
box compile
```

- Compile a new Phar file using a specific configuration file:

```
box compile {{[-c|--config]}} {{path/to/config}}
```

- Display information about the PHAR PHP extension:

```
box info
```

- Display information about a specific Phar file:

```
box info {{path/to/phar_file}}
```

- Validate the first found configuration file in the working directory:

```
box validate
```

- Verify the signature of a specific Phar file:

```
box verify {{path/to/phar_file}}
```

- Display help:

```
box help
```

boxes

Draw, remove, and repair ASCII art boxes.

More information: https://boxes.thomasjensen.com/boxes-man-1.html.

- Draw a box around a string:

```
echo "{{string}}" | boxes
```

- Remove a box from a string:

```
echo "{{string}}" | boxes {{[-r|--remove]}}
```

- Specify the box design:

```
echo "{{string}}" | boxes {{[-d|--design]}} {{parchment}}
```

- Specify the box size (in columns by lines):

```
echo "{{string}}" | boxes {{[-s|--size]}} {{10}}x{{5}}
```

- Align the box text [h]orizonally (at [l]eft, [c]enter or [r]ight):

```
echo "{{string}}" | boxes {{[-a|--align]}} h{{l|c|r}}
```

- Align the box text [v]ertically (at [t]op, [c]enter or [b]ottom):

```
echo "{{string}}" | boxes {{[-a|--align]}} v{{t|c|b}}
```

- [j]ustify the box text (at [l]eft, [c]enter or [r]ight):

```
echo "{{string}}" | boxes {{[-a|--align]}} j{{l|c|r}}{{vt}}
```

boxxy

Redirect file locations for programs that don't respect the XDG standards.

More information: https://github.com/queer/boxxy.

- Run a program with file redirections defined in `~/.config/boxxy/boxxy.yaml`:

```
boxxy {{program}}
```

- Scan your home directory for rule suggestions:

```
boxxy scan
```

- Trace what files a program touches and save a report in `boxxy-report.txt` in the current directory:

```
boxxy {{[-t|--trace]}} {{program}}
```

- Pass a redirection rule directly on the terminal:

```
boxxy {{[-r|--rule]}} {{path/to/file_or_directory}}:{{path/to/redirection}}:{{file|directory}} {{program}}
```

- View the config file:

```
boxxy config
```

- Display help:

```
boxxy -h
```

bpkg

A package manager for Bash scripts.

More information: https://github.com/bpkg/bpkg.

- Update the local index:

```
bpkg update
```

- Install a package globally:

```
bpkg install --global {{package}}
```

- Install a package in a subdirectory of the current directory:

```
bpkg install {{package}}
```

- Install a specific version of a package globally:

```
bpkg install {{package}}@{{version}} -g
```

- Show details about a specific package:

```
bpkg show {{package}}
```

- Run a command, optionally specifying its arguments:

```
bpkg run {{command}} {{argument1 argument2 ...}}
```

bpython

A fancy interface to the Python interpreter.

Provides syntax highlighting and many other nice-to-haves in REPL mode.

More information: https://manned.org/bpython.

- Start a REPL (interactive shell):

```
bpython
```

- Execute a specific Python file:

```
bpython {{path/to/file.py}}
```

- Execute a specific Python file and start a REPL:

```
bpython {{[-i|--interactive]}} {{path/to/file.py}}
```

- Use the specified configuration file instead of the default configuration:

```
bpython --config {{path/to/file.conf}}
```

bpytop

A resource monitor that shows information about the CPU, memory, disks, network and processes.

A Python version of **`bashtop`**.

More information: https://github.com/aristocratos/bpytop.

- Start `bpytop`:

  ```
  bpytop
  ```

- Start in minimal mode without memory and networking boxes:

  ```
  bpytop -m
  ```

- Toggle minimal mode:

  ```
  <m>
  ```

- Search for running programs or processes:

  ```
  <f>
  ```

- Change settings:

  ```
  <M>
  ```

- Display version:

  ```
  bpytop -v
  ```

bq

A Python-based tool for BigQuery, Google Cloud's fully managed and completely serverless enterprise data warehouse.

More information: https://cloud.google.com/bigquery/docs/reference/bq-cli-reference.

- Run query against a BigQuery table using standard SQL, add `--dry_run` flag to estimate the number of bytes read by the query:

```
bq query --nouse_legacy_sql 'SELECT COUNT(*) FROM {{DATASET_NAME}}.{{TABLE_NAME}}'
```

- Run a parameterized query:

```
bq query --use_legacy_sql=false --parameter='ts_value:TIMESTAMP:2016-12-07 08:00:00' 'SELECT TIMESTAMP_ADD(@ts_value, INTERVAL 1 HOUR)'
```

- Create a new dataset or table in the US location:

```
bq mk --location=US {{dataset_name}}.{{table_name}}
```

- List all datasets in a project:

```
bq ls --filter labels.{{key}}:{{value}} --max_results {{integer}} --format=prettyjson --project_id {{project_id}}
```

- Batch load data from a specific file in formats such as CSV, JSON, Parquet, and Avro to a table:

```
bq load --location {{location}} --source_format {{CSV|JSON|PARQUET|AVRO}} {{dataset}}.{{table}} {{path_to_source}}
```

- Copy one table to another:

```
bq cp {{dataset}}.{{OLD_TABLE}} {{dataset}}.{{new_table}}
```

- Display help:

```
bq help
```

brave

This command is an alias of **`chromium`**.

More information: https://support.brave.app/hc/en-us/articles/360044860011-How-Do-I-Use-Command-Line-Flags-in-Brave.

- View documentation for the original command:

```
tldr chromium
```

break

Break out of a **`for`**, **`while`**, **`until`** or **`select`** loop.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-break.

- Break out of a single loop:

```
while :; do break; done
```

- Break out of nested loops:

```
while :; do while :; do break 2; done; done
```

brew abv

This command is an alias of **`brew info`**.

- View documentation for the original command:

  ```
  tldr brew info
  ```

brew autoremove

Remove unused formulae previously installed as dependencies.

More information: https://docs.brew.sh/Manpage#autoremove---dry-run.

- Remove all unused formulae:

```
brew autoremove
```

- Print what would be removed, but don't actually remove anything:

```
brew autoremove {{[-n|--dry-run]}}
```

brew bundle

Bundler for Homebrew, Homebrew Cask and the Mac App Store.

More information: https://docs.brew.sh/Manpage#bundle-subcommand.

- Install packages from a Brewfile at the current path:

```
brew bundle
```

- Install packages from a specific Brewfile at a specific path:

```
brew bundle --file {{path/to/file}}
```

- Create a Brewfile from all installed packages:

```
brew bundle dump
```

- Uninstall all formulae not listed in the Brewfile:

```
brew bundle cleanup --force
```

- Check if there is anything to install or upgrade in the Brewfile:

```
brew bundle check
```

- List all entries in the Brewfile:

```
brew bundle list --all
```

brew cleanup

Remove stale lock files and outdated downloads for all formulas and casks.

More information: https://docs.brew.sh/Manpage#cleanup-options-formulacask-.

- Remove stale lock files and outdated downloads for all formulas/casks:

```
brew cleanup
```

- Remove stale lock files and outdated downloads for a specific formula/cask:

```
brew cleanup {{formula|cask}}
```

- Show what would be removed, but do not actually remove anything:

```
brew cleanup {{[-n|--dry-run]}}
```

- Display help:

```
brew cleanup {{[-h|--help]}}
```

brew info

Display information for Homebrew installation and formulae.

More information: https://docs.brew.sh/Manpage#info-abv-options-formulacask-.

- Display statistics for Homebrew installation:

```
brew info
```

- Display more information about a formula or cask:

```
brew info {{formula|cask}}
```

- Display verbose information about a formula or cask:

```
brew info {{[-v|--verbose]}} {{formula|cask}}
```

- Display information about a formula or cask in JSON format:

```
brew info --json {{formula|cask}}
```

- Print JSON of currently installed formulae:

```
brew info --json --installed
```

- Display help:

```
brew info {{[-h|--help]}}
```

brew install

Install a Homebrew formula or cask.

More information: https://docs.brew.sh/Manpage#install-options-formulacask-.

- Install a formula/cask:

```
brew install {{formula|cask}}
```

- Build and install a formula from source (dependencies will still be installed from bottles):

```
brew install {{[-s|--build-from-source]}} {{formula}}
```

- Download the manifest, print what would be installed but don't actually install anything:

```
brew install {{[-n|--dry-run]}} {{formula|cask}}
```

brew leaves

List installed formulas that are not dependencies of another installed formula or cask.

More information: https://docs.brew.sh/Manpage#leaves---installed-on-request---installed-as-dependency.

- List installed formulas that are not dependent on other installed formulas or casks:

```
brew leaves
```

- Only list leaves that were manually installed:

```
brew leaves {{[-r|--installed-on-request]}}
```

- Only list leaves that were installed as dependencies:

```
brew leaves {{[-p|--installed-as-dependency]}}
```

- Display help:

```
brew leaves {{[-h|--help]}}
```

brew list

List installed formulae/casks or their files.

More information: https://docs.brew.sh/Manpage#list-ls-options-installed_formulainstalled_cask-.

- List all installed formulae and casks:

```
brew {{[ls|list]}}
```

- List files belonging to an installed formula:

```
brew {{[ls|list]}} {{formula}}
```

- List artifacts of a cask:

```
brew {{[ls|list]}} {{cask}}
```

- List only formulae:

```
brew {{[ls|list]}} --formula
```

- List only casks:

```
brew {{[ls|list]}} --cask
```

- List only pinned formulae:

```
brew {{[ls|list]}} --pinned
```

brew outdated

List outdated casks and formulae.

To upgrade everything, use **`brew upgrade`**.

More information: https://docs.brew.sh/Manpage#outdated-options-formulacask-.

- List all outdated casks and formulae:

```
brew outdated
```

- List only outdated formulae:

```
brew outdated --formula
```

- List only outdated casks:

```
brew outdated --cask
```

brew reinstall

Uninstall and then reinstall a formula or cask using the same options it was originally installed with.

More information: https://docs.brew.sh/Manpage#reinstall-options-formulacask-.

- Reinstall a formula/cask:

```
brew reinstall {{formula|cask}}
```

- Reinstall a formula/cask and show verification and post-install steps:

```
brew reinstall {{[-v|--verbose]}} {{formula|cask}}
```

- Make some output more quiet:

```
brew reinstall {{[-q|--quiet]}} {{formula|cask}}
```

- Display help:

```
brew reinstall {{[-h|--help]}}
```

brew remove

This command is an alias of **`brew uninstall`**.

- View documentation for the original command:

```
tldr brew uninstall
```

brew rm

This command is an alias of **`brew uninstall`**.

- View documentation for the original command:

  ```
  tldr brew uninstall
  ```

brew search

Search for casks and formulae.

More information: https://docs.brew.sh/Manpage#search--s-options-textregex-.

- Search for casks and formulae using a keyword:

```
brew search {{keyword}}
```

- Search for casks and formulae using a `regex`:

```
brew search /{{regex}}/
```

- Enable searching through descriptions:

```
brew search --desc {{keyword}}
```

- Only search for formulae:

```
brew search --formula {{keyword}}
```

- Only search for casks:

```
brew search --cask {{keyword}}
```

brew services

Manage background services with **`launchctl`** on macOS or **`systemctl`** on Linux.

More information: https://docs.brew.sh/Manpage#services-subcommand.

- List all managed services for the current user:

```
brew services
```

- List more information about all managed services:

```
brew services info --all
```

- Start a service immediately and register it to launch at login (or boot):

```
brew services start {{formula}}
```

- Stop the service immediately and unregister it from launching at login (or boot):

```
brew services stop {{formula}}
```

- Stop (if necessary) and start the service immediately and register it to launch at login (or boot):

```
brew services restart {{formula}}
```

- Remove all unused services:

```
brew services cleanup
```

brew uninstall

Uninstall a Homebrew formula/cask.

Use **`brew autoremove`** to remove unused dependencies afterwards.

More information: https://docs.brew.sh/Manpage#uninstall-remove-rm-options-installed_formulainstalled_cask-.

- Uninstall a formula/cask:

```
brew {{[rm|uninstall]}} {{formula|cask}}
```

- Uninstall a cask and remove all associated files:

```
brew {{[rm|uninstall]}} --zap {{cask}}
```

brew update

Fetch the newest version of Homebrew and all formulae from GitHub using **`git`** and perform any necessary migrations.

To upgrade all installed formulae, use **`brew upgrade`**.

More information: https://docs.brew.sh/Manpage#update-up-options.

- Fetch the newest version of Homebrew and all formulae:

```
brew {{[up|update]}}
```

brew upgrade

Upgrade outdated formulae and casks.

More information: https://docs.brew.sh/Manpage#upgrade-options-installed_formulainstalled_cask-.

- Upgrade all outdated casks and formulae:

```
brew upgrade
```

- Upgrade a specific formula/cask:

```
brew upgrade {{formula|cask}}
```

- Print what would be upgraded, but don't actually upgrade anything:

```
brew upgrade {{[-n|--dry-run]}}
```

brew

Homebrew - a package manager for macOS and Linux.

Some subcommands such as **`install`** have their own usage documentation.

More information: https://docs.brew.sh/Manpage.

- Install the latest stable version of a formula or cask:

```
brew install {{formula|cask}}
```

- List all installed formulae and casks:

```
brew list
```

- Upgrade an installed formula or cask (if none is given, all installed formulae/casks are upgraded):

```
brew upgrade {{formula|cask}}
```

- Fetch the newest version of Homebrew and of all formulae and casks from the Homebrew source repository:

```
brew update
```

- Show formulae and casks that have a more recent version available:

```
brew outdated
```

- Search for available formulae (i.e. packages) and casks (i.e. native macOS `.app` packages):

```
brew search {{text}}
```

- Display information about a formula or a cask (version, installation path, dependencies, etc.):

```
brew info {{formula|cask}}
```

- Check the local Homebrew installation for potential problems:

```
brew doctor
```

brittany

Pretty-print Haskell source files.

More information: https://github.com/lspitzner/brittany#readme.

- Format a Haskell source file and print the result to `stdout`:

```
brittany {{path/to/file.hs}}
```

- Format all Haskell source files in the current directory in-place:

```
brittany --write-mode=inplace {{*.hs}}
```

- Check whether a Haskell source file needs changes and indicate the result through the programme's exit code:

```
brittany --check-mode {{path/to/file.hs}}
```

- Format a Haskell source file using the specified amount of spaces per indentation level and line length:

```
brittany --indent {{4}} --columns {{100}} {{path/to/file.hs}}
```

- Format a Haskell source file according to the style defined in the specified configuration file:

```
brittany --config-file {{path/to/config.yaml}} {{path/to/file.hs}}
```

brotli

Compress/uncompress files with Brotli compression.

More information: https://manned.org/brotli.

- Compress a file, creating a compressed version next to the file:

```
brotli {{path/to/file}}
```

- Decompress a file, creating an uncompressed version next to the file:

```
brotli {{[-d|--decompress]}} {{path/to/file.br}}
```

- Compress a file specifying the output filename:

```
brotli {{path/to/file}} {{[-o|--output]}} {{path/to/
compressed_output_file.br}}
```

- Decompress a Brotli file specifying the output filename:

```
brotli {{[-d|--decompress]}} {{path/to/compressed_file.br}}
{{[-o|--output]}} {{path/to/output_file}}
```

- Specify the compression quality (1=fastest (worst), 11=slowest (best)):

```
brotli {{[-q|--quality]}} {{11}} {{path/to/file}} {{[-o|--
output]}} {{path/to/compressed_output_file.br}}
```

browser-sync

A local web server that updates browser on file changes.

More information: https://browsersync.io/docs/command-line.

- Start a server from a specific directory:

```
browser-sync start --server {{path/to/directory}} --files {{path/to/directory}}
```

- Start a server from local directory, watching all CSS files in a directory:

```
browser-sync start --server --files '{{path/to/directory/*.css}}'
```

- Create configuration file:

```
browser-sync init
```

- Start Browsersync from configuration file:

```
browser-sync start --config {{config_file}}
```

browsh

View webpages on the terminal using a Firefox backend.

See also: **`carbonyl`**.

More information: https://www.brow.sh/docs/keybindings/.

- Start browsh:

```
browsh
```

- Start browsh in a specific webpage:

```
browsh --startup-url {{URL}}
```

- Focus URL bar:

```
<Ctrl l>
```

- Exit browsh:

```
<Ctrl q>
```

- Display help:

```
browsh {{[-h|--help]}}
```

bru

CLI for Bruno, an Opensource IDE for exploring and testing APIs.

More information: https://docs.usebruno.com/bru-cli/overview.

- Run all request files from the current directory:

```
bru run
```

- Run a single request from the current directory by specifying its filename:

```
bru run {{file.bru}}
```

- Run requests using an environment:

```
bru run --env {{environment_name}}
```

- Run requests using an environment with a variable:

```
bru run --env {{environment_name}} --env-var
{{variable_name}}={{variable_value}}
```

- Run request and collect the results in an output file:

```
bru run --output {{path/to/output.json}}
```

- Display help:

```
bru run --help
```

brushtopbm

Convert a Xerox doodle brush file into a PBM image.

More information: https://netpbm.sourceforge.net/doc/brushtopbm.html.

- Generate a PBM file as output for a Xerox doodle brush file as input:

```
brushtopbm {{path/to/file.brush}}
```

- Display version:

```
brushtopbm {{[-v|-version]}}
```

bshell

A GUI for browsing for SSH/VNC servers on the local network.

See also: **`bssh`**, **`bvnc`**.

More information: https://manned.org/bshell.

- Browse for both SSH and VNC servers:

```
bshell
```

- Browse for SSH servers only:

```
bshell --ssh
```

- Browse for VNC servers only:

```
bshell --vnc
```

- Browse for both SSH and VNC servers in a specified domain:

```
bshell --domain {{domain}}
```

bssh

A GUI tool for browsing for SSH/VNC servers on the local network.

See also: **`bvnc`**, **`bshell`**.

More information: https://manned.org/bssh.

- Browse for SSH servers:

```
bssh
```

- Browse for VNC servers:

```
bssh --vnc
```

- Browse for both SSH and VNC servers:

```
bssh --shell
```

- Browse for SSH servers in a specified domain:

```
bssh --domain {{domain}}
```

btm

Display system information about the CPU, memory, disks, network and processes.

An enhanced alternative to **`top`**.

See also: **`btop`**, **`glances`**, **`atop`**, **`htop`**, **`top`**.

More information: https://clementtsang.github.io/bottom/nightly/#usage-and-configuration.

- Show the default layout (CPU, memory, temperatures, disk, network, and processes):

```
btm
```

- Enable basic mode, removing charts and condensing data (similar to `top`):

```
btm {{[-b|--basic]}}
```

- Use big dots instead of small ones in charts:

```
btm {{[-m|--dot_marker]}}
```

- Show also battery charge and health status:

```
btm --battery
```

- Refresh every 250 milliseconds and show the last 30 seconds in the charts:

```
btm {{[-r|--rate]}} 250 {{[-t|--default_time_value]}} 30000
```

btop

A resource monitor that shows information about the CPU, memory, disks, network and processes.

A C++ version of **`bpytop`**.

See also: **`btm`**, **`glances`**, **`atop`**, **`htop`**, **`top`**.

More information: https://github.com/aristocratos/btop#command-line-options.

- Start `btop`:

```
btop
```

- Start `btop` with the specified settings preset:

```
btop {{[-p|--preset]}} {{0..9}}
```

- Start `btop` in TTY mode using 16 colors and TTY-friendly graph symbols:

```
btop {{[-t|--tty]}}
```

- Start `btop` in 256-color mode instead of 24-bit color mode:

```
btop {{[-l|--low-color]}}
```

- Set the update rate to 500 milliseconds:

```
btop {{[-u|--update]}} 500
```

- Exit `btop`:

```
<q>
```

- Display help:

```
btop {{[-h|--help]}}
```

builtin

Execute shell builtins.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-builtin.

- Run a shell builtin:

```
builtin {{command}}
```

buku

Browser-independent bookmark manager.

More information: https://github.com/jarun/Buku#usage.

- Display all bookmarks matching "keyword" and with "privacy" tag:

```
buku {{keyword}} {{[-t|--stag]}} {{privacy}}
```

- Add bookmark with tags "search engine" and "privacy":

```
buku {{[-a|--add]}} {{https://example.com}} {{search engine}}, {{privacy}}
```

- Delete a bookmark:

```
buku {{[-d|--delete]}} {{bookmark_id}}
```

- Open editor to edit a bookmark:

```
buku {{[-w|--write]}} {{bookmark_id}}
```

- Remove "search engine" tag from a bookmark:

```
buku {{[-u|--update]}} {{bookmark_id}} --tag - {{search engine}}
```

bun build

Bundle JavaScript and TypeScript files with Bun's fast native bundler.

More information: https://bun.com/docs/bundler.

- Bundle an entry point to a single output file:

```
bun build {{path/to/entry.ts}} --outfile {{path/to/output.js}}
```

- Bundle multiple entry points to an output directory:

```
bun build {{path/to/entry1.ts path/to/entry2.ts ...}} --outdir {{path/to/dist}}
```

- Bundle with source maps for debugging:

```
bun build {{path/to/entry.ts}} --outfile {{path/to/output.js}} --sourcemap
```

- Bundle with minification for production:

```
bun build {{path/to/entry.ts}} --outfile {{path/to/output.js}} --minify
```

- Bundle with a specific target environment:

```
bun build {{path/to/entry.ts}} --outfile {{path/to/output.js}} --target {{browser|bun|node}}
```

- Bundle to a standalone executable:

```
bun build {{path/to/entry.ts}} --compile --outfile {{path/to/executable}}
```

- Watch for file changes and rebuild automatically:

```
bun build {{path/to/entry.ts}} --outfile {{path/to/output.js}} --watch
```

- Bundle with external dependencies not included in the output:

```
bun build {{path/to/entry.ts}} --outfile {{path/to/output.js}} --external {{react react-dom}}
```

bun c

This command is an alias of **`bun create`**.

- View documentation for the original command:

```
tldr bun create
```

bun create

Create a new project from a template.

Note: **`c`** can be used as an alias for **`create`**.

More information: https://bun.com/docs/runtime/templating/create.

- Create a new project from an official template interactively:

```
bun create {{template}}
```

- Create a new project from an official template in a new directory:

```
bun create {{template}} {{path/to/destination}}
```

- Create a new project from a GitHub repository template:

```
bun create {{https://github.com/username/repo}} {{path/to/destination}}
```

- Create a new project from a local template:

```
bun create {{./path/to/template}} {{path/to/destination}}
```

- Create a new project, overwriting the destination directory if it exists:

```
bun create {{template}} {{path/to/destination}} --force
```

- Create a new project without initializing a Git repository automatically:

```
bun create {{template}} {{path/to/destination}} --no-git
```

- Create a new project without installing dependencies automatically:

```
bun create {{template}} {{path/to/destination}} --no-install
```

bun i

This command is an alias of **`bun install`**.

- View documentation for the original command:

```
tldr bun install
```

bun install

Install JavaScript dependencies for a project from **`package.json`**.

More information: https://bun.com/docs/cli/install.

- Install all dependencies listed in `package.json`:

  ```
  bun {{[i|install]}}
  ```

- Install a single package (this is an alias for `bun add`):

  ```
  bun {{[i|install]}} {{package_name}}@{{version}}
  ```

- Install a package globally:

  ```
  bun {{[i|install]}} {{[-g|--global]}} {{package_name}}
  ```

- Install only production dependencies (skips `devDependencies`):

  ```
  bun {{[i|install]}} --production
  ```

- Install dependencies exactly from the `bun.lockb` lockfile (frozen lockfile):

  ```
  bun {{[i|install]}} --frozen-lockfile
  ```

- Force re-download all packages from the registry, ignoring the cache:

  ```
  bun {{[i|install]}} {{[-f|--force]}}
  ```

bun link

Bun command to register the current package as a linkable package.

More information: https://bun.com/docs/pm/cli/link.

- Link the package globally:

```
bun link
```

- Link a package locally to a project:

```
bun link {{package_name}}
```

- Link directly from a local path:

```
bun link {{path/to/package}}
```

- Unlink a package:

```
bun unlink {{package_name}}
```

- Unlink a global package:

```
bun unlink --global {{package_name}}
```

- Display help:

```
bun link --help
```

bun remove

Remove a dependency from **`package.json`**.

Note: **`rm`** can be used as an alias for **`remove`**.

More information: https://bun.sh/docs/cli/remove.

- Remove a dependency:

```
bun remove {{package_name}}
```

- Remove multiple dependencies:

```
bun remove {{package_name1 package_name2 ...}}
```

- Remove a globally installed package:

```
bun remove {{[-g|--global]}} {{package_name}}
```

- Remove a dependency without updating the `package.json` file:

```
bun remove --no-save {{package_name}}
```

- Run the command without actually removing packages (simulate the removal):

```
bun remove --dry-run {{package_name}}
```

bun rm

This command is an alias of **`bun remove`**.

- View documentation for the original command:

```
tldr bun remove
```

bun run

Execute a JavaScript/TypeScript file, or a script from **`package.json`**.

More information: https://bun.com/docs/cli/run.

- Run a script defined in `package.json`:

  ```
  bun run {{script_name}}
  ```

- Run a JavaScript or TypeScript file directly:

  ```
  bun run {{path/to/file.ts}}
  ```

- Run a file in "watch" mode (restarts automatically when the file changes):

  ```
  bun run --watch {{path/to/file.ts}}
  ```

- Run a file using a specific configuration file:

  ```
  bun run --config {{path/to/bunfig.toml}} {{path/to/file.ts}}
  ```

bun test

Run tests using Bun's built-in test runner.

It is a fast, Jest-compatible test runner that looks for **`*.test.ts`** (and similar) files.

More information: https://bun.com/docs/cli/test.

- Run all test files found in the project:

```
bun test
```

- Run a specific test file or directory:

```
bun test {{path/to/file.test.ts}}
```

- Run tests in "watch" mode (re-runs automatically on file changes):

```
bun test --watch
```

- Run tests and generate a code coverage report:

```
bun test --coverage
```

bun

JavaScript runtime and toolkit.

Includes a bundler, a test runner, and a package manager.

More information: https://bun.com/docs.

- Create a new Bun project in the current directory:

```
bun init
```

- Run a JavaScript file or a `package.json` script:

```
bun run {{path/to/file|script_name}}
```

- Run unit tests:

```
bun test
```

- Download and install all the packages listed as dependencies in `package.json`:

```
bun {{[i|install]}}
```

- Add a dependency to `package.json`:

```
bun {{[a|add]}} {{module_name}}
```

- Remove a dependency from `package.json`:

```
bun {{[rm|remove]}} {{module_name}}
```

- Start a REPL (interactive shell):

```
bun repl
```

- Upgrade Bun to the latest version:

```
bun upgrade
```

bundle

Dependency manager for the Ruby programming language.

More information: https://bundler.io/man/bundle.1.html.

- Install all gems defined in the `Gemfile` expected in the working directory:

```
bundle install
```

- Execute a command in the context of the current bundle:

```
bundle exec {{command}} {{arguments}}
```

- Update all gems by the rules defined in the `Gemfile` and regenerate `Gemfile.lock`:

```
bundle update
```

- Update one or more specific gem(s) defined in the `Gemfile`:

```
bundle update {{gem_name1}} {{gem_name2}}
```

- Update one or more specific gems(s) defined in the `Gemfile` but only to the next patch version:

```
bundle update --patch {{gem_name1}} {{gem_name2}}
```

- Update all gems within the given group in the `Gemfile`:

```
bundle update --group {{development}}
```

- List installed gems in the `Gemfile` with newer versions available:

```
bundle outdated
```

- Create a new gem skeleton:

```
bundle gem {{gem_name}}
```

bundler

Dependency manager for the Ruby programming language.

`bundler` is a common name for the command **`bundle`**, but not a command itself.

More information: https://bundler.io/man/bundle.1.html.

- View documentation for the original command:

```
tldr bundle
```

bundletool dump

Manipulate Android Application Bundles.

More information: https://developer.android.com/tools/bundletool.

- Display the `AndroidManifest.xml` of the base module:

```
bundletool dump manifest --bundle {{path/to/bundle.aab}}
```

- Display a specific value from the `AndroidManifest.xml` using XPath:

```
bundletool dump manifest --bundle {{path/to/bundle.aab}} --
xpath {{/manifest/@android:versionCode}}
```

- Display the `AndroidManifest.xml` of a specific module:

```
bundletool dump manifest --bundle {{path/to/bundle.aab}} --
module {{name}}
```

- Display all the resources in the application bundle:

```
bundletool dump resources --bundle {{path/to/bundle.aab}}
```

- Display the configuration for a specific resource:

```
bundletool dump resources --bundle {{path/to/bundle.aab}} --
resource {{type/name}}
```

- Display the configuration and values for a specific resource using the ID:

```
bundletool dump resources --bundle {{path/to/bundle.aab}} --
resource {{0x7f0e013a}} --values
```

- Display the contents of the bundle configuration file:

```
bundletool dump config --bundle {{path/to/bundle.aab}}
```

bundletool validate

Manipulate Android Application Bundles.

More information: https://developer.android.com/tools/bundletool.

- Verify a bundle and display detailed information about it:

```
bundletool validate --bundle {{path/to/bundle.aab}}
```

bundletool

Manipulate Android Application Bundles.

Some subcommands such as **`validate`** have their own usage documentation.

More information: https://developer.android.com/tools/bundletool.

- Display help for a subcommand:

```
bundletool help {{subcommand}}
```

- Generate APKs from an application bundle (prompts for keystore password):

```
bundletool build-apks --bundle {{path/to/bundle.aab}} --ks
{{path/to/key.keystore}} --ks-key-alias {{key_alias}} --
output {{path/to/file.apks}}
```

- Generate APKs from an application bundle giving the keystore password:

```
bundletool build-apks --bundle {{path/to/bundle.aab}} --ks
{{path/to/key.keystore}} --ks-key-alias {{key_alias}} --ks-
pass {{pass:the_password}} --output {{path/to/file.apks}}
```

- Generate APKs including only one single APK for universal usage:

```
bundletool build-apks --bundle {{path/to/bundle.aab}} --mode
{{universal}} --ks {{path/to/key.keystore}} --ks-key-alias
{{key_alias}} --output {{path/to/file.apks}}
```

- Install the right combination of APKs to an emulator or device:

```
bundletool install-apks --apks {{path/to/file.apks}}
```

- Estimate the download size of an application:

```
bundletool get-size total --apks {{path/to/file.apks}}
```

- Generate a device specification JSON file for an emulator or device:

```
bundletool get-device-spec --output {{path/to/file.json}}
```

- Verify a bundle and display detailed information about it:

```
bundletool validate --bundle {{path/to/bundle.aab}}
```

bunzip2

This command is an alias of **`bzip2 --decompress`**.

- View documentation for the original command:

```
tldr bzip2
```

bup

Backup system based on the Git packfile format, providing incremental saves and global deduplication.

More information: https://manned.org/bup.

- Initialize a backup repository in a given local directory:

```
bup {{[-d|--bup-dir]}} {{path/to/repository}} init
```

- Prepare a given directory before taking a backup:

```
bup {{[-d|--bup-dir]}} {{path/to/repository}} index {{path/to/directory}}
```

- Backup a directory to the repository specifying its name:

```
bup {{[-d|--bup-dir]}} {{path/to/repository}} save {{[-n|--name]}} {{backup_name}} {{path/to/directory}}
```

- Show the backup snapshots currently stored in the repository:

```
bup {{[-d|--bup-dir]}} {{path/to/repository}} ls
```

- Restore a specific backup snapshot to a target directory:

```
bup {{[-d|--bup-dir]}} {{path/to/repository}} restore {{[-C|--outdir]}} {{path/to/target_directory}} {{backup_name}}
```

busybox

A collection of small system utilities in a single executable.

Executing **`busybox`** via a symlink is equivalent to running **`busybox symlink_name`**.

Linux distributions that use BusyBox will usually provide symlinks for all programs.

More information: https://www.busybox.net/downloads/BusyBox.html.

- Execute a BusyBox function:

```
busybox {{ls|rm|mkdir|cat|...}} {{args}}
```

- Display help and a list of functions:

```
busybox --help
```

buzzphrase

Output a random buzzphrase. Written in Node.js.

More information: https://github.com/atomantic/buzzphrase.

- Generate a string of three random phrases containing an adjective, a past tense verb and a plural noun:

  ```
  buzzphrase
  ```

- Print a phrase formatted as [i]mperative verb + past tense [v]erb + [a]djective + plural [N]oun:

  ```
  buzzphrase '{{{i} {v} {a} {N}}}'
  ```

- Print k phrases formatted as present participle [V]erb + [a]djective + singular [n]oun + [f]inal:

  ```
  buzzphrase {{k}} '{{{V} {a} {n} {f}}}'
  ```

bvnc

A GUI tool for browsing for SSH/VNC servers on the local network.

See also: **`bssh`**, **`bshell`**.

More information: https://manned.org/bvnc.

- Browse for VNC servers:

```
bvnc
```

- Browse for SSH servers:

```
bvnc --ssh
```

- Browse for both VNC and SSH servers:

```
bvnc --shell
```

- Browse for VNC servers in a specified domain:

```
bvnc --domain {{domain}}
```

bw

Access and manage a Bitwarden vault.

More information: https://help.bitwarden.com/article/cli/.

- Log in to a Bitwarden user account:

```
bw login
```

- Log out of a Bitwarden user account:

```
bw logout
```

- Search and display items from Bitwarden vault:

```
bw list items --search {{github}}
```

- Display a particular item from Bitwarden vault:

```
bw get item {{github}}
```

- Create a folder in Bitwarden vault:

```
{{echo -n '{"name":"My Folder1"}' | base64}} | bw create
folder
```

byobu

Window manager and terminal multiplexer.

See also: **`tmux`**, **`screen`**.

More information: https://www.byobu.org/documentation.

- Start a new session:

  `byobu`

- Configuration and help:

  `byobu-config`

- Select tmux (default) or screen backend:

  `byobu-select-backend`

- Enable automatic startup after login into text console:

  `byobu-enable`

- Disable automatic startup after login into text console:

  `byobu-disable`

- Detach from `byobu`:

  `<F6>`

- Kill a window:

  `<Ctrl a><k>`

bzcat

This command is an alias of **`bzip2 --decompress --stdout`**.

- View documentation for the original command:

  ```
  tldr bzip2
  ```

bzegrep

Find extended **regex** patterns in **bzip2** compressed files using **egrep**.

More information: https://manned.org/bzegrep.

- Search for extended regex (supporting ?, +, {}, () and |) in a compressed file (case-sensitive):

```
bzegrep "{{search_pattern}}" {{path/to/file}}
```

- Search for extended regex (supporting ?, +, {}, () and |) in a compressed file (case-insensitive):

```
bzegrep {{[-i|--ignore-case]}} "{{search_pattern}}" {{path/to/file}}
```

- Search for lines that do not match a pattern:

```
bzegrep {{[-v|--invert-match]}} "{{search_pattern}}" {{path/to/file}}
```

- Print file name and line number for each match:

```
bzegrep {{[-H|--with-filename]}} {{[-n|--line-number]}} "{{search_pattern}}" {{path/to/file}}
```

- Search for lines matching a pattern, printing only the matched text:

```
bzegrep {{[-o|--only-matching]}} "{{search_pattern}}" {{path/to/file}}
```

- Recursively search files in a bzip2 compressed tar archive for a pattern:

```
bzegrep {{[-r|--recursive]}} "{{search_pattern}}" {{path/to/file}}
```

bzfgrep

Find any fixed strings separated by new lines in **bzip2** compressed files using **fgrep**.

More information: https://manned.org/bzfgrep.

- Search for lines matching the list of search strings separated by new lines in a compressed file (case-sensitive):

```
bzfgrep "{{search_string}}" {{path/to/file}}
```

- Search for lines matching the list of search strings separated by new lines in a compressed file (case-insensitive):

```
bzfgrep {{[-i|--ignore-case]}} "{{search_string}}" {{path/to/file}}
```

- Search for lines that do not match the list of search strings separated by new lines in a compressed file:

```
bzfgrep {{[-v|--invert-match]}} "{{search_string}}" {{path/to/file}}
```

- Print file name and line number for each match:

```
bzfgrep {{[-H|--with-filename]}} {{[-n|--line-number]}} "{{search_string}}" {{path/to/file}}
```

- Search for lines matching a pattern, printing only the matched text:

```
bzfgrep {{[-o|--only-matching]}} "{{search_string}}" {{path/to/file}}
```

- Recursively search files in a bzip2 compressed tar archive for the given list of strings:

```
bzfgrep {{[-r|--recursive]}} "{{search_string}}" {{path/to/file}}
```

bzgrep

Find patterns in **bzip2** compressed files using **grep**.

More information: https://manned.org/bzgrep.

- Search for a pattern within a compressed file:

```
bzgrep "{{search_pattern}}" {{path/to/file}}
```

- Use extended `regex` (supports `?`, `+`, `{}`, `()` and `|`), in case-insensitive mode:

```
bzgrep {{[-E|--extended-regexp]}} {{[-i|--ignore-case]}}
"{{search_pattern}}" {{path/to/file}}
```

- Print 3 lines of [C]ontext around, [B]efore, or [A]fter each match:

```
bzgrep --{{context|before-context|after-context}} {{3}}
"{{search_pattern}}" {{path/to/file}}
```

- Print file name and line number for each match:

```
bzgrep {{[-H|--with-filename]}} {{[-n|--line-number]}}
"{{search_pattern}}" {{path/to/file}}
```

- Search for lines matching a pattern, printing only the matched text:

```
bzgrep {{[-o|--only-matching]}} "{{search_pattern}}" {{path/
to/file}}
```

- Recursively search files in a bzip2 compressed tar archive for a pattern:

```
bzgrep {{[-r|--recursive]}} "{{search_pattern}}" {{path/to/
tar_file}}
```

- Search `stdin` for lines that do not match a pattern:

```
cat {{path/to/bz_compressed_file}} | bzgrep {{[-v|--invert-
match]}} "{{search_pattern}}"
```

bzip2

A block-sorting file compressor.

See also: **`bzcat`**, **`bunzip2`**, **`bzip2recover`**.

More information: https://manned.org/bzip2.

- Compress a file:

```
bzip2 {{path/to/file_to_compress}}
```

- Decompress a file:

```
bzip2 {{[-d|--decompress]}} {{path/to/compressed_file.bz2}}
```

- Decompress a file to `stdout`:

```
bzip2 {{[-dc|--decompress --stdout]}} {{path/to/
compressed_file.bz2}}
```

- Test the integrity of each file inside the archive file:

```
bzip2 {{[-t|--test]}} {{path/to/compressed_file.bz2}}
```

- Show the compression ratio for each file processed with detailed information:

```
bzip2 {{[-v|--verbose]}} {{path/to/compressed_files.bz2}}
```

- Decompress a file overwriting existing files:

```
bzip2 {{[-f|--force]}} {{path/to/compressed_file.bz2}}
```

- Display help:

```
bzip2 {{[-h|--help]}}
```

bzip2recover

Attempt to recover data from a damaged **`.bz2`** file.

See also: **`bzip2`**.

More information: https://manned.org/bzip2recover.

- Recover all intact blocks from a damaged `.bz2` file:

```
bzip2recover {{damaged_file.bz2}}
```

bzip3

An efficient statistical file compressor.

More information: https://manned.org/bzip3.

- Compress a file:

```
bzip3 {{path/to/file_to_compress}}
```

- Decompress a file:

```
bzip3 {{[-d|--decode]}} {{path/to/compressed_file.bz3}}
```

- Decompress a file to `stdout`:

```
bzip3 {{[-dc|--decode --stdout]}} {{path/to/
compressed_file.bz3}}
```

- Test the integrity of each file inside the archive file:

```
bzip3 {{[-t|--test]}} {{path/to/compressed_file.bz3}}
```

- Show the compression ratio for each file processed with detailed information:

```
bzip3 {{[-v|--verbose]}} {{path/to/compressed_files.bz3}}
```

- Decompress a file overwriting existing files:

```
bzip3 {{[-d|--decode]}} {{[-f|--force]}} {{path/to/
compressed_file.bz3}}
```

- Display help:

```
bzip3 {{[-h|--help]}}
```

c++

This command is an alias of **g++**.

- View documentation for original command:

```
tldr g++
```

c99

Compile C programs according to the ISO C standard.

More information: https://manned.org/c99.

- Compile source file(s) and create an executable:

```
c99 {{file.c}}
```

- Compile source file(s) and specify the executable [o]utput filename:

```
c99 -o {{executable_name}} {{file.c}}
```

- Compile source file(s) and create object file(s):

```
c99 -c {{file.c}}
```

- Compile source file(s), link with object file(s), and create an executable:

```
c99 {{file.c}} {{file.o}}
```

cabal

Interface to the Haskell package infrastructure (Cabal).

Manage Haskell projects and Cabal packages from the Hackage package repository.

More information: https://cabal.readthedocs.io/en/latest/getting-started.html.

- Search and list packages from Hackage:

```
cabal list {{search_string}}
```

- Show information about a package:

```
cabal info {{package}}
```

- Download and install a package:

```
cabal install {{package}}
```

- Create a new Haskell project in the current directory:

```
cabal init
```

- Build the project in the current directory:

```
cabal build
```

- Run tests of the project in the current directory:

```
cabal test
```

cadaver

WebDAV client for Unix.

More information: https://manned.org/cadaver.

- Connect to the server , open the root collection:

```
cadaver {{http://dav.example.com/}}
```

- Connect to a server using a specific port and open a collection:

```
cadaver {{http://dav.example.com:8022/path/to/collection/}}
```

- Connect to a server using SSL:

```
cadaver {{https://davs.example.com/}}
```

caddy

An enterprise-ready open source web server with automatic HTTPS, written in Go.

More information: https://caddyserver.com/docs/command-line.

- Start Caddy in the foreground:

```
caddy run
```

- Start Caddy with the specified Caddyfile:

```
caddy run --config {{path/to/Caddyfile}}
```

- Start Caddy in the background:

```
caddy start
```

- Stop a background Caddy process:

```
caddy stop
```

- Run a simple file server on the specified port with a browsable interface:

```
caddy file-server --listen :{{8000}} --browse
```

- Run a reverse proxy server:

```
caddy reverse-proxy --from :{{80}} --to localhost:{{8000}}
```

cake

The processor for the CakePHP framework.

More information: https://cakephp.org.

- Display basic information about the current app and available commands:

```
cake
```

- List available routes:

```
cake routes
```

- Clear configuration caches:

```
cake cache clear_all
```

- Build the metadata cache:

```
cake schema_cache build --connection {{connection}}
```

- Clear the metadata cache:

```
cake schema_cache clear
```

- Clear a single cache table:

```
cake schema_cache clear {{table_name}}
```

- Start a development web server (defaults to port 8765):

```
cake server
```

- Start a REPL (interactive shell):

```
cake console
```

cal

Display a calendar with the current day highlighted.

See also: **`gcal`**.

More information: https://manned.org/cal.1p.

- Display a calendar for the current month:

```
cal
```

- Display a calendar for a specific year:

```
cal {{year}}
```

- Display a calendar for a specific month and year:

```
cal {{month}} {{year}}
```

calc

An interactive arbitrary-precision calculator in the terminal.

More information: https://manned.org/calc.

- Start `calc` in interactive mode:

```
calc
```

- Perform a calculation in non-interactive mode:

```
calc '{{85 * (36 / 4)}}'
```

- Don't format the output (for use with [p]ipes):

```
calc -p '{{4/3 * pi() * 5^3}}'
```

- Perform a calculation and then switch to [i]nteractive mode:

```
calc -i '{{sqrt(2)}}'
```

- Start `calc` in a specific permission [m]ode (0 to 7, defaults to 7):

```
calc -m {{mode}}
```

- View an introduction to `calc`:

```
calc help intro
```

- View an overview of `calc`:

```
calc help overview
```

- Open the `calc` manual:

```
calc help
```

calendar

Display upcoming events from a calendar file.

More information: https://manned.org/calendar.

- Show events for today and tomorrow (or the weekend on Friday) from the default calendar:

```
calendar
```

- Look [A]head, showing events for the next 30 days:

```
calendar -A {{30}}
```

- Look [B]ack, showing events for the previous 7 days:

```
calendar -B {{7}}
```

- Show events from a custom calendar [f]ile:

```
calendar -f {{path/to/file}}
```

calibre-server

A server application to distribute e-books over a network.

Note: E-books must already be imported into the library using the GUI or the **`calibredb`** CLI.

Part of the Calibre e-book library.

More information: https://manual.calibre-ebook.com/generated/en/calibre-server.html.

- Start a server to distribute e-books. Access at http://localhost:8080:

  ```
  calibre-server
  ```

- Start server on different port. Access at http://localhost:port:

  ```
  calibre-server --port {{port}}
  ```

- Password protect the server:

  ```
  calibre-server --username {{username}} --password
  {{password}}
  ```

calibredb

Manipulate an e-book database.

Part of the Calibre e-book library.

More information: https://manual.calibre-ebook.com/generated/en/calibredb.html.

- List e-books in the library with additional information:

```
calibredb list
```

- Search for e-books displaying additional information:

```
calibredb list --search {{search_term}}
```

- Search for just ids of e-books:

```
calibredb search {{search_term}}
```

- Add one or more e-books to the library:

```
calibredb add {{path/to/file1 path/to/file2 ...}}
```

- Recursively add all e-books under a directory to the library:

```
calibredb add {{[-r|--recurse]}} {{path/to/directory}}
```

- Remove one or more e-books from the library. You need the e-book IDs (see above):

```
calibredb remove {{id1 id2 ...}}
```

caller

Print function context.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-caller.

- Print the line and filename where the current function was called:

```
caller
```

- Print the line, function and filename where the current function was called:

```
caller 0
```

- Print the line, the function name and the filename of a function call n frames back:

```
caller {{n}}
```

calligraflow

Calligra's flowchart and diagram application.

See also: **`calligrastage`**, **`calligrawords`**, **`calligrasheets`**.

More information: https://manned.org/calligraflow.

- Launch the flowchart and diagram application:

  ```
  calligraflow
  ```

- Open a specific file:

  ```
  calligraflow {{path/to/file}}
  ```

- Display help or version:

  ```
  calligraflow --{{help|version}}
  ```

calligrasheets

Calligra's spreadsheet application.

See also: **calligraflow**, **calligrastage**, **calligrawords**.

More information: https://manned.org/calligrasheets.

- Launch the spreadsheet application:

```
calligrasheets
```

- Open a specific spreadsheet:

```
calligrasheets {{path/to/spreadsheet}}
```

- Display help or version:

```
calligrasheets --{{help|version}}
```

calligrastage

Calligra's presentation application.

See also: **calligraflow**, **calligrawords**, **calligrasheets**.

More information: https://manned.org/calligrastage.

- Launch the presentation application:

```
calligrastage
```

- Open a specific presentation:

```
calligrastage {{path/to/presentation}}
```

- Display help or version:

```
calligrastage --{{help|version}}
```

calligrawords

Calligra's word processor application.

See also: **`calligraflow`**, **`calligrastage`**, **`calligrasheets`**.

More information: https://manned.org/calligrawords.

- Launch the word processor application:

```
calligrawords
```

- Open a specific document:

```
calligrawords {{path/to/document}}
```

- Display help or version:

```
calligrawords --{{help|version}}
```

cancel

Cancel print jobs.

See also: **`lp`**, **`lpmove`**, **`lpstat`**.

More information: https://openprinting.github.io/cups/doc/man-cancel.html.

- Cancel the current job of the default printer (set with `lpoptions -d printer_name`):

```
cancel
```

- Cancel the jobs of the default printer owned by a specific [u]ser:

```
cancel -u {{username}}
```

- Cancel the current job of a specific printer:

```
cancel {{printer}}
```

- Cancel a specific job from a specific printer:

```
cancel {{printer}}-{{job_id}}
```

- Cancel [a]ll jobs of all printers:

```
cancel -a
```

- Cancel [a]ll jobs of a specific printer:

```
cancel -a {{printer}}
```

- Cancel the current job of a specific [h]ost server and then delete ([x]) job data files:

```
cancel -h {{server}} -x
```

carbon-now

Create beautiful images of code.

More information: https://github.com/mixn/carbon-now-cli.

- Create an image from a file using default settings:

```
carbon-now {{path/to/file}}
```

- Create an image from a text in clipboard using default settings:

```
carbon-now --from-clipboard
```

- Create an image from `stdin` using default settings and copy to the clipboard:

```
{{input}} | carbon-now --to-clipboard
```

- Create images interactively for custom settings and optionally save a preset:

```
carbon-now {{[-i|--interactive]}} {{path/to/file}}
```

- Create images from a previously saved preset:

```
carbon-now {{[-p|--preset]}} {{preset}} {{path/to/file}}
```

- Start at a specified line of text:

```
carbon-now {{[-s|--start]}} {{line}} {{path/to/file}}
```

- End at a specific line of text:

```
carbon-now {{[-e|--end]}} {{line}} {{path/to/file}}
```

- Open image in a browser instead of saving:

```
carbon-now --open {{path/to/file}}
```

carbonyl

View webpages on the terminal using a Chromium backend.

See also: **`browsh`**.

More information: https://github.com/fathyb/carbonyl.

- Open an `about:blank` page:

  `carbonyl`

- Open a webpage:

  `carbonyl {{https://example.com}}`

- Exit carbonyl:

  `<Ctrl c>`

- Display help:

  `carbonyl {{[-h|--help]}}`

cargo add

Add dependencies to a Rust project's **`Cargo.toml`** manifest.

More information: https://doc.rust-lang.org/cargo/commands/cargo-add.html.

- Add the latest version of a dependency to the current project:

```
cargo add {{dependency}}
```

- Add a specific version of a dependency:

```
cargo add {{dependency}}@{{version}}
```

- Add a dependency and enable one or more specific features:

```
cargo add {{dependency}} {{[-F|--features]}} {{feature_1}},
{{feature_2}}
```

- Add an optional dependency, which then gets exposed as a feature of the crate:

```
cargo add {{dependency}} --optional
```

- Add a local crate as a dependency:

```
cargo add --path {{path/to/crate_directory}}
```

- Add a development or build dependency:

```
cargo add {{dependency}} --{{dev|build}}
```

- Add a dependency with all default features disabled:

```
cargo add {{dependency}} --no-default-features
```

cargo bench

Compile and execute benchmarks.

More information: https://doc.rust-lang.org/cargo/commands/cargo-bench.html.

- Execute all benchmarks of a package:

```
cargo bench
```

- Don't stop when a benchmark fails:

```
cargo bench --no-fail-fast
```

- Compile, but don't run benchmarks:

```
cargo bench --no-run
```

- Benchmark the specified benchmark:

```
cargo bench --bench {{benchmark}}
```

- Benchmark with the given profile (default: `bench`):

```
cargo bench --profile {{profile}}
```

- Benchmark all example targets:

```
cargo bench --examples
```

- Benchmark all binary targets:

```
cargo bench --bins
```

- Benchmark the package's library:

```
cargo bench --lib
```

cargo binstall

Install Rust binaries from CI artifacts.

Falls back to **`cargo install`** (from source code) if there are no binaries available.

More information: https://github.com/cargo-bins/cargo-binstall.

- Install a package from https://crates.io:

  ```
  cargo binstall {{package}}
  ```

- Install a specific version of a package (latest by default):

  ```
  cargo binstall {{package}}@{{version}}
  ```

- Install a package and disable confirmation prompts:

  ```
  cargo binstall {{[-y|--no-confirm]}} {{package}}
  ```

cargo build

Compile a local package and all of its dependencies.

More information: https://doc.rust-lang.org/cargo/commands/cargo-build.html.

- Build the package or packages defined by the `Cargo.toml` manifest file in the local path:

```
cargo {{[b|build]}}
```

- Build artifacts in release mode, with optimizations:

```
cargo {{[b|build]}} {{[-r|--release]}}
```

- Require that `Cargo.lock` is up to date:

```
cargo {{[b|build]}} --locked
```

- Build all packages in the workspace:

```
cargo {{[b|build]}} --workspace
```

- Build a specific package:

```
cargo {{[b|build]}} {{[-p|--package]}} {{package}}
```

- Build only the specified binary:

```
cargo {{[b|build]}} --bin {{name}}
```

- Build only the specified test target:

```
cargo {{[b|build]}} --test {{test_name}}
```

cargo check

Check a local package and all of its dependencies for errors.

More information: https://doc.rust-lang.org/cargo/commands/cargo-check.html.

- Check the current package:

```
cargo {{[c|check]}}
```

- Check all tests:

```
cargo {{[c|check]}} --tests
```

- Check the integration tests in `tests/integration_test1.rs`:

```
cargo {{[c|check]}} --test {{integration_test1}}
```

- Check the current package with the features `feature1` and `feature2`:

```
cargo {{[c|check]}} {{[-F|--features]}} {{feature1,feature2}}
```

- Check the current package with default features disabled:

```
cargo {{[c|check]}} --no-default-features
```

cargo clean

Remove generated artifacts in the **`target`** directory.

More information: https://doc.rust-lang.org/cargo/commands/cargo-clean.html.

- Remove the entire `target` directory:

  ```
  cargo clean
  ```

- Remove documentation artifacts (the `target/doc` directory):

  ```
  cargo clean --doc
  ```

- Remove release artifacts (the `target/release` directory):

  ```
  cargo clean {{[-r|--release]}}
  ```

- Remove artifacts in the directory of the given profile (in this case, `target/debug`):

  ```
  cargo clean --profile {{dev}}
  ```

cargo clippy

A collection of lints to catch common mistakes and improve your Rust code.

More information: https://github.com/rust-lang/rust-clippy.

- Run checks over the code in the current directory:

  ```
  cargo clippy
  ```

- Require that `Cargo.lock` is up to date:

  ```
  cargo clippy --locked
  ```

- Run checks on all packages in the workspace:

  ```
  cargo clippy --workspace
  ```

- Run checks for a package:

  ```
  cargo clippy --package {{package}}
  ```

- Run checks for a lint group (see https://rust-lang.github.io/rust-clippy/stable/index.html#?groups=cargo,complexity,correctness,deprecated,nursery,pedantic,perf,restriction,style,s

  ```
  cargo clippy -- {{[-W|--warn]}} clippy::{{lint_group}}
  ```

- Treat warnings as errors:

  ```
  cargo clippy -- {{[-D|--deny]}} warnings
  ```

- Run checks and ignore warnings:

  ```
  cargo clippy -- {{[-A|--allow]}} warnings
  ```

- Apply Clippy suggestions automatically:

  ```
  cargo clippy --fix
  ```

cargo deb

Create Debian packages from Cargo projects.

More information: https://github.com/kornelski/cargo-deb.

- Create a Debian package from a project:

```
cargo deb
```

- Write the `.deb` file to the specified file or directory:

```
cargo deb {{[-o|--output]}} {{path/to/file_or_directory}}
```

- Compile for the specified Rust target triple:

```
cargo deb --target {{x86_64-unknown-linux-gnu}}
```

- Select which package to use in a Cargo workspace:

```
cargo deb {{[-p|--package]}} {{package_name}}
```

- Immediately install the created package:

```
cargo deb --install
```

cargo doc

Build the documentation of Rust packages.

More information: https://doc.rust-lang.org/cargo/commands/cargo-doc.html.

- Build the documentation for the current project and all dependencies:

```
cargo {{[d|doc]}}
```

- Do not build documentation for dependencies:

```
cargo {{[d|doc]}} --no-deps
```

- Build and open the documentation in a browser:

```
cargo {{[d|doc]}} --open
```

- Build and view the documentation of a particular package:

```
cargo {{[d|doc]}} --open {{[-p|--package]}} {{package}}
```

cargo fetch

Fetch dependencies of a package from the network.

More information: https://doc.rust-lang.org/cargo/commands/cargo-fetch.html.

- Fetch dependencies specified in `Cargo.lock` (for all targets):

```
cargo fetch
```

- Fetch dependencies for the specified target:

```
cargo fetch --target {{target_triple}}
```

cargo fix

Automatically fix lint warnings reported by **`rustc`**.

More information: https://doc.rust-lang.org/cargo/commands/cargo-fix.html.

- Fix code even if it already has compiler errors:

  ```
  cargo fix --broken-code
  ```

- Fix code even if the working directory has changes:

  ```
  cargo fix --allow-dirty
  ```

- Migrate a package to the next Rust edition:

  ```
  cargo fix --edition
  ```

- Fix the package's library:

  ```
  cargo fix --lib
  ```

- Fix the specified integration test:

  ```
  cargo fix --test {{name}}
  ```

- Fix all members in the workspace:

  ```
  cargo fix --workspace
  ```

cargo fmt

Run **`rustfmt`** on all source files in a Rust project.

More information: https://github.com/rust-lang/rustfmt.

- Format all source files:

```
cargo fmt
```

- Check for formatting errors without writing to the files:

```
cargo fmt --check
```

- Pass arguments to each `rustfmt` call:

```
cargo fmt -- {{rustfmt_args}}
```

cargo generate-lockfile

Generate the **`Cargo.lock`** file for the current package. Similar to **`cargo update`**, but has less options.

If the lockfile already exists it will be rebuilt with latest version of every package.

More information: https://doc.rust-lang.org/stable/cargo/commands/cargo-generate-lockfile.html.

- Generate a `Cargo.lock` file with the latest version of every package:

  ```
  cargo generate-lockfile
  ```

cargo help

Display help on **`cargo`** and its subcommands.

More information: https://doc.rust-lang.org/cargo/commands/cargo-help.html.

- Display general help:

```
cargo help
```

- Display help for a subcommand:

```
cargo help {{subcommand}}
```

cargo info

Display information about a Rust package.

More information: https://doc.rust-lang.org/cargo/commands/cargo-info.html.

- Display information about a package on https://crates.io:

```
cargo info {{package}}
```

- Display information about a specific version of a package:

```
cargo info {{package}}@{{version}}
```

- Display additional information about a package:

```
cargo info {{[-v|--verbose]}} {{package}}
```

cargo init

Create a new Cargo package.

Equivalent of **`cargo new`**, but specifying a directory is optional.

More information: https://doc.rust-lang.org/cargo/commands/cargo-init.html.

- Initialize a Rust project with a binary target in the current directory:

```
cargo init
```

- Initialize a Rust project with a binary target in the specified directory:

```
cargo init {{path/to/directory}}
```

- Initialize a Rust project with a library target in the current directory:

```
cargo init --lib
```

- Initialize a version control system repository in the project directory (default: `git`):

```
cargo init --vcs {{git|hg|pijul|fossil|none}}
```

- Set the package name (default: directory name):

```
cargo init --name {{name}}
```

cargo install

Build and install a Rust binary.

More information: https://doc.rust-lang.org/cargo/commands/cargo-install.html.

- Install a package from https://crates.io (the version is optional - latest by default):

```
cargo install {{package}}@{{version}}
```

- Install a package from the specified Git repository:

```
cargo install --git {{repo_url}}
```

- Build from the specified branch/tag/commit when installing from a Git repository:

```
cargo install --git {{repo_url}} --{{branch|tag|rev}} {{branch_name|tag|commit_hash}}
```

- Install a package from a local directory:

```
cargo install --path {{path/to/package}}
```

- List all installed packages and their versions:

```
cargo install --list
```

cargo locate-project

Print the full path to the **`Cargo.toml`** manifest of a project.

If the project is part of a workspace, the manifest of the project is shown, rather than that of the workspace.

More information: https://doc.rust-lang.org/cargo/commands/cargo-locate-project.html.

- Display the JSON object with full path to the `Cargo.toml` manifest:

  ```
  cargo locate-project
  ```

- Display the project path in the specified format:

  ```
  cargo locate-project --message-format {{plain|json}}
  ```

- Display the `Cargo.toml` manifest located at the root of the workspace as opposed to the current workspace member:

  ```
  cargo locate-project --workspace
  ```

- Display the `Cargo.toml` manifest of a specific directory:

  ```
  cargo locate-project --manifest-path {{path/to/Cargo.toml}}
  ```

cargo login

Save an API token from the registry locally.

The token is used to authenticate to a package registry. You can remove it using **`cargo logout`**.

More information: https://doc.rust-lang.org/cargo/commands/cargo-login.html.

- Add an API token to the local credential storage (located in `$CARGO_HOME/credentials.toml`):

  ```
  cargo login
  ```

- Use the specified registry (registry names can be defined in the configuration - the default is https://crates.io):

  ```
  cargo login --registry {{name}}
  ```

cargo logout

Remove an API token from the registry locally.

The token is used to authenticate to a package registry. You can add it back using **`cargo login`**.

More information: https://doc.rust-lang.org/cargo/commands/cargo-logout.html.

- Remove an API token from the local credential storage (located in `$CARGO_HOME/credentials.toml`):

  ```
  cargo logout
  ```

- Use the specified registry (registry names can be defined in the configuration - the default is https://crates.io):

  ```
  cargo logout --registry {{name}}
  ```

cargo metadata

Output the workspace members and resolved dependencies of current package as JSON.

Note: The output format is subject to change in future versions of Cargo.

More information: https://doc.rust-lang.org/cargo/commands/cargo-metadata.html.

- Print the workspace members and resolved dependencies of the current package:

  ```
  cargo metadata
  ```

- Print only the workspace members and do not fetch dependencies:

  ```
  cargo metadata --no-deps
  ```

- Print metadata in a specific format based on the specified version:

  ```
  cargo metadata --format-version {{version}}
  ```

- Print metadata with the `resolve` field including dependencies only for the given target triple (Note: The `packages` array will still include the dependencies for all targets):

  ```
  cargo metadata --filter-platform {{target_triple}}
  ```

cargo msrv

Manage the Minimum Supported Rust Version (MSRV) of a project.

More information: https://gribnau.dev/cargo-msrv.

- Display the MSRVs of dependencies (as specified in their `Cargo.toml`):

  ```
  cargo msrv list
  ```

- Find the MSRV of the project by trying to compile it with various toolchains:

  ```
  cargo msrv find
  ```

- Show the MSRV of the project as specified in `Cargo.toml`:

  ```
  cargo msrv show
  ```

- Set the MSRV in `Cargo.toml` to a given Rust version:

  ```
  cargo msrv set {{version}}
  ```

- Verify whether the MSRV is satisfiable by compiling the project using the specified version of Rust:

  ```
  cargo msrv verify
  ```

cargo new

Create a new Cargo package.

Equivalent of **`cargo init`**, but specifying a directory is required.

More information: https://doc.rust-lang.org/cargo/commands/cargo-new.html.

- Create a new Rust project with a binary target:

```
cargo new {{path/to/directory}}
```

cargo owner

Manage the owners of a crate on the registry.

More information: https://doc.rust-lang.org/cargo/commands/cargo-owner.html.

- Invite the given user or team as an owner:

```
cargo owner {{[-a|--add]}} {{username|
github:org_name:team_name}} {{crate}}
```

- Remove the given user or team as an owner:

```
cargo owner {{[-r|--remove]}} {{username|
github:org_name:team_name}} {{crate}}
```

- List owners of a crate:

```
cargo owner {{[-l|--list]}} {{crate}}
```

- Use the specified registry (registry names can be defined in the configuration - the default is https://crates.io):

```
cargo owner --registry {{name}}
```

cargo package

Assemble a local package into a distributable tarball (a **`.crate`** file).

Similar to **`cargo publish --dry-run`**, but has more options.

More information: https://doc.rust-lang.org/cargo/commands/cargo-package.html.

- Perform checks and create a `.crate` file (equivalent of `cargo publish --dry-run`):

  ```
  cargo package
  ```

- Display what files would be included in the tarball without actually creating it:

  ```
  cargo package {{[-l|--list]}}
  ```

cargo pkgid

Print the fully qualified package ID specifier for a package or dependency in the current workspace.

More information: https://doc.rust-lang.org/cargo/commands/cargo-pkgid.html.

- Print the fully qualified package specification for the current project:

```
cargo pkgid
```

- Print the fully qualified package specification for the specified package:

```
cargo pkgid {{partial_pkgspec}}
```

cargo publish

Upload a package to a registry.

Note: You have to add an authentication token using **`cargo login`** before publishing a package.

More information: https://doc.rust-lang.org/cargo/commands/cargo-publish.html.

- Perform checks, create a `.crate` file and upload it to the registry:

  ```
  cargo publish
  ```

- Perform checks, create a `.crate` file but don't upload it (equivalent of `cargo package`):

  ```
  cargo publish {{[-n|--dry-run]}}
  ```

- Use the specified registry (registry names can be defined in the configuration - the default is https://crates.io):

  ```
  cargo publish --registry {{name}}
  ```

cargo remove

Remove dependencies from a Rust project's **`Cargo.toml`** manifest.

More information: https://doc.rust-lang.org/cargo/commands/cargo-remove.html.

- Remove a dependency from the current project:

```
cargo remove {{dependency}}
```

- Remove a development or build dependency:

```
cargo remove --{{dev|build}} {{dependency}}
```

- Remove a dependency of the given target platform:

```
cargo remove --target {{target}} {{dependency}}
```

cargo report

Display various kinds of reports.

More information: https://doc.rust-lang.org/cargo/commands/cargo-report.html.

- Display a report of crates which will eventually stop compiling:

```
cargo report future-incompatibilities
```

- Display a report with the specified Cargo-generated ID:

```
cargo report future-incompatibilities --id {{id}}
```

- Display a report for the specified package:

```
cargo report future-incompatibilities {{[-p|--package]}} {{package}}
```

cargo run

Run the current Cargo package.

Note: The working directory of the executed binary will be set to the current working directory.

More information: https://doc.rust-lang.org/cargo/commands/cargo-run.html.

- Run the default binary target:

```
cargo {{[r|run]}}
```

- Run the specified binary:

```
cargo {{[r|run]}} --bin {{name}}
```

- Run the specified example:

```
cargo {{[r|run]}} --example {{name}}
```

- Activate a space or comma separated list of features:

```
cargo {{[r|run]}} {{[-F|--features]}} "{{feature1
feature2 ...}}"
```

- Disable the default features:

```
cargo {{[r|run]}} --no-default-features
```

- Activate all available features:

```
cargo {{[r|run]}} --all-features
```

- Run with the given profile:

```
cargo {{[r|run]}} --profile {{name}}
```

cargo rustc

Compile a Rust package. Similar to **`cargo build`**, but you can pass extra options to the compiler.

See **`rustc --help`** for all available options.

More information: https://doc.rust-lang.org/cargo/commands/cargo-rustc.html.

- Build the package and pass options to `rustc`:

```
cargo rustc -- {{rustc_options}}
```

- Build artifacts in release mode, with optimizations:

```
cargo rustc {{[-r|--release]}}
```

- Compile with architecture-specific optimizations for the current CPU:

```
cargo rustc {{[-r|--release]}} -- -C target-cpu=native
```

- Compile with speed optimizations:

```
cargo rustc -- -C opt-level {{1|2|3}}
```

- Compile with [s]ize optimizations (`z` also turns off loop vectorization):

```
cargo rustc -- -C opt-level {{s|z}}
```

- Check if your package uses unsafe code:

```
cargo rustc --lib -- -D unsafe-code
```

- Build a specific package:

```
cargo rustc {{[-p|--package]}} {{package}}
```

- Build only the specified binary:

```
cargo rustc --bin {{name}}
```

cargo rustdoc

Build the documentation of Rust packages.

Similar to **`cargo doc`**, but you can pass options to **`rustdoc`**. See **`rustdoc --help`** for all available options.

More information: https://doc.rust-lang.org/cargo/commands/cargo-rustdoc.html.

- Pass options to `rustdoc`:

```
cargo rustdoc -- {{rustdoc_options}}
```

- Warn about a documentation lint:

```
cargo rustdoc -- --warn rustdoc::{{lint_name}}
```

- Ignore a documentation lint:

```
cargo rustdoc -- --allow rustdoc::{{lint_name}}
```

- Document the package's library:

```
cargo rustdoc --lib
```

- Document the specified binary:

```
cargo rustdoc --bin {{name}}
```

- Document the specified example:

```
cargo rustdoc --example {{name}}
```

- Document the specified integration test:

```
cargo rustdoc --test {{name}}
```

cargo search

Search for packages on https://crates.io.

The crates are displayed along with descriptions in TOML format suitable for copying into **`Cargo.toml`**.

More information: https://doc.rust-lang.org/cargo/commands/cargo-search.html.

- Search for packages:

  ```
  cargo search {{query}}
  ```

- Show n results (default: 10, max: 100):

  ```
  cargo search --limit {{n}} {{query}}
  ```

cargo test

Execute the unit and integration tests of a Rust package.

More information: https://doc.rust-lang.org/cargo/commands/cargo-test.html.

- Only run tests containing a specific string in their names:

```
cargo {{[t|test]}} {{test_name}}
```

- Set the number of simultaneous running test cases:

```
cargo {{[t|test]}} -- --test-threads {{count}}
```

- Test artifacts in release mode, with optimizations:

```
cargo {{[t|test]}} {{[-r|--release]}}
```

- Test all packages in the workspace:

```
cargo {{[t|test]}} --workspace
```

- Run tests for a specific package:

```
cargo {{[t|test]}} {{[-p|--package]}} {{package}}
```

- Run tests without hiding output from test executions:

```
cargo {{[t|test]}} -- --nocapture
```

cargo tree

Display a tree visualization of a dependency graph.

Note: In the tree, dependencies of packages marked with `(*)` have already been shown elsewhere in the graph, and so are not repeated.

More information: https://doc.rust-lang.org/cargo/commands/cargo-tree.html.

- Show a dependency tree of the current project:

```
cargo tree
```

- Only show dependencies up to the specified depth (e.g. when n is 1, display only direct dependencies):

```
cargo tree --depth {{n}}
```

- Do not display the given package (and its dependencies) in the tree:

```
cargo tree --prune {{package_spec}}
```

- Show all occurrences of repeated dependencies:

```
cargo tree --no-dedupe
```

- Only show normal/build/development dependencies:

```
cargo tree {{[-e|--edges]}} {{normal|build|dev}}
```

cargo uninstall

Remove a Rust binary installed globally using **`cargo install`**.

More information: https://doc.rust-lang.org/cargo/commands/cargo-uninstall.html.

- Remove an installed binary:

```
cargo uninstall {{package_spec}}
```

cargo update

Update dependencies as recorded in **`Cargo.lock`**.

More information: https://doc.rust-lang.org/cargo/commands/cargo-update.html.

- Update dependencies in `Cargo.lock` to the latest possible version:

```
cargo update
```

- Display what would be updated, but don't actually write the lockfile:

```
cargo update {{[-n|--dry-run]}}
```

- Update only the specified dependencies:

```
cargo update --package {{dependency1}} --package
{{dependency2}} --package {{dependency3}}
```

- Set a specific dependency to a specific version:

```
cargo update --package {{dependency}} --precise {{1.2.3}}
```

cargo vendor

Vendor all dependencies of a project into the specified directory (default: **`vendor`**).

More information: https://doc.rust-lang.org/cargo/commands/cargo-vendor.html.

- Vendor dependencies and configure `cargo` to use the vendored sources in the current project:

```
cargo vendor {{path/to/directory}} > .cargo/config.toml
```

cargo verify-project

Check the correctness of the **`Cargo.toml`** manifest and print the result as a JSON object.

Note: this command is deprecated and may be removed in the future.

More information: https://doc.rust-lang.org/cargo/commands/deprecated-and-removed.html.

- Check the correctness of the current project's manifest:

```
cargo verify-project
```

- Check the correctness of the specified manifest file:

```
cargo verify-project --manifest-path {{path/to/Cargo.toml}}
```

cargo version

Display **`cargo`** version information.

More information: https://doc.rust-lang.org/cargo/commands/cargo-version.html.

- Display version:

```
cargo version
```

- Display additional build information:

```
cargo version {{[-v|--verbose]}}
```

cargo yank

Remove a pushed crate from the index. This should only be used when you accidentally release a significantly broken crate.

Note: This does not remove any data. The crate is still present after a yank - this just prevents new projects from using it.

More information: https://doc.rust-lang.org/cargo/commands/cargo-yank.html.

- Yank the specified version of a crate:

  ```
  cargo yank {{crate}}@{{version}}
  ```

- Undo a yank (i.e. allow downloading it again):

  ```
  cargo yank --undo {{crate}}@{{version}}
  ```

- Use the specified registry (registry names can be defined in the configuration - the default is https://crates.io):

  ```
  cargo yank --registry {{name}} {{crate}}@{{version}}
  ```

cargo

Manage Rust projects and their module dependencies (crates).

Some subcommands such as **`build`** have their own usage documentation.

More information: https://doc.rust-lang.org/cargo.

- Search for crates:

```
cargo search {{search_string}}
```

- Install a binary crate:

```
cargo install {{crate_name}}
```

- List installed binary crates:

```
cargo install --list
```

- Create a new binary or library Rust project in the specified directory (or the current working directory by default):

```
cargo init --{{bin|lib}} {{path/to/directory}}
```

- Add a dependency to `Cargo.toml` in the current directory:

```
cargo add {{dependency}}
```

- Build the Rust project in the current directory using the release profile:

```
cargo {{[b|build]}} {{[-r|--release]}}
```

- Build the Rust project in the current directory using the nightly compiler (requires `rustup`):

```
cargo +nightly {{[b|build]}}
```

- Build using a specific number of threads (default is the number of logical CPUs):

```
cargo {{[b|build]}} --jobs {{number_of_threads}}
```

cariddi

Crawl URLs and scan for endpoints, secrets, api keys, file extensions, tokens, and more from a list of domains.

More information: https://github.com/edoardottt/cariddi/wiki.

- Hunt for secrets using custom `regex`es and output results in JSON:

```
cat {{path/to/urls.txt}} | cariddi -s -sf {{path/to/custom_secrets.txt}} -json
```

- Hunt for juicy endpoints with high concurrency and timeout with plain output results:

```
cat {{path/to/urls.txt}} | cariddi -e -c {{250}} -t {{15}} -plain
```

- Crawl with debug mode and store HTTP responses and output results in `txt` file:

```
cat {{path/to/urls.txt}} | cariddi -debug -sr -ot {{path/to/debug_output.txt}}
```

- Perform an intensive crawl with a proxy and random user agent and output results in `html` file:

```
cat {{path/to/urls.txt}} | cariddi -intensive -proxy {{http://127.0.0.1:8080}} -rua -oh {{path/to/intensive_crawl.html}}
```

- Hunt for errors and useful information with a custom delay and use `.cariddi_cache` folder as cache:

```
cat {{path/to/urls.txt}} | cariddi -err -info -d {{3}} -cache
```

- Show example uses:

```
cariddi -examples
```

carp

REPL and build tool for Carp.

More information: https://carp-lang.github.io/carp-docs/Manual.html.

- Start a REPL (interactive shell):

```
carp
```

- Start a REPL with a custom prompt:

```
carp --prompt "{{> }}"
```

- Build a `carp` file:

```
carp -b {{path/to/file.carp}}
```

- Build and run a file:

```
carp -x {{path/to/file.carp}}
```

- Build a file with optimizations enabled:

```
carp -b --optimize {{path/to/file.carp}}
```

- Transpile a file to C code:

```
carp --generate-only {{path/to/file.carp}}
```

case

Bash builtin construct for creating multi-choice conditional statements.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-case.

- Match a variable against string literals to decide which command to run:

```
case {{$COUNTRULE}} in {{words}}) {{wc --words README}} ;;
{{lines}}) {{wc --lines README}} ;; esac
```

- Combine patterns with |, use * as a fallback pattern:

```
case {{$COUNTRULE}} in {{[wW]|words}}) {{wc --words
README}} ;; {{[lL]|lines}}) {{wc --lines README}} ;; *)
{{echo "what?"}} ;; esac
```

- Allow matching multiple patterns:

```
case {{$ANIMAL}} in {{cat}}) {{echo "It's a cat"}} ;;& {{cat|
dog}}) {{echo "It's a cat or a dog"}} ;;& *) {{echo
"Fallback"}} ;; esac
```

- Continue to the next pattern's commands without checking the pattern:

```
case {{$ANIMAL}} in {{cat}}) {{echo "It's a cat"}} ;&
{{dog}}) {{echo "It's either a dog or cat fell through"}} ;&
*) {{echo "Fallback"}} ;; esac
```

- Display help:

```
help case
```

cat

Print and concatenate files.

More information: https://manned.org/cat.1posix.

- Print the contents of a file to `stdout`:

```
cat {{path/to/file}}
```

- Concatenate several files into an output file:

```
cat {{path/to/file1 path/to/file2 ...}} > {{path/to/output_file}}
```

- Append several files to an output file:

```
cat {{path/to/file1 path/to/file2 ...}} >> {{path/to/output_file}}
```

- Copy the contents of a file into an output file without buffering:

```
cat -u {{/dev/tty12}} > {{/dev/tty13}}
```

- Write `stdin` to a file:

```
cat - > {{path/to/file}}
```

catimg

Image printing in the terminal.

See also: **`pixterm`**, **`chafa`**.

More information: https://manned.org/catimg.

- Print a JPEG, PNG, or GIF to the terminal:

```
catimg {{path/to/file}}
```

- Double the [r]esolution of an image:

```
catimg -r 2 {{path/to/file}}
```

- Disable 24-bit color for better [t]erminal support:

```
catimg -t {{path/to/file}}
```

- Specify a custom [w]idth or [H]eight:

```
catimg {{-w|-H}} {{40}} {{path/to/file}}
```

cava

Cross-platform audio visualizer.

More information: https://github.com/karlstav/cava#usage.

- Start the visualizer:

  `cava`

- Use the specified configuration file:

  `cava -p {{path/to/file}}`

- Increase/Decrease sensitivity:

  `{{<ArrowUp>|<ArrowDown>}}`

- Increase/Decrease the number of bars:

  `{{<ArrowRight>|<ArrowLeft>}}`

- Reload the configuration file (this resets the number of bars and sensitivity):

  `<r>`

- Reload colors from the configuration file:

  `<c>`

- Cycle the foreground/background color:

  `{{<f>|<b>}}`

- Quit:

  `<q>`

cavif

Convert PNG/JPEG images to AVIF. Written in Rust.

See also: **`convert`**.

More information: https://github.com/kornelski/cavif-rs.

- Convert a JPEG file to AVIF, saving it to `file.avif`:

```
cavif {{path/to/image.jpg}}
```

- Adjust the image quality and convert a PNG file to AVIF:

```
cavif --quality {{1..100}} {{path/to/image.png}}
```

- Specify the output location:

```
cavif {{path/to/image.jpg}} --output {{path/to/output.avif}}
```

- Overwrite the destination file if it already exists:

```
cavif --overwrite {{path/to/image.jpg}}
```

cb

Cut, copy, and paste anything in the terminal.

More information: https://github.com/Slackadays/Clipboard.

- Show all clipboards:

```
cb
```

- Copy a file to the clipboard:

```
cb copy {{path/to/file}}
```

- Copy some text to the clipboard:

```
cb copy "{{Some example text}}"
```

- Copy piped data to the clipboard:

```
echo "{{Some example text}}" | cb
```

- Paste clipboard content:

```
cb paste
```

- Pipe out clipboard content:

```
cb | cat
```

- Show clipboard history:

```
cb history
```

- Show clipboard information:

```
cb info
```

cbonsai

A beautifully random bonsai tree generator.

See also: **`cmatrix`**.

More information: https://gitlab.com/jallbrit/cbonsai.

- Generate a bonsai in live mode:

```
cbonsai {{[-l|--live]}}
```

- Generate a bonsai in infinite mode:

```
cbonsai {{[-i|--infinite]}}
```

- Set the growth factor of the tree (default: 32):

```
cbonsai {{[-L|--life]}} {{0..200}}
```

- Set the branching factor of the tree (default: 5):

```
cbonsai {{[-M|--multiplier]}} {{0..20}}
```

- Run in screensaver mode (equivalent to `--live --infinite` but any keypress exits):

```
cbonsai {{[-S|--screensaver]}}
```

- Append a message to the bonsai:

```
cbonsai {{[-m|--message]}} "{{message}}"
```

- Display extra information about the bonsai:

```
cbonsai {{[-v|--verbose]}}
```

- Display help:

```
cbonsai {{[-h|--help]}}
```

cbt

Utility for reading data from Google Cloud's Bigtable.

More information: https://cloud.google.com/bigtable/docs/cbt-reference.

- List tables in the current project:

```
cbt ls
```

- Print count of rows in a specific table in the current project:

```
cbt count "{{table_name}}"
```

- Display a single row from a specific table with only 1 (most recent) cell revision per column in the current project:

```
cbt lookup "{{table_name}}" "{{row_key}}" cells-per-column={{1}}
```

- Display a single row with only specific column(s) (omit qualifier to return entire family) in the current project:

```
cbt lookup "{{table_name}}" "{{row_key}}" columns="{{family1:qualifier1,family2:qualifier2,...}}"
```

- Search up to 5 rows in the current project by a specific `regex` pattern and print them:

```
cbt read "{{table_name}}" regex="{{row_key_pattern}}" count={{5}}
```

- Read a specific range of rows and print only returned row keys in the current project:

```
cbt read {{table_name}} start={{start_row_key}} end={{end_row_key}} keys-only=true
```

ccache

C/C++ compiler cache.

Note: Packages usually provide symlinks for compilers in **`/usr/lib/ccache/bin`**. Prepend this directory to **`$PATH`** to automatically use **`ccache`** for them.

More information: https://ccache.dev/manual/latest.html.

- Show current cache statistics:

```
ccache {{[-s|--show-stats]}}
```

- Clear all cache:

```
ccache {{[-C|--clear]}}
```

- Reset statistics (but not cache itself):

```
ccache {{[-z|--zero-stats]}}
```

- Compile C code and cache compiled output (to use `ccache` on all `gcc` invocations, see the note above):

```
ccache gcc {{path/to/file.c}}
```

ccomps

Decompose graphs into their connected components.

Graphviz filters: **`acyclic`**, **`bcomps`**, **`comps`**, **`edgepaint`**, **`gvcolor`**, **`gvpack`**, **`mingle`**, **`nop`**, **`sccmap`**, **`tred`**, & **`unflatten`**.

More information: https://graphviz.org/pdf/ccomps.1.pdf.

- Decompose one or more graphs into their connected components:

```
ccomps {{path/to/input1.gv path/to/input2.gv ...}} > {{path/to/output.gv}}
```

- Print the number of nodes, edges, and connected components in one or more graphs:

```
ccomps -v -s {{path/to/input1.gv path/to/input2.gv ...}}
```

- Write each connected component to numbered filenames based on `output.gv`:

```
ccomps -x -o {{path/to/output.gv}} {{path/to/input1.gv path/to/input2.gv ...}}
```

- Display help:

```
ccomps -?
```

cd

Change the current working directory.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-cd.

- Go to the specified directory:

```
cd {{path/to/directory}}
```

- Go up to the parent of the current directory:

```
cd ..
```

- Go to the home directory of the current user:

```
cd
```

- Go to the home directory of the specified user:

```
cd ~{{username}}
```

- Go to the previously chosen directory:

```
cd -
```

- Go to the root directory:

```
cd /
```

cdecl

Compose and decode C and C++ type declarations.

More information: https://manned.org/cdecl.

- Compose English phrase into C declaration, and create [c]ompilable output (include `;` and `{}`):

```
cdecl -c {{phrase}}
```

- Explain C declaration in English:

```
cdecl explain {{C_declaration}}
```

- Cast a variable to another type:

```
cdecl cast {{variable_name}} to {{type}}
```

- Run in [i]nteractive mode:

```
cdecl -i
```

cdk

A CLI for AWS Cloud Development Kit (CDK).

More information: https://docs.aws.amazon.com/cdk/latest/guide/cli.html.

- List the stacks in the app:

```
cdk ls
```

- Synthesize and print the CloudFormation template for the specified stack(s):

```
cdk synth {{stack_name}}
```

- Deploy one or more stacks:

```
cdk deploy {{stack_name1 stack_name2 ...}}
```

- Destroy one or more stacks:

```
cdk destroy {{stack_name1 stack_name2 ...}}
```

- Compare the specified stack with the deployed stack or a local CloudFormation template:

```
cdk diff {{stack_name}}
```

- Create a new CDK project in the current directory for a specified language:

```
cdk init {{[-l|--language]}} {{language}}
```

- Open the CDK API reference in your browser:

```
cdk docs
```

certutil

Manage keys and certificates in both NSS databases and other NSS tokens.

More information: https://manned.org/certutil.

- Create a [N]ew certificate database in the current [d]irectory:

```
certutil -N -d .
```

- [L]ist all certificates in a database:

```
certutil -L -d .
```

- List all private [K]eys in a database specifying the password [f]ile:

```
certutil -K -d . -f {{path/to/password_file.txt}}
```

- [A]dd the signed certificate to the requesters database specifying a [n]ickname, [t]rust attributes and an [i]nput CRT file:

```
certutil -A -n "{{server_certificate}}" -t ",," -i {{path/to/file.crt}} -d .
```

- Add subject alternative names to a given [c]ertificate with a specific key size ([g]):

```
certutil -S -f {{path/to/password_file.txt}} -d . -t ",," -c "{{server_certificate}}" -n "{{server_name}}" -g {{2048}} -s "CN={{common_name}},O={{organization}}"
```

cf

Manage apps and services on Cloud Foundry.

More information: https://docs.cloudfoundry.org/cf-cli/getting-started.html.

- Log in to the Cloud Foundry API:

```
cf login -a {{api_url}}
```

- Push an app using the default settings:

```
cf push {{app_name}}
```

- View the services available from your organization:

```
cf marketplace
```

- Create a service instance:

```
cf create-service {{service}} {{plan}} {{service_name}}
```

- Connect an application to a service:

```
cf bind-service {{app_name}} {{service_name}}
```

- Run a script whose code is included in the app, but runs independently:

```
cf run-task {{app_name}} "{{script_command}}" --name
{{task_name}}
```

- Start an interactive SSH session with a VM hosting an app:

```
cf ssh {{app_name}}
```

- View a dump of recent app logs:

```
cf logs {{app_name}} --recent
```

cfssl

Cloudflare's PKI and TLS toolkit.

See also: **`openssl`**.

More information: https://github.com/cloudflare/cfssl.

- Show certificate information of a host:

```
cfssl certinfo -domain {{www.google.com}}
```

- Decode certificate information from a file:

```
cfssl certinfo -cert {{path/to/certificate.pem}}
```

- Scan host(s) for SSL/TLS issues:

```
cfssl scan {{host1 host2 ...}}
```

- Display help for a subcommand:

```
cfssl {{genkey|gencsr|certinfo|sign|gencrl|ocspdump|
ocsprefresh|ocspsign|ocspserve|scan|bundle|crl|print-
defaults|revoke|gencert|serve|version|selfsign|info}} -h
```

chafa

Image printing in the terminal.

See also: **`catimg`**, **`pixterm`**.

More information: https://hpjansson.org/chafa/man.

- Render an image directly in the terminal:

```
chafa {{path/to/file}}
```

- Render an image with 24-bit color:

```
chafa {{[-c|--colors]}} full {{path/to/file}}
```

- Improve image rendering with small color palettes using dithering:

```
chafa {{[-c|--colors]}} 16 --dither ordered {{path/to/file}}
```

- Render an image, making it appear pixelated:

```
chafa --symbols vhalf {{path/to/file}}
```

- Render a monochrome image with only braille characters:

```
chafa {{[-c|--colors]}} none --symbols braille {{path/to/
file}}
```

chainctl

The official CLI for Chainguard.

More information: https://edu.chainguard.dev/chainguard/chainctl/chainctl-docs/chainctl/.

- Authenticate to the Chainguard Platform:

  ```
  chainctl auth login
  ```

- Logout from the Chainguard Platform:

  ```
  chainctl auth logout
  ```

- Update to the latest version:

  ```
  chainctl update
  ```

- List images available to your account:

  ```
  chainctl images list
  ```

- List image repositories available to your account:

  ```
  chainctl images repos list
  ```

- Examine the history of an image tag in chainctl (e.g., image=python tag=3):

  ```
  chainctl images history {{image}}:{{tag}}
  ```

- List package version data from repositories available to your account (e.g., package_name=go):

  ```
  chainctl packages versions list {{package_name}}
  ```

- Display version:

  ```
  chainctl version
  ```

charm

Set of tools that makes adding a backend to your terminal-based applications, without worrying about user accounts, data storage and encryption.

More information: https://github.com/charmbracelet/charm.

- Backup your Charm account keys:

```
charm backup-keys
```

- Backup Charm account keys to a specific location:

```
charm backup-keys {{[-o|--output]}} {{path/to/output_file.tar}}
```

- Import previously backed up Charm account keys:

```
charm import-keys "{{charm-keys-backup.tar}}"
```

- Find where your `cloud.charm.sh` folder resides on your machine:

```
charm where
```

- Start your Charm server:

```
charm serve
```

- Print linked SSH keys:

```
charm keys
```

- Print your Charm ID:

```
charm id
```

chars

Display names and codes for various ASCII and Unicode characters and code points.

More information: https://github.com/boinkor-net/chars#running.

- Look up a character by its value:

```
chars '{{ß}}'
```

- Look up a character by its Unicode code point:

```
chars {{U+1F63C}}
```

- Look up possible characters given an ambiguous code point:

```
chars {{10}}
```

- Look up a control character:

```
chars "{{^C}}"
```

chatgpt

Shell script to use OpenAI's ChatGPT and DALL-E from the terminal.

More information: https://github.com/0xacx/chatGPT-shell-cli.

- Start in chat mode:

```
chatgpt
```

- Give a prompt to answer to:

```
chatgpt {{[-p|--prompt]}} "{{What is the regex to match an email address?}}"
```

- Start in chat mode using a specific model (default is `gpt-3.5-turbo`):

```
chatgpt {{[-m|--model]}} {{gpt-4}}
```

- Start in chat mode with an initial prompt:

```
chatgpt {{[-i|--init-prompt]}} "{{You are Rick, from Rick and Morty. Respond to questions using his mannerism and include insulting jokes.}}"
```

- Pipe the result of a command to `chatgpt` as a prompt:

```
echo "{{How to view running processes on Ubuntu?}}" | chatgpt
```

- Generate an image using DALL-E:

```
chatgpt {{[-p|--prompt]}} "{{image: A white cat}}"
```

cheat

Create and view interactive cheat sheets.

More information: https://github.com/cheat/cheat.

- Show example usage of a command:

```
cheat {{command}}
```

- Edit the cheat sheet for a command:

```
cheat {{[-e|--edit]}} {{command}}
```

- List the available cheat sheets:

```
cheat {{[-l|--list]}}
```

- Search available the cheat sheets for a specified command name:

```
cheat {{[-s|--search]}} {{command}}
```

- Display version:

```
cheat {{[-v|--version]}}
```

cheatshh

CLI cheatsheet with customized descriptions, tldr and groups, to look into for your reference.

Press **<Enter>** to a command to copy it to your clipboard and exit.

More information: https://github.com/AnirudhG07/cheatshh/blob/main/docs/man/cheatshh.1.

- Add a new command to the cheatshheet:

  `cheatshh {{[-a|--add]}}`

- Edit an existing command's description or group in the cheatshheet:

  `cheatshh {{[-ec|--edit-command]}}`

- Delete an existing command from the cheatshheet:

  `cheatshh {{[-dc|--delete-command]}}`

- Create a new group:

  `cheatshh {{[-g|--group]}}`

- Edit an existing group's name or description in the cheatsheet:

  `cheatshh {{[-eg|--edit-group]}}`

- Delete an existing group and it's sub commands from commands.json file:

  `cheatshh {{[-dg|--delete-group]}}`

- Display man pages after tldr in the preview:

  `cheatshh {{[-m|--man]}}`

checkov

Checkov is a static code analysis tool for Infrastructure as Code (IaC).

It is also a software composition analysis (SCA) tool for images and open source packages.

More information: https://www.checkov.io/1.Welcome/Quick%20Start.html.

- Scan a directory containing IaC (Terraform, Cloudformation, ARM, Ansible, Bicep, Dockerfile, etc):

  ```
  checkov --directory {{path/to/directory}}
  ```

- Scan an IaC file, omitting code blocks in the output:

  ```
  checkov --compact --file {{path/to/file}}
  ```

- List all checks for all IaC types:

  ```
  checkov --list
  ```

checksec

Check security properties of executables.

More information: https://github.com/slimm609/checksec.sh.

- List security properties of an executable binary file:

```
checksec --file={{path/to/binary}}
```

- List security properties recursively of all executable files in a directory:

```
checksec --dir={{path/to/directory}}
```

- List security properties of a process:

```
checksec --proc={{pid}}
```

- List security properties of the running kernel:

```
checksec --kernel
```

chezmoi

A multi-machine dotfile manager, written in Go.

See also: **`stow`**, **`tuckr`**, **`vcsh`**, **`homeshick`**.

More information: https://chezmoi.io.

- Setup up `chezmoi`, creating a Git repository in `~/.local/share/chezmoi`:

  ```
  chezmoi init
  ```

- Set up `chezmoi` from existing dotfiles of a Git repository:

  ```
  chezmoi init {{repository_url}}
  ```

- Start tracking one or more dotfiles:

  ```
  chezmoi add {{path/to/dotfile1 path/to/dotfile2 ...}}
  ```

- Update repository with local changes:

  ```
  chezmoi re-add {{path/to/dotfile1 path/to/dotfile2 ...}}
  ```

- Edit the source state of a tracked dotfile:

  ```
  chezmoi edit {{path/to/dotfile_or_symlink}}
  ```

- See pending changes:

  ```
  chezmoi diff
  ```

- Apply the changes:

  ```
  chezmoi -v apply
  ```

- Pull changes from a remote repository and apply them:

  ```
  chezmoi update
  ```

chgrp

Change group ownership of files and directories.

More information: https://www.gnu.org/software/coreutils/manual/html_node/chgrp-invocation.html.

- Change the owner group of a file/directory:

```
chgrp {{group}} {{path/to/file_or_directory}}
```

- Recursively change the owner group of a directory and its contents:

```
chgrp {{[-R|--recursive]}} {{group}} {{path/to/directory}}
```

- Change the owner group of a symbolic link:

```
chgrp {{[-h|--no-dereference]}} {{group}} {{path/to/symlink}}
```

- Change the owner group of a file/directory to match a reference file:

```
chgrp --reference {{path/to/reference_file}} {{path/to/file_or_directory}}
```

chisel

Create TCP/UDP tunnels, transported over HTTP, secured via SSH.

Includes both client and server in the same **`chisel`** executable.

More information: https://github.com/jpillora/chisel.

- Run a Chisel server:

```
chisel server
```

- Run a Chisel server listening to a specific port:

```
chisel server {{[-p|--port]}} {{server_port}}
```

- Run a chisel server that accepts authenticated connections using username and password:

```
chisel server --auth {{username}}:{{password}}
```

- Connect to a Chisel server and tunnel a specific port to a remote server and port:

```
chisel client {{server_ip}}:{{server_port}} {{local_port}}:
{{remote_server}}:{{remote_port}}
```

- Connect to a Chisel server and tunnel a specific host and port to a remote server and port:

```
chisel client {{server_ip}}:{{server_port}} {{local_host}}:
{{local_port}}:{{remote_server}}:{{remote_port}}
```

- Connect to a Chisel server using username and password authentication:

```
chisel client --auth {{username}}:{{password}} {{server_ip}}:
{{server_port}} {{local_port}}:{{remote_server}}:
{{remote_port}}
```

- Initialize a Chisel server in reverse mode on a specific port, also enabling SOCKS5 proxy (on port 1080) functionality:

```
chisel server {{[-p|--port]}} {{server_port}} --reverse --
socks5
```

- Connect to a Chisel server at specific IP and port, creating a reverse tunnel mapped to a local SOCKS proxy:

```
chisel client {{server_ip}}:{{server_port}} R:socks
```

chkfont

Verify the format of a FIGlet font file (**`.flf`**).

See also: **`figlet`**, **`figlist`**, **`showfigfonts`**.

More information: https://manned.org/chkfont.

- Check a font file for formatting errors:

```
chkfont {{path/to/font.flf}}
```

- Check all `.flf` font files in a directory:

```
chkfont {{path/to/fonts}}/*.flf
```

chkrootkit

Scan system for rootkits.

More information: https://manned.org/chkrootkit.

- Enable [q]uiet mode and suppress normal test results:

```
chkrootkit -q
```

- Enable e[x]pert mode and produce additional outputs:

```
chkrootkit -x
```

- Enable [d]ebug mode to show all output:

```
chkrootkit -d
```

- Specify [e]xcluded files for some tests:

```
chkrootkit -e "{{path/to/file}}"
```

- Specify a directory as the [r]oot for testing (e.g. mounted `ext` drives):

```
chkrootkit -r {{path/to/directory}}
```

- Ignore [n]fs-mounted directories:

```
chkrootkit -n
```

- Invoke [T]ests and ignore specific filesystem types:

```
chkrootkit -T {{filesystemtype}}
```

- Generate [l]ist of available tests:

```
chkrootkit -l
```

chmod

Change the access permissions of a file or directory.

More information: https://www.gnu.org/software/coreutils/manual/html_node/chmod-invocation.html.

- Give the [u]ser who owns a file the right to e[x]ecute it:

```
chmod u+x {{path/to/file}}
```

- Give the [u]ser rights to [r]ead and [w]rite to a file/directory:

```
chmod u+rw {{path/to/file_or_directory}}
```

- Remove e[x]ecutable rights from the [g]roup:

```
chmod g-x {{path/to/file}}
```

- Give [a]ll users rights to [r]ead and e[x]ecute:

```
chmod a+rx {{path/to/file}}
```

- Give [o]thers (not in the file owner's group) the same rights as the [g]roup:

```
chmod o=g {{path/to/file}}
```

- Remove all rights from [o]thers:

```
chmod o= {{path/to/file}}
```

- Change permissions recursively giving [g]roup and [o]thers the ability to [w]rite:

```
chmod {{[-R|--recursive]}} g+w,o+w {{path/to/directory}}
```

- Recursively give [a]ll users [r]ead permissions to files. Also give e[X]ecute permissions to files that have at least one execution permission and to all sub-directories:

```
chmod {{[-R|--recursive]}} a+rX {{path/to/directory}}
```

choose

A human-friendly and fast alternative to cut and (sometimes) awk.

More information: https://github.com/theryangeary/choose#usage.

- Print the 5th item from a line (starting from 0):

```
choose {{4}}
```

- Print the first, 3rd, and 5th item from a line, where items are separated by ':' instead of whitespace:

```
choose {{[-f|--field-separator]}} '{{:}}' {{0}} {{2}} {{4}}
```

- Print everything from the 2nd to 5th item on the line, including the 5th:

```
choose {{1}}:{{4}}
```

- Print everything from the 2nd to 5th item on the line, excluding the 5th:

```
choose {{[-x|--exclusive]}} {{1}}:{{4}}
```

- Print the beginning of the line to the 3rd item:

```
choose :{{2}}
```

- Print all items from the beginning of the line until the 3rd item (exclusive):

```
choose {{[-x|--exclusive]}} :{{2}}
```

- Print all items from the 3rd to the end of the line:

```
choose {{2}}:
```

- Print the last item from a line:

```
choose {{-1}}
```

chown

Change user and group ownership of files and directories.

More information: https://www.gnu.org/software/coreutils/manual/html_node/chown-invocation.html.

- Change the owner user of a file/directory:

```
sudo chown {{user}} {{path/to/file_or_directory}}
```

- Change the owner user and group of a file/directory:

```
sudo chown {{user}}:{{group}} {{path/to/file_or_directory}}
```

- Change the owner user and group to both have the name `user`:

```
sudo chown {{user}}: {{path/to/file_or_directory}}
```

- Recursively change the owner of a directory and its contents:

```
sudo chown {{[-R|--recursive]}} {{user}} {{path/to/directory}}
```

- Change the owner of a symbolic link:

```
sudo chown {{[-h|--no-dereference]}} {{user}} {{path/to/symlink}}
```

- Change the owner of a file/directory to match a reference file:

```
sudo chown --reference {{path/to/reference_file}} {{path/to/file_or_directory}}
```

chroma

A general-purpose syntax highlighter.

The **`--lexer`** option is usually unnecessary, as it will be automatically determined based on the file extension.

More information: https://github.com/alecthomas/chroma.

- Highlight source code from a file with the Python lexer and output to `stdout`:

```
chroma --lexer {{python}} {{path/to/source_file.py}}
```

- Highlight source code from a file with the Go lexer and output to an HTML file:

```
chroma --lexer {{go}} --formatter {{html}} {{path/to/
source_file.go}} > {{path/to/target_file.html}}
```

- Highlight source code from `stdin` with the C++ lexer and output to an SVG file, using the Monokai style:

```
{{command}} | chroma --lexer {{c++}} --formatter {{svg}} --
style {{monokai}} > {{path/to/target_file.svg}}
```

- List available lexers, styles and formatters:

```
chroma --list
```

chromium

Open-source web browser principally developed and maintained by Google.

Note: You may need to replace the **`chromium`** command with your desired web browser, such as **`brave`**, **`google-chrome`**, **`opera`**, or **`vivaldi`**.

More information: https://www.chromium.org/developers/how-tos/run-chromium-with-flags/.

- Open a specific URL or file:

```
chromium {{https://example.com|path/to/file.html}}
```

- Open in incognito mode:

```
chromium --incognito {{example.com}}
```

- Open in a new window:

```
chromium --new-window {{example.com}}
```

- Open in application mode (without toolbars, URL bar, buttons, etc.):

```
chromium --app={{https://example.com}}
```

- Use a proxy server:

```
chromium --proxy-server="{{socks5://hostname:
66}}" {{example.com}}
```

- Open with a custom profile directory:

```
chromium --user-data-dir={{path/to/directory}}
```

- Open without CORS validation (useful to test an API):

```
chromium --user-data-dir={{path/to/directory}} --disable-web-
security
```

- Open with a DevTools window for each tab opened:

```
chromium --auto-open-devtools-for-tabs
```

chronic

Display **`stdout`** and **`stderr`** of a command if and only if it fails.

More information: https://manned.org/chronic.

- Display `stdout` and `stderr` of the specified command if and only if it produces a non-zero exit code or crashes:

```
chronic {{command}} {{option1 option2 ...}}
```

- Display `stdout` and `stderr` of the specified command if and only if it produces a non-empty `stderr`:

```
chronic -e {{command}} {{option1 option2 ...}}
```

- Enable [v]erbose mode:

```
chronic -v {{command}} {{option1 option2 ...}}
```

chronyc

Query the Chrony NTP daemon.

More information: https://chrony-project.org/doc/4.6.1/chronyc.html.

- Start `chronyc` in interactive mode:

  `chronyc`

- Display tracking stats for the Chrony daemon:

  `chronyc tracking`

- Print the time sources that Chrony is currently using:

  `chronyc sources`

- Display stats for sources currently used by chrony daemon as a time source:

  `chronyc sourcestats`

- Step the system clock immediately, bypassing any slewing:

  `chronyc makestep`

- Display verbose information about each NTP source:

  `chronyc ntpdata`

chroot

Run command or interactive shell with special root directory.

More information: https://www.gnu.org/software/coreutils/manual/html_node/chroot-invocation.html.

- Run command as new root directory:

```
sudo chroot {{path/to/new_root}} {{command}}
```

- Use a specific user and group:

```
sudo chroot --userspec {{username_or_id:group_name_or_id}}
```

cidr

Simplifies IPv4/IPv6 CIDR network prefix management with counting, overlap checking, explanation, and subdivision.

More information: https://github.com/bschaatsbergen/cidr.

- Explain a CIDR range:

  ```
  cidr explain {{10.0.0.0/16}}
  ```

- Check whether an address belongs to a CIDR range:

  ```
  cidr contains {{10.0.0.0/16}} {{10.0.14.5}}
  ```

- Get a count of all addresses in a CIDR range:

  ```
  cidr count {{10.0.0.0/16}}
  ```

- Check whether two CIDR ranges overlap:

  ```
  cidr overlaps {{10.0.0.0/16}} {{10.0.14.0/22}}
  ```

- Divide a CIDR range into a specific number of networks:

  ```
  cidr divide {{10.0.0.0/16}} {{9}}
  ```

circo

Render an image of a **circular** network graph from a **graphviz** file.

Layouts: **dot**, **neato**, **twopi**, **circo**, **fdp**, **sfdp**, **osage** & **patchwork**.

More information: https://graphviz.org/doc/info/command.html.

- Render a PNG image with a filename based on the input filename and output format (uppercase -O):

```
circo -T {{png}} -O {{path/to/input.gv}}
```

- Render a SVG image with the specified output filename (lowercase -o):

```
circo -T {{svg}} -o {{path/to/image.svg}} {{path/to/input.gv}}
```

- Render the output in PS, PDF, SVG, Fig, PNG, GIF, JPEG, JSON, or DOT format:

```
circo -T {{format}} -O {{path/to/input.gv}}
```

- Render a GIF image using `stdin` and `stdout`:

```
echo "{{digraph {this -> that} }}" | circo -T {{gif}} > {{path/to/image.gif}}
```

- Display help:

```
circo -?
```

circup

The CircuitPython library updater.

More information: https://github.com/adafruit/circup.

- Interactively update modules on a device:

  ```
  circup update
  ```

- Install a new library:

  ```
  circup install {{library_name}}
  ```

- Search for a library:

  ```
  circup show {{partial_name}}
  ```

- List all libraries on a connected device in `requirements.txt` format:

  ```
  circup freeze
  ```

- Save all libraries on a connected device in `requirements.txt` in current directory:

  ```
  circup freeze {{[-r|--requirement]}}
  ```

cjxl

Compress images to JPEG XL.

Accepted input extensions are PNG, APNG, GIF, JPEG, EXR, PPM, PFM, PAM, PGX, and JXL.

More information: https://github.com/libjxl/libjxl.

- Convert an image to JPEG XL:

```
cjxl {{path/to/image.ext}} {{path/to/output.jxl}}
```

- Set quality to lossless and maximize compression of the resulting image:

```
cjxl --distance 0 --effort 9 {{path/to/image.ext}} {{path/to/output.jxl}}
```

- Display an extremely detailed help page:

```
cjxl {{[-h -v -v -v -v|--help --verbose --verbose --verbose --verbose]}}
```

cksum

Calculate CRC checksums and byte counts of a file.

Note: On old UNIX systems the CRC implementation may differ.

More information: https://www.gnu.org/software/coreutils/manual/html_node/cksum-invocation.html.

- Display a 32-bit checksum, size in bytes and filename:

```
cksum {{path/to/file}}
```

ClamAV

Open-source anti-virus program.

ClamAV isn't a command, but a set of commands.

More information: https://www.clamav.net.

- View documentation for scanning files using the `clamd` daemon:

  `tldr clamdscan`

- View documentation for scanning files without the `clamd` daemon running:

  `tldr clamscan`

- View documentation for updating the virus definitions:

  `tldr freshclam`

clamdscan

Scan for viruses using the ClamAV Daemon.

More information: https://docs.clamav.net/manual/Usage/Scanning.html#clamdscan.

- Scan a file or directory for vulnerabilities:

```
clamdscan {{path/to/file_or_directory}}
```

- Scan data from `stdin`:

```
{{command}} | clamdscan -
```

- Scan the current directory and output only infected files:

```
clamdscan --infected
```

- Print the scan report to a log file:

```
clamdscan --log {{path/to/log_file}}
```

- Move infected files to a specific directory:

```
clamdscan --move {{path/to/quarantine_directory}}
```

- Remove infected files:

```
clamdscan --remove
```

- Use multiple threads to scan a directory:

```
clamdscan --multiscan
```

- Pass the file descriptor instead of streaming the file to the daemon:

```
clamdscan --fdpass
```

clamscan

A virus scanner.

More information: https://docs.clamav.net/manual/Usage/Scanning.html#clamscan.

- Scan a file for vulnerabilities:

```
clamscan {{path/to/file}}
```

- Scan all files recursively in a specific directory:

```
clamscan {{[-r|--recursive]}} {{path/to/directory}}
```

- Scan data from `stdin`:

```
{{command}} | clamscan -
```

- Specify a virus database file or directory of files:

```
clamscan {{[-d|--database]}} {{path/to/
database_file_or_directory}}
```

- Scan the current directory and output only infected files:

```
clamscan {{[-i|--infected]}}
```

- Print the scan report to a log file:

```
clamscan {{[-l|--log]}} {{path/to/log_file}}
```

- Move infected files to a specific directory:

```
clamscan --move {{path/to/quarantine_directory}}
```

- Remove infected files:

```
clamscan --remove yes
```

clang++

Compile C++ source files.

Part of LLVM.

More information: https://clang.llvm.org.

- Compile a set of source code files into an executable binary:

```
clang++ {{path/to/source1.cpp path/to/source2.cpp ...}} {{[-o|--output]}} {{path/to/output_executable}}
```

- Activate output of all errors and warnings:

```
clang++ {{path/to/source.cpp}} -Wall {{[-o|--output]}} {{output_executable}}
```

- Show common warnings, debug symbols in output, and optimize without affecting debugging:

```
clang++ {{path/to/source.cpp}} -Wall {{[-g|--debug]}} -Og {{[-o|--output]}} {{path/to/output_executable}}
```

- Choose a language standard to compile for:

```
clang++ {{path/to/source.cpp}} -std={{c++20}} {{[-o|--output]}} {{path/to/output_executable}}
```

- Include libraries located at a different path than the source file:

```
clang++ {{path/to/source.cpp}} {{[-o|--output]}} {{path/to/output_executable}} -I{{path/to/header_path}} -L{{path/to/library_path}} -l{{path/to/library_name}}
```

- Compile source code into LLVM Intermediate Representation (IR):

```
clang++ {{[-S|--assemble]}} -emit-llvm {{path/to/source.cpp}} {{[-o|--output]}} {{path/to/output.ll}}
```

- Optimize the compiled program for performance:

```
clang++ {{path/to/source.cpp}} -O{{1|2|3|fast}} {{[-o|--output]}} {{path/to/output_executable}}
```

- Display version:

```
clang++ --version
```

clang-check

Check basic errors and work with Clang's Abstract Syntax Tree (AST).

Part of Clang's LibTooling and is useful for debugging and analyzing C/C++ code.

More information: https://manned.org/clang-check.

- Run default checks on a source file:

```
clang-check {{path/to/file.cpp}} --
```

- Dump the Abstract Syntax Tree for debugging:

```
clang-check {{path/to/file.cpp}} -ast-dump --
```

- Filter AST by Name:

```
clang-check {{path/to/file.cpp}} -ast-dump -ast-dump-filter
FunctionName
```

- Pretty-Print AST:

```
clang-check {{path/to/file.cpp}} -ast-print --
```

clang-cpp

This command is an alias of **`clang++`**.

- View documentation for the original command:

```
tldr clang++
```

clang-format

Auto-format C/C++/Java/JavaScript/Objective-C/Protobuf/C# code.

More information: https://clang.llvm.org/docs/ClangFormat.html.

- Format a file and print the result to `stdout`:

```
clang-format {{path/to/file}}
```

- Format a file in-place:

```
clang-format -i {{path/to/file}}
```

- Format a file using a predefined coding style:

```
clang-format --style {{LLVM|GNU|Google|Chromium|Microsoft|
Mozilla|WebKit}} {{path/to/file}}
```

- Format a file using the `.clang-format` file in one of the parent directories of the source file:

```
clang-format --style=file {{path/to/file}}
```

- Generate a custom `.clang-format` file:

```
clang-format --style {{LLVM|GNU|Google|Chromium|Microsoft|
Mozilla|WebKit}} --dump-config > {{.clang-format}}
```

clang-tidy

An LLVM-based C/C++ linter to find style violations, bugs and security flaws through static analysis.

More information: https://clang.llvm.org/extra/clang-tidy/.

- Run default checks on a source file:

```
clang-tidy {{path/to/file.cpp}}
```

- Don't run any checks other than the `cppcoreguidelines` checks on a file:

```
clang-tidy {{path/to/file.cpp}} -checks={{-
*,cppcoreguidelines-*}}
```

- List all available checks:

```
clang-tidy -checks={{*}} -list-checks
```

- Specify defines and includes as compilation options (after `--`):

```
clang-tidy {{path/to/file.cpp}} -- -I{{my_project/include}} -
D{{definitions}}
```

clang

Compile C, C++, and Objective-C source files. Can be used as a drop-in replacement for GCC.

Part of LLVM.

More information: https://clang.llvm.org/docs/ClangCommandLineReference.html.

- Compile multiple source files into an executable:

```
clang {{path/to/source1.c path/to/source2.c ...}} {{[-o|--output]}} {{path/to/output_executable}}
```

- Activate output of all errors and warnings:

```
clang {{path/to/source.c}} -Wall {{[-o|--output]}} {{output_executable}}
```

- Show common warnings, debug symbols in output, and optimize without affecting debugging:

```
clang {{path/to/source.c}} -Wall {{[-g|--debug]}} -Og {{[-o|--output]}} {{path/to/output_executable}}
```

- Include libraries from a different path:

```
clang {{path/to/source.c}} {{[-o|--output]}} {{path/to/output_executable}} -I{{path/to/header}} -L{{path/to/library}} -l{{library_name}}
```

- Compile source code into LLVM Intermediate Representation (IR):

```
clang {{[-S|--assemble]}} -emit-llvm {{path/to/source.c}} {{[-o|--output]}} {{path/to/output.ll}}
```

- Compile source code into an object file without linking:

```
clang {{[-c|--compile]}} {{path/to/source.c}}
```

- Optimize the compiled program for performance:

```
clang {{path/to/source.c}} -O{{1|2|3|fast}} {{[-o|--output]}} {{path/to/output_executable}}
```

- Display version:

```
clang --version
```

clangd

Language server that provides IDE-like features to editors.

It should be used via an editor plugin rather than invoked directly.

More information: https://manpages.ubuntu.com/manpages/man1/clangd.1.html.

- Display available options:

```
clangd --help
```

- List of available options:

```
clangd --help-list
```

- Display version:

```
clangd --version
```

claude

An agent-based coding tool that understands your code base and helps you code faster through natural language commands.

More information: https://docs.anthropic.com/en/docs/claude-code/overview.

- Execute with prompt:

  ```
  claude prompt
  ```

- Update `claude`:

  ```
  claude update
  ```

- Get the list of specified MCP servers:

  ```
  claude mcp list
  ```

- Create commit with command:

  ```
  claude commit
  ```

- Get the list of configurations:

  ```
  claude config list
  ```

clear

Clears the screen of the terminal.

More information: https://manned.org/clear.

- Clear the screen:

```
clear
```

- Clear the screen but keep the terminal's scrollback buffer (equivalent to pressing `<Ctrl l>` in Bash):

```
clear -x
```

- Indicate the type of terminal to clean (defaults to the value of the environment variable `TERM`):

```
clear -T {{type_of_terminal}}
```

- Display the version of `ncurses` used by `clear`:

```
clear -V
```

clementine

A modern music player and library organizer.

See also: **audacious**, **qmmp**, **cmus**, **mpv**.

More information: https://manned.org/clementine.

- Start the GUI or bring it to front:

```
clementine
```

- Start playing music:

```
clementine {{url|path/to/music.ext}}
```

- Toggle between pausing and playing:

```
clementine {{[-t|--play-pause]}}
```

- Stop playback:

```
clementine {{[-s|--stop]}}
```

- Skip to the next or previous track:

```
clementine --{{next|previous}}
```

- Create a new playlist with one or more music files or URLs:

```
clementine {{[-c|--create]}} {{url1 url2 ...|path/to/
music1.ext path/to/music2.ext ...}}
```

- Load a playlist file:

```
clementine {{[-l|--load]}} {{path/to/playlist.ext}}
```

- Play a specific track in the currently loaded playlist:

```
clementine {{[-k|--play-track]}} {{5}}
```

cli4

Python command-line interface for Cloudflare API.

More information: https://github.com/cloudflare/python-cloudflare.

- Display account information:

```
cli4 {{/user}}
```

- List all zones:

```
cli4 {{/zones}}
```

- List DNS records for a specific zone:

```
cli4 {{/zones/:example.com/dns_records}}
```

- Create a new DNS record:

```
cli4 --post {{name=example.com}} {{type=A}}
{{content=192.0.2.1}} {{/zones/:example.com/dns_records}}
```

- Update an existing DNS record:

```
cli4 --put {{name=sub.example.com}} {{type=A}}
{{content=192.0.2.2}} {{/zones/:example.com/
dns_records/:record_id}}
```

- Delete a DNS record:

```
cli4 --delete {{/zones/:example.com/dns_records/:record_id}}
```

- Purge all cache for a zone:

```
cli4 --post {{purge_everything=true}} {{/zones/:example.com/
purge_cache}}
```

cli53

Command line tool for Amazon Route 53.

More information: https://github.com/barnybug/cli53.

- List domains:

```
cli53 list
```

- Create a domain:

```
cli53 create {{example.com}} --comment "{{comment}}"
```

- Export a bind zone file to `stdout`:

```
cli53 export {{example.com}}
```

- Create a `www` subdomain pointing to a relative record in the same zone:

```
cli53 {{[rc|rrcreate]}} {{example.com}} '{{www 300 CNAME
lb}}'
```

- Create a `www` subdomain pointing to an external address (must end with a dot):

```
cli53 {{[rc|rrcreate]}} {{example.com}} '{{www 300 CNAME
lb.example.com.}}'
```

- Create a `www` subdomain pointing to an IP address:

```
cli53 {{[rc|rrcreate]}} {{example.com}} '{{www 300 A
150.130.110.1}}'
```

- Replace a `www` subdomain pointing to a different IP:

```
cli53 {{[rc|rrcreate]}} --replace '{{www 300 A
150.130.110.2}}'
```

- Delete a record A:

```
cli53 {{[rd|rrdelete]}} {{example.com}} {{www}} {{A}}
```

clido

Save-state TODO app for the terminal.

More information: https://codeberg.org/Oglo12/clido/wiki.

- Create a list:

```
clido --new {{name}}
```

- Load a list:

```
clido --load {{name}}
```

- Delete a list:

```
clido --remove {{name}}
```

- List all lists:

```
clido --lists
```

- Toggle autowrite:

```
clido toggle-autowrite
```

- Open a list in a text editor:

```
clido edit {{text_editor}}
```

- Display help:

```
clido -h
```

- Display version:

```
clido -v
```

clifm

The command-line file manager.

See also: **`vifm`**, **`ranger`**, **`mc`**, **`nautilus`**.

More information: https://github.com/leo-arch/clifm.

- Start CliFM:

  `clifm`

- Open the file or directory whose ELN (entry list number) is 12:

  `12`

- Create a new file and a new directory:

  `<n>file dir/`

- Search for PDF files in the current directory:

  `</>*.pdf`

- Select all PNG files in the current directory:

  `<s> *.png`

- Remove the previously selected files (use `<t>` to send the files to the recycle bin instead):

  `<r>sel`

- Display help:

  `<?>`

- Exit CliFM:

  `<q>`

clip-view

Command Line Interface Pages render.

Render for a TlDr-like project with much a more extensive syntax and several render modes.

More information: https://github.com/command-line-interface-pages/v2-tooling/tree/main/clip-view.

- Render specific local pages:

```
clip-view {{path/to/page1.clip path/to/page2.clip ...}}
```

- Render specific remote pages:

```
clip-view {{page_name1 page_name2 ...}}
```

- Render pages by a specific render:

```
clip-view --render {{tldr|tldr-colorful|docopt|docopt-
colorful}} {{page_name1 page_name2 ...}}
```

- Render pages with a specific color theme:

```
clip-view --theme {{path/to/local_theme.yaml|
remote_theme_name}} {{page_name1 page_name2 ...}}
```

- Clear a page or theme cache:

```
clip-view --clear-{{page|theme}}-cache
```

- Display help:

```
clip-view --help
```

- Display version:

```
clip-view --version
```

clj

Clojure tool to start a REPL or invoke a function with data.

All options can be defined in a **`deps.edn`** file.

More information: https://clojure.org/guides/deps_and_cli.

- Start a REPL (interactive shell):

  ```
  clj
  ```

- Execute a function:

  ```
  clj -X {{namespace/function_name}}
  ```

- Run the main function of a specified namespace:

  ```
  clj -M {{[-m|--main]}} {{namespace}} {{args}}
  ```

- Prepare a project by resolving dependencies, downloading libraries, and making/caching classpaths:

  ```
  clj -P
  ```

- Start an nREPL server with the CIDER middleware:

  ```
  clj -Sdeps '{:deps {nrepl {:mvn/version "0.7.0"} cider/cider-nrepl {:mvn/version "0.25.2"}}}' {{[-m|--main]}} nrepl.cmdline --middleware '["cider.nrepl/cider-middleware"]' --interactive
  ```

- Start a REPL for ClojureScript and open a web browser:

  ```
  clj -Sdeps '{:deps {org.clojure/clojurescript {:mvn/version "1.10.758"}}}' {{[-m|--main]}} cljs.main {{[-r|--repl]}}
  ```

cloc

Count lines of code.

More information: https://github.com/AlDanial/cloc#options-.

- Count all the lines of code in a directory:

  `cloc {{path/to/directory}}`

- Compare 2 directory structures and count the differences between them:

  `cloc --diff {{path/to/directory1}} {{path/to/directory2}}`

- Ignore files that are ignored by VCS, such as files specified in `.gitignore`:

  `cloc --vcs git {{path/to/directory}}`

- Display the results for each file instead of each language:

  `cloc --by-file {{path/to/directory}}`

clockwork-cli

Interface for the Clockwork PHP debugging framework.

More information: https://github.com/ptrofimov/clockwork-cli.

- Monitor Clockwork logs for the current project:

  ```
  clockwork-cli
  ```

- Monitor Clockwork logs for a specific project:

  ```
  clockwork-cli {{path/to/directory}}
  ```

- Monitor Clockwork logs for multiple projects:

  ```
  clockwork-cli {{path/to/directory1 path/to/directory2 ...}}
  ```

clojure

This command is an alias of **`clj`**.

- View documentation for the original command:

  ```
  tldr clj
  ```

cloudflared

Create a persistent connection to the Cloudflare network.

More information: https://developers.cloudflare.com/argo-tunnel/.

- Authenticate and associate the connection to a domain in the Cloudflare account:

```
cloudflared tunnel login
```

- Create a tunnel with a specific name:

```
cloudflared tunnel create {{name}}
```

- Establish a tunnel to a host in Cloudflare from the local server:

```
cloudflared tunnel --hostname {{hostname}} localhost:
{{port_number}}
```

- Establish a tunnel to a host in Cloudflare from the local server, without verifying the local server's certificate:

```
cloudflared tunnel --hostname {{hostname}} localhost:
{{port_number}} --no-tls-verify
```

- Save logs to a file:

```
cloudflared tunnel --hostname {{hostname}} http://localhost:
{{port_number}} --loglevel {{panic|fatal|error|warn|info|
debug}} --logfile {{path/to/file}}
```

- Install cloudflared as a system service:

```
cloudflared service install
```

cluster

Find clusters in a graph and augment the graph with this information.

More information: https://graphviz.org/pdf/cluster.1.pdf.

- Generate clusters that optimize modularity and print the result to `stdout`:

```
cluster {{input.gv}}
```

- Specify a target number of [C]lusters (approximate) to generate (0 by default, meaning a number that approximately optimizes the modularity):

```
cluster -C {{5}} {{input.gv}}
```

- Use a different [c]lustering method (0: modularity clustering, 1: modularity quality):

```
cluster -c {{0|1}} {{input.gv}}
```

- Save the [o]utput to a file:

```
cluster -o {{output.gv}} {{input.gv}}
```

- Enable [v]erbose mode:

```
cluster -v {{input.gv}}
```

clusterdb

Cluster (reorganize) a PostgreSQL database.

More information: https://www.postgresql.org/docs/current/app-clusterdb.html.

- Cluster a specific database:

```
clusterdb {{database_name}}
```

- Cluster all databases:

```
clusterdb {{[-a|--all]}}
```

- Cluster a specific table in a database:

```
clusterdb {{[-t|--table]}} {{table_name}} {{database_name}}
```

cmake

Cross-platform build automation system, that generates recipes for native build systems.

More information: https://cmake.org/cmake/help/latest/manual/cmake.1.html.

- Generate a build recipe in the current directory with `CMakeLists.txt` from a project directory:

```
cmake {{path/to/project_directory}}
```

- Use a generated recipe in a given directory to build artifacts:

```
cmake --build {{path/to/build_directory}}
```

- Install the build artifacts into `/usr/local/` and strip debugging symbols:

```
cmake --install {{path/to/build_directory}} --strip
```

- Generate a build recipe, with build type set to `Release` with CMake variable:

```
cmake {{path/to/project_directory}} -D
CMAKE_BUILD_TYPE=Release
```

- Generate a build recipe using `generator_name` as the underlying build system:

```
cmake -G {{generator_name}} {{path/to/project_directory}}
```

- Install the build artifacts using a custom prefix for paths:

```
cmake --install {{path/to/build_directory}} --strip --prefix
{{path/to/directory}}
```

- Run a custom build target:

```
cmake --build {{path/to/build_directory}} {{[-t|--target]}}
{{target_name}}
```

- Display help:

```
cmake {{[-h|--help]}}
```

cmark

Convert CommonMark Markdown formatted text to other formats.

More information: https://github.com/commonmark/cmark.

- Render a CommonMark Markdown file to HTML:

```
cmark --to html {{filename.md}}
```

- Convert data from `stdin` to LaTeX:

```
cmark --to latex
```

- Convert straight quotes to smart quotes:

```
cmark --smart --to html {{filename.md}}
```

- Validate UTF-8 characters:

```
cmark --validate-utf8 {{filename.md}}
```

cmatrix

Shows a scrolling Matrix like screen in the terminal.

See also: **`cbonsai`**.

More information: https://manned.org/cmatrix.

- Enable [a]synchronous scrolling:

```
cmatrix -a
```

- Change the text [C]olor (green by default):

```
cmatrix -C {{red}}
```

- Enable [r]ainbow mode:

```
cmatrix -r
```

- Use a screen [u]pdate delay of 100 centiseconds (1 second):

```
cmatrix -u 100
```

- Use [s]creensaver mode where the first keystroke exits:

```
cmatrix -s
```

cmctl

Manage cert-manager resources inside your cluster.

Check cert signing status, approve/deny requests, and issue new certificate requests.

More information: https://cert-manager.io/docs/usage/cmctl/.

- Check if the cert-manager API is ready:

```
cmctl check api
```

- Check the status of a certificate:

```
cmctl status certificate {{cert_name}}
```

- Create a new certificate request based on an existing certificate:

```
cmctl create certificaterequest my-cr --from-certificate-file {{cert.yaml}}
```

- Create a new certificate request, fetch the signed certificate, and set a maximum wait time:

```
cmctl create certificaterequest my-cr --from-certificate-file {{cert.yaml}} --fetch-certificate --timeout {{20m}}
```

cmd-wrapped

View shell history statistics.

More information: https://crates.io/crates/cmd-wrapped.

- Fetch current statistics:

```
cmd-wrapped
```

- Fetch statistics from a specific year:

```
cmd-wrapped {{year}}
```

- Specify a shell:

```
cmd-wrapped {{[-s|--shell]}} {{shell}}
```

- Display help:

```
cmd-wrapped {{[-h|--help]}}
```

- Display version:

```
cmd-wrapped {{[-V|--version]}}
```

cmp

Compare two files byte by byte.

More information: https://www.gnu.org/software/diffutils/manual/diffutils.html#Invoking-cmp.

- Output char and line number of the first difference between two files:

```
cmp {{path/to/file1}} {{path/to/file2}}
```

- Output info of the first difference: char, line number, bytes, and values:

```
cmp {{[-b|--print-bytes]}} {{path/to/file1}} {{path/to/file2}}
```

- Output the byte numbers and values of every difference:

```
cmp {{[-l|--verbose]}} {{path/to/file1}} {{path/to/file2}}
```

- Compare files but output nothing, yield only the exit status:

```
cmp {{[-s|--quiet]}} {{path/to/file1}} {{path/to/file2}}
```

cmuwmtopbm

Convert a CMU window manager bitmap to a PBM image.

See also: **pbmtocmuwm**.

More information: https://netpbm.sourceforge.net/doc/cmuwmtopbm.html.

- Convert a CMU window manager bitmap to a PBM image:

```
cmuwmtopbm {{path/to/image.pbm}} > {{path/to/output.bmp}}
```

code

Cross platform and extensible code editor.

More information: https://code.visualstudio.com/docs/configure/command-line.

- Start Visual Studio Code:

```
code
```

- Open specific files/directories:

```
code {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Compare two specific files:

```
code {{[-d|--diff]}} {{path/to/file1}} {{path/to/file2}}
```

- Open specific files/directories in a new window:

```
code {{[-n|--new-window]}} {{path/to/file_or_directory1 path/
to/file_or_directory2 ...}}
```

- Install/uninstall a specific extension:

```
code --{{install|uninstall}}-extension
{{publisher.extension}}
```

- Display diagnostic and process information about the running code window:

```
code {{[-s|--status]}}
```

- Print installed extensions with their versions:

```
code --list-extensions --show-versions
```

- Start the editor as a superuser (root) while storing user data in a specific directory:

```
sudo code --user-data-dir {{path/to/directory}}
```

codecrafters

Practice writing complex software.

More information: https://docs.codecrafters.io/cli/usage.

- Run tests without committing changes:

  ```
  codecrafters test
  ```

- Run tests for all previous stages and the current stage without committing changes:

  ```
  codecrafters test --previous
  ```

- Commit changes and submit, to move to the next stage:

  ```
  codecrafters submit
  ```

- Update language version:

  ```
  codecrafters update-buildpack
  ```

codespell

Spellchecker for source code.

More information: https://github.com/codespell-project/codespell.

- Check for typos in all text files in the current directory, recursively:

```
codespell
```

- Correct all typos found in-place:

```
codespell --write-changes
```

- Skip files with names that match the specified pattern (accepts a comma-separated list of patterns using wildcards):

```
codespell --skip "{{pattern}}"
```

- Use a custom dictionary file when checking (`--dictionary` can be used multiple times):

```
codespell --dictionary {{path/to/file.txt}}
```

- Do not check words that are listed in the specified file:

```
codespell --ignore-words {{path/to/file.txt}}
```

- Do not check the specified words:

```
codespell --ignore-words-list
{{ignored_word1,ignored_word2,...}}
```

- Print 3 lines of context around, before or after each match:

```
codespell --{{context|before-context|after-context}} {{3}}
```

- Check file names for typos, in addition to file contents:

```
codespell --check-filenames
```

codex

Natural language code assistant for the terminal, powered by OpenAI.

Reads and edits files in your current directory to fulfill requests.

More information: https://github.com/openai/codex.

- Start an interactive Codex session in the current directory:

```
codex
```

- Run a single Codex command using a prompt:

```
codex "{{your prompt}}"
```

- Run a prompt with automatic approval of all file edits and commands:

```
codex {{[-a|--approval-mode]}} full-auto "{{your prompt}}"
```

- Use a specific provider and model:

```
codex --provider {{provider_name}} {{[-m|--model]}} {{model_name}} "{{your prompt}}"
```

- Load the entire repository as context (experimental):

```
codex --full-context "{{your prompt}}"
```

- Show the resource usage for the current session (run this command inside a session):

```
/cost
```

- Display help:

```
codex --help
```

coffee

Execute CoffeeScript scripts or compiles them into JavaScript.

More information: https://coffeescript.org#cli.

- Run a script:

```
coffee {{path/to/file.coffee}}
```

- Compile to JavaScript and save to a file with the same name:

```
coffee --compile {{path/to/file.coffee}}
```

- Compile to JavaScript and save to a given output file:

```
coffee --compile {{path/to/file.coffee}} --output {{path/to/file.js}}
```

- Start a REPL (interactive shell):

```
coffee --interactive
```

- Watch script for changes and re-run script:

```
coffee --watch {{path/to/file.coffee}}
```

cola

This command is an alias of **`git-cola`**.

- View documentation for the original command:

```
tldr git-cola
```

colima

Container runtimes for macOS and Linux with minimal setup.

More information: https://github.com/abiosoft/colima.

- Start the daemon in the background:

```
colima start
```

- Create a configuration file and use it:

```
colima start --edit
```

- Start and setup containerd (install `nerdctl` to use containerd via `nerdctl`):

```
colima start --runtime containerd
```

- Start with Kubernetes (`kubectl` is required):

```
colima start --kubernetes
```

- Customize CPU count, RAM memory and disk space (in GiB):

```
colima start --cpu {{number}} --memory {{memory}} --disk {{storage_space}}
```

- Use Docker via Colima (Docker is required):

```
colima start
```

- List containers with their information and status:

```
colima list
```

- Show runtime status:

```
colima status
```

:

Returns a successful exit status code of 0.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-_003a.

- Return a successful exit code:

```
:
```

- Make a command always exit with 0:

```
{{command}} || :
```

colordiff

A wrapper around **`diff`** that produces the same output but with pretty syntax highlighting.

Color schemes can be customized.

More information: https://github.com/kimmel/colordiff.

- Compare files:

```
colordiff {{file1}} {{file2}}
```

- Output in two columns:

```
colordiff -y {{file1}} {{file2}}
```

- Ignore case differences in file contents:

```
colordiff -i {{file1}} {{file2}}
```

- Report when two files are the same:

```
colordiff -s {{file1}} {{file2}}
```

- Ignore whitespace:

```
colordiff -w {{file1}} {{file2}}
```

colorls

Beautify **`ls`** command output, with color and font-awesome icons. Available as a Ruby gem.

More information: https://github.com/athityakumar/colorls.

- List files one per line:

  ```
  colorls -1
  ```

- List all files, including hidden files:

  ```
  colorls --all
  ```

- Long format list (permissions, ownership, size, and modification date) of all files:

  ```
  colorls --long --all
  ```

- Only list directories:

  ```
  colorls --dirs
  ```

colorpicker

A minimalist X11 colorpicker.

Any mouse gesture except **`<LeftClick>`** will exit the program.

More information: https://github.com/ym1234/colorpicker.

- Launch colorpicker and print the hexadecimal and RGB value of each clicked pixel to `stdout`:

  ```
  colorpicker
  ```

- Only print the color of one clicked pixel and then exit:

  ```
  colorpicker --one-shot
  ```

- Print the color of each clicked pixel and quit when a key is pressed:

  ```
  colorpicker --quit-on-keypress
  ```

- Only print the RGB value:

  ```
  colorpicker --rgb
  ```

- Only print the hexadecimal value:

  ```
  colorpicker --hex
  ```

combine

Perform set operations on lines of two files.

The order of the output lines is determined by the order of the lines in the first file.

See also: **`diff`**.

More information: https://manned.org/combine.

- Output lines that are in both specified files:

```
combine {{path/to/file1}} and {{path/to/file2}}
```

- Output lines that are in the first but not in the second file:

```
combine {{path/to/file1}} not {{path/to/file2}}
```

- Output lines that in are in either of the specified files:

```
combine {{path/to/file1}} or {{path/to/file2}}
```

- Output lines that are in exactly one of the specified files:

```
combine {{path/to/file1}} xor {{path/to/file2}}
```

comby

Tool for structural code search and replace that supports many languages.

More information: https://github.com/comby-tools/comby.

- Match and rewrite templates, and print changes:

```
comby '{{assert_eq!(:[a], :[b])}}' '{{assert_eq!(:[b], :[a])}}' {{.rs}}
```

- Match and rewrite with rewrite properties:

```
comby '{{assert_eq!(:[a], :[b])}}' '{{assert_eq!(:[b].Capitalize, :[a])}}' {{.rs}}
```

- Match and rewrite in-place:

```
comby -in-place '{{match_pattern}}' '{{rewrite_pattern}}'
```

- Only perform matching and print matches:

```
comby -match-only '{{match_pattern}}' ""
```

comm

Select or reject lines common to two files. Both files must be sorted.

More information: https://www.gnu.org/software/coreutils/manual/html_node/comm-invocation.html.

- Produce three tab-separated columns: lines only in first file, lines only in second file and common lines:

```
comm {{file1}} {{file2}}
```

- Print only lines common to both files:

```
comm -12 {{file1}} {{file2}}
```

- Print only lines common to both files, reading one file from `stdin`:

```
cat {{file1}} | comm -12 - {{file2}}
```

- Get lines only found in first file, saving the result to a third file:

```
comm -23 {{file1}} {{file2}} > {{file1_only}}
```

- Print lines only found in second file, when the files aren't sorted:

```
comm -13 <(sort {{file1}}) <(sort {{file2}})
```

comma

This command is an alias of `,`.

- View documentation for the original command:

```
tldr ,
```

command

Command forces the shell to execute the program and ignore any functions, builtins and aliases with the same name.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-command.

- Execute the `ls` program literally, even if an `ls` alias exists:

```
command {{ls}}
```

- Find and execute a command using a default `$PATH` (`/bin:/usr/bin:/sbin:/usr/sbin:/etc:/usr/etc`) that guarantees to find all standard utilities:

```
command -p {{command_name}}
```

- Display the path to the executable or the alias definition of a specific command:

```
command -v {{command_name}}
```

compare

This command is an alias of **`magick compare`**.

- View documentation for the original command:

```
tldr magick compare
```

compgen

Bash built-in command for generating possible completion matches in Bash.

Usually used within custom completion functions.

See also: **`complete`**, **`compopt`**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-compgen.

- List all shell built-ins, aliases, functions and executables that you could run:

```
compgen -c
```

- List all commands that you could run that start with a specified string and save results to `COMPREPLY`:

```
compgen -V COMPREPLY -c {{str}}
```

- Match against a wordlist:

```
compgen -W "{{apple orange banana}}" {{a}}
```

- List all aliases:

```
compgen -a
```

- List all functions that you could run:

```
compgen -A function
```

- Show shell reserved keywords:

```
compgen -k
```

- See all available commands/aliases starting with `ls`:

```
compgen -ac {{ls}}
```

- List all users on the system:

```
compgen -u
```

complete

Get and set argument autocompletion rules of shell commands in Bash.

The specified completions will be invoked when **<Tab>** is pressed in Bash.

See also: **compgen**, **compopt**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-complete.

- Set arguments of a command to autocomplete through a function (completion response is sent in `COMPREPLY` variable):

```
complete -F {{function}} {{command}}
```

- Set arguments of a command to autocomplete through another command (`$1` is the command, `$2` is the argument the cursor is on, and `$3` is the argument preceding the cursor):

```
complete -C {{autocomplete_command}} {{command}}
```

- Set arguments of a command to autocomplete to shell builtins:

```
complete -b {{command}}
```

- Apply autocompletion without appending a space to the completed word:

```
complete -o nospace -F {{function}} {{command}}
```

- List all loaded complete specifications:

```
complete -p
```

- List loaded complete specifications for a command:

```
complete -p {{command}}
```

compopt

Print or change the completion options for a command.

`-o` means enabled and **`+o`** means disabled.

See also: **`compgen`**, **`complete`**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-compopt.

- Print the completion options for given command:

```
compopt {{command}}
```

- Enable or disable a completion option of a command:

```
compopt {{-o|+o}} {{option1}} {{-o|+o}} {{option2}} {{command}}
```

- Print the options for the currently executing completion:

```
compopt
```

- Enable or disable a completion option of a command:

```
compopt {{-o|+o}} {{option1}} {{-o|+o}} {{option2}}
```

composer audit

Analyze a PHP project's dependencies to detect known security vulnerabilities and list affected packages.

See also: **`composer`**.

More information: https://getcomposer.org/doc/03-cli.md#audit.

- Check for security vulnerabilities in your current project:

```
composer audit
```

- Omit dev dependencies in the audit:

```
composer audit --no-dev
```

- Filter vulnerabilities by output format:

```
composer audit --format {{table|plain|json|summary}}
```

- Output audit results to a file in JSON format:

```
composer audit --format json > audit_report.json
```

- Verify whether a specific package in your project is affected by security issues:

```
composer audit {{vendor}}/{{package}}
```

composer-require-checker

Analyze Composer dependencies for soft dependencies.

More information: https://github.com/maglnet/ComposerRequireChecker.

- Analyze a Composer JSON file:

```
composer-require-checker check {{path/to/composer.json}}
```

- Analyze a Composer JSON file with a specific configuration:

```
composer-require-checker check --config-file {{path/to/
config.json}} {{path/to/composer.json}}
```

composer

A package-based dependency manager for PHP projects.

More information: https://getcomposer.org/doc/03-cli.md.

- Interactively create a `composer.json` file:

  ```
  composer init
  ```

- Add a package as a dependency for this project, adding an entry to `composer.json`:

  ```
  composer require {{user/package}}
  ```

- Install all the dependencies in this project's `composer.json` and create `composer.lock`:

  ```
  composer install
  ```

- Uninstall a package from this project, removing it as a dependency from `composer.json` and `composer.lock`:

  ```
  composer remove {{user/package}}
  ```

- Update all the dependencies in this project's `composer.json` and note new versions in `composer.lock` file:

  ```
  composer update
  ```

- Update only `composer.lock` after updating `composer.json` manually:

  ```
  composer update --lock
  ```

- Learn more about why a dependency can't be installed:

  ```
  composer why-not {{user/package}}
  ```

- Update composer to its latest version:

  ```
  composer self-update
  ```

conan

The open source, decentralized and cross-platform package manager to create and share all your native binaries.

Some subcommands such as **`frogarian`** have their own usage documentation.

More information: https://docs.conan.io/2/reference/commands.html.

- Install packages based on `conanfile.txt`:

  ```
  conan install {{.}}
  ```

- Install packages and create configuration files for a specific generator:

  ```
  conan install -g {{generator}}
  ```

- Install packages, building from source:

  ```
  conan install {{.}} --build
  ```

- Search for locally installed packages:

  ```
  conan search {{package}}
  ```

- Search for remote packages:

  ```
  conan search {{package}} -r {{remote}}
  ```

- List remotes:

  ```
  conan remote list
  ```

conda activate

Activate a conda environment.

See also: **`conda deactivate`** to deactivate a conda environment.

More information: https://docs.conda.io/projects/conda/en/stable/dev-guide/deep-dives/activation.html.

- Activate an existing environment named `myenv`:

  ```
  conda activate myenv
  ```

- Activate an existing environment located at custom path:

  ```
  conda activate {{path/to/myenv}}
  ```

- Stack `myenv` environment on top of a previous environment making libraries/commands/variables from both accessible:

  ```
  conda activate --stack myenv
  ```

- Start a clean environment `myenv` without stacking it making previous environment libraries/commands/variables not accessible:

  ```
  conda activate --no-stack myenv
  ```

- Display help:

  ```
  conda activate {{[-h|--help]}}
  ```

conda clean

Delete temporary or unused files: index cache, lock files, unused cache packages, tarballs, and log files.

More information: https://docs.conda.io/projects/conda/en/stable/commands/clean.html.

- Delete all temporary or unused files verbosely and say yes to all confirmations:

```
conda clean {{[-avy|--all --verbose --yes]}}
```

- Delete only index cache, tarballs, and log files:

```
conda clean {{[-itl|--index-cache --tarballs --logfiles]}}
```

- Delete only temporary [c]ache files that could not be deleted earlier due to being in use:

```
conda clean {{[-c|--tempfiles]}} {{path/to/tempfiles}}
```

- Delete only unused packages. Might delete packages installed with softlinks:

```
conda clean {{[-p|--packages]}}
```

- Force delete all writable packages. More broad than the `--all` option. Will delete packages installed with softlinks:

```
conda clean {{[-f|--force-pkgs-dirs]}}
```

- Display help:

```
conda clean {{[-h|--help]}}
```

conda compare

Compare packages between conda environments.

More information: https://docs.conda.io/projects/conda/en/stable/commands/compare.html.

- Compare packages in the current directory to packages from the file `file.yml`:

```
conda compare file.yml
```

- Compare packages in environment named `myenv` to packages from the file `file.yml`:

```
conda compare {{[-n|--name]}} myenv {{path/to/file.yml}}
```

- Compare packages in environment `myenv` at custom path (i.e. prefix) to packages from the file `file.yml`:

```
conda compare {{[-p|--prefix]}} {{path/to/myenv}} {{path/to/file.yml}}
```

- Display help:

```
conda create {{[-h|--help]}}
```

conda config

Modify configuration values in `.condarc`.

More information: https://docs.conda.io/projects/conda/en/latest/commands/config.html.

- Show all configuration values:

```
conda config --show
```

- Show current value of a configuration option:

```
conda config --show {{config_option}}
```

- Set a configuration value:

```
conda config --set {{key}} {{value}}
```

- Remove a configuration value:

```
conda config --remove {{key}} {{value}}
```

- Append a value to an existing configuration key list:

```
conda config --append {{key}} {{value}}
```

- Prepend a value to an existing configuration key list:

```
conda config --prepend {{key}} {{value}}
```

- Describe given configuration option:

```
conda config --describe {{config_option}}
```

conda create

Create new conda environments.

More information: https://docs.conda.io/projects/conda/en/latest/commands/create.html.

- Create a new environment named `py39`, install Python 3.9, NumPy v1.11 or above in it, and the latest stable version of SciPy. Say yes to all confirmations:

```
conda create {{[-ny|--name --yes]}} {{py39}} python={{3.9}} "{{numpy>=1.11 scipy}}"
```

- Create a new environment named `myenv` and install packages listed in files:

```
conda create {{[-n|--name]}} myenv --file {{file1.yml}} --file {{file2.yml}}
```

- Create a new environment named `myenv` at custom path (i.e. prefix):

```
conda create {{[-p|--prefix]}} {{path/to/myenv}}
```

- Make exact copy of an environment named `py39`:

```
conda create --clone py39 {{[-n|--name]}} {{py39-copy}}
```

- Display help:

```
conda create {{[-h|--help]}}
```

conda deactivate

Deactivate a conda environment.

More information: https://docs.conda.io/projects/conda/en/latest/commands/deactivate.html.

- Deactivate a conda environment:

```
conda deactivate
```

conda doctor

Display a health report for your environment.

More information: https://docs.conda.io/projects/conda/en/latest/commands/doctor.html.

- View report for the currently active environment:

```
conda doctor
```

- Specify an environment by name:

```
conda doctor {{[-n|--name]}} {{environment_name}}
```

- Specify an environment by its path:

```
conda doctor {{[-p|--prefix]}} {{path/to/environment}}
```

- Enable verbose output (Note: the `-v` flag can be repeated to increase verbosity):

```
conda doctor {{[-v|--verbose]}}
```

conda env

Manage conda environments.

More information: https://docs.conda.io/projects/conda/en/latest/commands/env/index.html.

- Create an environment from an environment file (YAML, TXT, etc.):

```
conda env create {{[-f|--file]}} {{path/to/file}}
```

- Delete an environment and everything in it:

```
conda env remove {{[-n|--name]}} {{environment_name}}
```

- Update an environment based on an environment file:

```
conda env update {{[-f|--file]}} {{path/to/file}} --prune
```

- List all environments:

```
conda env list
```

- View environment details:

```
conda env export
```

- List environment variables:

```
conda env config vars list
```

- Set environment variables:

```
conda env config vars set {{my_var}}={{value}}
```

conda export

Export environment details.

More information: https://docs.conda.io/projects/conda/en/latest/commands/export.html.

- Export the current environment details to `stdout`:

```
conda export
```

- Export the current environment details to a `YAML` file:

```
conda export {{[-f|--file]}} {{path/to/environment.yaml}}
```

- Export details in a specific format:

```
conda export --format {{environment-json|environment-yaml|explicit|json|reqs|requirements|txt|yaml|yml}}
```

- Target an environment by name:

```
conda export {{[-n|--name]}} {{environment_name}}
```

- Target an environment by its path:

```
conda export {{[-p|--prefix]}} {{path/to/environment}}
```

- Include a specific channel:

```
conda export {{[-c|--channel]}} {{channel_name}}
```

conda info

Display details about the conda installation.

More information: https://docs.conda.io/projects/conda/en/latest/commands/info.html.

- Show all information:

```
conda info {{[-a|--all]}}
```

- Display base environment path:

```
conda info --base
```

- List all conda environments:

```
conda info {{[-e|--envs]}}
```

- List environment variables:

```
conda info {{[-s|--system]}}
```

- Display list of channels with tokens exposed:

```
conda info --unsafe-channels
```

conda init

Initialize conda for shell interaction.

Most shells need to be closed and restarted for changes to take effect.

More information: https://docs.conda.io/projects/conda/en/latest/commands/init.html.

- Initialize a specific shell (if none is specified, defaults to `bash` for UNIX and `powershell` for Windows):

  ```
  conda init {{zsh|bash|powershell|fish|tcsh|xonsh}}
  ```

- Initialize all available shells:

  ```
  conda init --all
  ```

- Initialize conda for all users on the system:

  ```
  conda init --system
  ```

- Don't initialize conda for the current user:

  ```
  conda init --no-user
  ```

- Add `condabin/` directory to PATH:

  ```
  conda init --condabin
  ```

- Undo effects of the last `conda init`:

  ```
  conda init --reverse
  ```

conda install

Install packages into an existing conda environment.

More information: https://docs.conda.io/projects/conda/en/latest/commands/install.html.

- Install one or more package into the currently active conda environment:

```
conda install {{package1 package2 ...}}
```

- Install a single package into the currently active conda environment using channel conda-forge:

```
conda install {{[-c|--channel]}} conda-forge {{package}}
```

- Install a single package into the currently active conda environment using channel conda-forge and ignoring other channels:

```
conda install {{[-c|--channel]}} conda-forge --override-channels {{package}}
```

- Install a specific version of a package:

```
conda install {{package}}={{version}}
```

- Install a package into a specific environment:

```
conda install {{[-n|--name]}} {{environment}} {{package}}
```

- Update a package in the current environment:

```
conda install --upgrade {{package}}
```

- Install a package and agree to the transactions without prompting:

```
conda install {{[-y|--yes]}} {{package}}
```

conda list

List installed packages in a conda environment.

More information: https://docs.conda.io/projects/conda/en/stable/commands/list.html.

- List all packages in the current environment:

```
conda list
```

- List packages in the named environment:

```
conda list {{[-n|--name]}} {{environment}}
```

- List packages installed in a given path:

```
conda list {{[-p|--prefix]}} {{path/to/environment}}
```

- Filter installed packages using `regex`:

```
conda list {{regex}}
```

- Save packages for future use:

```
conda list {{[-e|--export]}} > {{path/to/package-list.txt}}
```

conda notices

Retrieve latest channel notifications.

More information: https://docs.conda.io/projects/conda/en/latest/commands/notices.html.

- Show notices for the default channel and all `.condarc` channels:

```
conda notices
```

- Include a specific channel:

```
conda notices {{[-c|--channel]}} {{channel_name}}
```

- Ignore default and `.condarc` channels:

```
conda notices {{[-c|--channel]}} {{channel_name}} --override-channels
```

conda remove

Remove packages from a conda environment.

More information: https://docs.conda.io/projects/conda/en/latest/commands/remove.html.

- Remove `scipy` from the currently active environment:

```
conda remove scipy
```

- Remove a list of packages from the specified environment:

```
conda remove {{[-n|--name]}} {{environment_name}} {{package1 package2 ...}}
```

- Remove all packages and the environment itself:

```
conda remove {{[-n|--name]}} {{environment_name}} --all
```

- Remove all packages, but keep the environment:

```
conda remove {{[-n|--name]}} {{environment_name}} --all --keep-env
```

conda rename

Rename an existing conda environment.

The base environment and the currently-active environment cannot be renamed.

More information: https://docs.conda.io/projects/conda/en/latest/commands/rename.html.

- Rename an environment via its name:

```
conda rename {{[-n|--name]}} {{current_name}} {{new_name}}
```

- Rename an environment via its full path (i.e. prefix):

```
conda rename {{[-p|--prefix]}} {{path/to/env}} {{new_name}}
```

conda repoquery

Advanced search for packages in a conda repository.

More information: https://docs.conda.io/projects/conda/en/latest/commands/repoquery/index.html.

- Show available versions of the specified package:

```
conda repoquery search {{package}}
```

- Show dependencies of the specified package:

```
conda repoquery depends {{package}}
```

- Show packages that depend on the specified package:

```
conda repoquery whoneeds {{package}}
```

conda run

Run an executable command in a conda environment.

More information: https://docs.conda.io/projects/conda/en/latest/commands/run.html.

- Run a command in the currently active environment:

```
conda run {{command}}
```

- Target an environment by name:

```
conda run {{[-n|--name]}} {{environment_name}} {{command}}
```

- Target an environment by its path (i.e. prefix):

```
conda run {{[-p|--path]}} {{path/to/env}} {{command}}
```

conda search

Search for packages and show their details.

More information: https://docs.conda.io/projects/conda/en/latest/commands/search.html.

- Search for a specific package:

```
conda search {{package_name}}
```

- Search for a package along with its details:

```
conda search {{package_name}} {{[-i|--info]}}
```

- Search for packages containing `string` in the package name:

```
conda search "*string*"
```

- Search for specific version of the package:

```
conda search "{{package_name}}>={{package_version}}"
```

- Search a package within a specific channel:

```
conda search {{channel}}::{{package_name}}
```

- Search if package is installed in any local environments:

```
conda search --envs {{package_name}}
```

conda update

Update packages within a conda environment, including conda itself.

More information: https://docs.conda.io/projects/conda/en/latest/commands/update.html.

- Update all packages in the current environment:

```
conda update {{[--all|--update-all]}}
```

- Update a specific package in the current environment:

```
conda update {{package_name}}
```

- Update conda itself in the base environment:

```
conda update {{[-n|--name]}} base conda
```

- Update packages while ignoring pinned packages:

```
conda update --no-pin
```

- Update packages in offline mode:

```
conda update --offline
```

conda

Package, dependency and environment management for any programming language.

Some subcommands such as **`create`** have their own usage documentation.

More information: https://docs.conda.io/projects/conda/en/latest/commands/index.html.

- Create a new environment, installing named packages into it:

```
conda create {{[-n|--name]}} {{environment_name}} {{python=3.9 matplotlib}}
```

- List all environments:

```
conda info {{[-e|--envs]}}
```

- Activate an environment:

```
conda activate {{environment_name}}
```

- Deactivate an environment:

```
conda deactivate
```

- Delete an environment (remove all packages):

```
conda remove {{[-n|--name]}} {{environment_name}} --all
```

- Install packages into the current environment:

```
conda install {{python=3.4 numpy}}
```

- List currently installed packages in current environment:

```
conda list
```

- Delete unused packages and caches:

```
conda clean {{[-a|--all]}}
```

confettysh

Display animated confetti and fireworks over SSH.

See also: **`ssh`**.

More information: https://github.com/charmbracelet/confettysh.

- Start a local `confettysh` server:

  `confettysh`

- Run the server on a custom port:

  `confettysh {{[-p|--port]}} {{2323}}`

- Connect to the local server and show fireworks:

  `ssh {{[-p|--port]}} {{2222}} localhost -t fireworks`

consul kv

Distributed key-value store with health checking and service discovery.

More information: https://learn.hashicorp.com/consul/getting-started/kv.

- Read a value from the key-value store:

  ```
  consul kv get {{key}}
  ```

- Store a new key-value pair:

  ```
  consul kv put {{key}} {{value}}
  ```

- Delete a key-value pair:

  ```
  consul kv delete {{key}}
  ```

consul

Distributed key-value store with health checking and service discovery.

Some subcommands such as **kv** have their own usage documentation.

More information: https://www.consul.io/commands.

- Display help:

```
consul --help
```

- Display help for a subcommand:

```
consul {{subcommand}} --help
```

- Display version:

```
consul --version
```

continue

Skip to the next iteration of a **`for`**, **`while`**, **`until`** or **`select`** loop.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-continue.

- Skip to the next iteration:

```
while :; do continue; {{echo "This will never be reached"}};
done
```

- Skip to the next iteration from within a nested loop:

```
for i in {{{1..3}}}; do {{echo $i}}; while :; do continue 2;
done; done
```

convert

This command is an alias of **`magick convert`**.

Note: This alias is deprecated since ImageMagick 7. It has been replaced by **`magick`**.

Use **`magick convert`** if you need to use the old tool in versions 7+.

- View documentation for the original command:

```
tldr magick convert
```

convmv

Convert filenames (NOT file content) from one encoding to another.

More information: https://www.j3e.de/linux/convmv/man/.

- Test filename encoding conversion (don't actually change the filename):

```
convmv -f {{from_encoding}} -t {{to_encoding}} {{input_file}}
```

- Convert filename encoding and rename the file to the new encoding:

```
convmv -f {{from_encoding}} -t {{to_encoding}} --notest
{{input_file}}
```

copr-cli

Interface with Fedora-Projects copr instance for building RPMs and publishing them.

More information: https://manned.org/copr-cli.

- Show user logged in to copr:

```
copr-cli whoami
```

- Build a local spec file on copr:

```
copr-cli build {{repository}} {{path/to/spec_file}}
```

- Check status of builds:

```
copr-cli list-builds {{repository}}
```

- Trigger a copr build of a spec-file from public (Git) repository:

```
copr-cli buildscm {{repository}} --clone-url {{https://
git.example.org/repo}} --spec {{spec_file_name}}
```

copr

This command is an alias of **`copr-cli`**.

- View documentation for the original command:

  ```
  tldr copr-cli
  ```

coproc

Bash builtin for creating interactive asynchronous subshells.

More information: https://www.gnu.org/software/bash/manual/bash.html#Coprocesses.

- Run a subshell asynchronously:

```
coproc { {{command1; command2; ...}}; }
```

- Create a coprocess with a specific name:

```
coproc {{name}} { {{command1; command2; ...}}; }
```

- Write to a specific coprocess `stdin`:

```
echo "{{input}}" >&"${{{name[1]}}}"
```

- Read from a specific coprocess `stdout`:

```
read <&"${{{name[0]}}}" {{variable}}
```

- Create a coprocess which repeatedly reads `stdin` and runs some commands on the input:

```
coproc {{name}} { while read {{line}}; do {{command1; command2; ...}}; done }
```

- Create a coprocess which repeatedly reads `stdin`, runs a pipeline on the input, and writes the output to `stdout`:

```
coproc {{name}} { while read {{line}}; do {{echo "$line"}} | {{command1 | command2 | ...}} | cat /dev/fd/0; done }
```

- Create and use a coprocess running `bc`:

```
coproc BC { bc --mathlib; }; echo "1/3" >&"${BC[1]}"; read <&"${BC[0]}" output; echo "$output"
```

copyq

Clipboard manager with advanced features.

More information: https://copyq.readthedocs.io/en/latest/command-line.html.

- Launch CopyQ to store clipboard history:

```
copyq
```

- Show current clipboard content:

```
copyq clipboard
```

- Insert raw text into the clipboard history:

```
copyq add -- {{text1}} {{text2}} {{text3}}
```

- Insert text containing escape sequences ('\n', '\t') into the clipboard history:

```
copyq add {{firstline\nsecondline}}
```

- Print the content of the first 3 items in the clipboard history:

```
copyq read 0 1 2
```

- Copy a file's contents into the clipboard:

```
copyq < {{path/to/file.txt}} copy
```

- Copy a JPEG image into the clipboard:

```
copyq < {{path/to/image.jpg}} copy image/jpeg
```

cordova

Mobile apps with HTML, CSS & JS.

More information: https://cordova.apache.org/docs/en/latest/guide/cli/.

- Create a Cordova project:

```
cordova create {{path/to/directory}} {{package}} {{project_name}}
```

- Display the current workspace status:

```
cordova info
```

- Add a Cordova platform:

```
cordova platform add {{platform}}
```

- Remove a Cordova platform:

```
cordova platform remove {{platform}}
```

- Add a Cordova plugin:

```
cordova plugin add {{pluginid}}
```

- Remove a Cordova plugin:

```
cordova plugin remove {{pluginid}}
```

core-validate-commit

Validate commit messages for Node.js core.

More information: https://github.com/nodejs/core-validate-commit.

- Validate the current commit:

```
core-validate-commit
```

- Validate a specific commit:

```
core-validate-commit {{commit_hash}}
```

- Validate a range of commits:

```
git rev-list {{commit_hash}}..HEAD | xargs core-validate-commit
```

- List all validation rules:

```
core-validate-commit {{[-l|--list]}}
```

- List all valid Node.js subsystems:

```
core-validate-commit {{[-ls|--list-subsystem]}}
```

- Validate the current commit formatting the output in tap format:

```
core-validate-commit {{[-t|--tap]}}
```

- Display help:

```
core-validate-commit {{[-h|--help]}}
```

corepack

Zero-runtime-dependency package acting as bridge between Node projects and their package managers.

More information: https://github.com/nodejs/corepack.

- Add the Corepack shims to the Node.js installation directory to make them available as global commands:

```
corepack enable
```

- Add the Corepack shims to a specific directory:

```
corepack enable --install-directory {{path/to/directory}}
```

- Remove the Corepack shims from the Node.js installation directory:

```
corepack disable
```

- Prepare a specific package manager:

```
corepack prepare {{package_manager}}@{{version}} --activate
```

- Prepare the package manager configured for the project in the current path:

```
corepack prepare
```

- Use a package manager without installing it as a global command:

```
corepack {{npm|pnpm|yarn}} {{package_manager_arguments}}
```

- Install a package manager from the specified archive:

```
corepack hydrate {{path/to/corepack.tgz}}
```

- Display help for a subcommand:

```
corepack {{subcommand}} --help
```

coreutils

Uutils coreutils is a cross-platform reimplementation of the GNU coreutils in Rust Language.

Uutils includes a multi-call binary from which the utils can be invoked. This reduces the binary size of the binary and can be useful for portability.

More information: https://uutils.github.io/coreutils/docs/multicall.html.

- Run a utility with arguments:

```
coreutils {{util}} {{util_options}}
```

- List files in [l]ong format:

```
coreutils ls -l
```

- Display help for `ls`:

```
coreutils ls --help
```

cosign

Container Signing, Verification and Storage in an OCI registry.

More information: https://github.com/sigstore/cosign.

- Generate a key-pair:

```
cosign generate-key-pair
```

- Sign a container and store the signature in the registry:

```
cosign sign -key {{cosign.key}} {{image}}
```

- Sign a container image with a key pair stored in a Kubernetes secret:

```
cosign sign -key k8s://{{namespace}}/{{key}} {{image}}
```

- Sign a blob with a local key pair file:

```
cosign sign-blob --key {{cosign.key}} {{path/to/file}}
```

- Verify a container against a public key:

```
cosign verify -key {{cosign.pub}} {{image}}
```

- Verify images with a public key in a Dockerfile:

```
cosign dockerfile verify -key {{cosign.pub}} {{path/to/
Dockerfile}}
```

- Verify an image with a public key stored in a Kubernetes secret:

```
cosign verify -key k8s://{{namespace}}/{{key}} {{image}}
```

- Copy a container image and its signatures:

```
cosign copy {{example.com/src:latest}} {{example.com/
dest:latest}}
```

cotton

Markdown test specification runner.

More information: https://github.com/chonla/cotton.

- Use a specific base URL:

```
cotton -u {{base_url}} {{path/to/file.md}}
```

- Disable certificate verification (insecure mode):

```
cotton -u {{base_url}} -i {{path/to/file.md}}
```

- Stop running when a test fails:

```
cotton -u {{base_url}} -s {{path/to/file.md}}
```

couchdb

Command-line interface for Apache CouchDB database server.

More information: https://couchdb.apache.org.

- Start CouchDB:

```
couchdb
```

- Start CouchDB interactive shell:

```
couchdb -i
```

- Start CouchDB as a background process:

```
couchdb -b
```

- Kill the background process (Note: It will respawn if needed):

```
couchdb -k
```

- Shutdown the background process:

```
couchdb -d
```

cowsay

Print ASCII art (by default a cow) saying or thinking something.

More information: https://manned.org/cowsay.

- Print an ASCII cow saying "hello, world":

```
cowsay "{{hello, world}}"
```

- Print an ASCII cow saying text from `stdin`:

```
echo "{{hello, world}}" | cowsay
```

- List all available art types:

```
cowsay -l
```

- Print the specified ASCII art saying "hello, world":

```
cowsay -f {{art}} "{{hello, world}}"
```

- Print a dead thinking ASCII cow:

```
cowthink -d "{{I'm just a cow, not a great thinker...}}"
```

- Print an ASCII cow with custom eyes saying "hello, world":

```
cowsay -e {{characters}} "{{hello, world}}"
```

cp

Copy files and directories.

More information: https://www.gnu.org/software/coreutils/manual/html_node/cp-invocation.html.

- Copy a file to another location:

```
cp {{path/to/source_file.ext}} {{path/to/target_file.ext}}
```

- Copy a file into another directory, keeping the filename:

```
cp {{path/to/source_file.ext}} {{path/to/
target_parent_directory}}
```

- Recursively copy a directory's contents to another location (if the destination exists, the directory is copied inside it):

```
cp {{[-r|--recursive]}} {{path/to/source_directory}} {{path/
to/target_directory}}
```

- Copy a directory recursively, in verbose mode (shows files as they are copied):

```
cp {{[-vr|--verbose --recursive]}} {{path/to/
source_directory}} {{path/to/target_directory}}
```

- Copy multiple files at once to a directory:

```
cp {{[-t|--target-directory]}} {{path/to/
destination_directory}} {{path/to/file1 path/to/file2 ...}}
```

- Copy all files with a specific extension to another location, in interactive mode (prompts user before overwriting):

```
cp {{[-i|--interactive]}} {{*.ext}} {{path/to/
target_directory}}
```

- Follow symbolic links before copying:

```
cp {{[-L|--dereference]}} {{link}} {{path/to/
target_directory}}
```

- Use the full path of source files, creating any missing intermediate directories when copying:

```
cp --parents {{source/path/to/file}} {{path/to/target_file}}
```

cpan

Query, download and build perl modules from CPAN sites.

More information: https://manned.org/cpan.

- Install a module (`-i` is optional):

```
cpan {{-i}} {{module_name}}
```

- Force install a module (`-i` is not optional):

```
cpan -fi {{module_name}}
```

- Upgrade all installed modules:

```
cpan -u
```

- Recompile modules:

```
cpan -r
```

cpdf

Manipulate PDF files.

More information: https://www.coherentpdf.com/cpdfmanual/cpdfmanual.html.

- Select pages 1, 2, 3 and 6 from a source document and write those to a destination document:

```
cpdf {{path/to/source_document.pdf}} {{1-3,6}} -o {{path/to/destination_document.pdf}}
```

- Merge two documents into a new one:

```
cpdf -merge {{path/to/source_document_one.pdf}} {{path/to/source_document_two.pdf}} -o {{path/to/destination_document.pdf}}
```

- Show the bookmarks of a document:

```
cpdf -list-bookmarks {{path/to/document.pdf}}
```

- Split a document into ten-page chunks, writing them to `chunk001.pdf`, `chunk002.pdf`, etc:

```
cpdf -split {{path/to/document.pdf}} -o {{path/to/chunk%%%.pdf}} -chunk {{10}}
```

- Encrypt a document using 128bit encryption, providing `fred` as owner password and `joe` as user password:

```
cpdf -encrypt {{128bit}} {{fred}} {{joe}} {{path/to/source_document.pdf}} -o {{path/to/encrypted_document.pdf}}
```

- Decrypt a document using the owner password `fred`:

```
cpdf -decrypt {{path/to/encrypted_document.pdf}} owner={{fred}} -o {{path/to/decrypted_document.pdf}}
```

- Show the annotations of a document:

```
cpdf -list-annotations {{path/to/document.pdf}}
```

- Create a new document from an existing one with additional metadata:

```
cpdf -set-metadata {{path/to/metadata.xml}} {{path/to/source_document.pdf}} -o {{path/to/destination_document.pdf}}
```

cpio

Copy files in and out of archives.

Supports the following archive formats: cpio's custom binary, old ASCII, new ASCII, crc, HPUX binary, HPUX old ASCII, old tar, and POSIX.1 tar.

More information: https://www.gnu.org/software/cpio/manual/cpio.html#Invoking-cpio.

- Take a list of file names from `stdin` and add them onto an archive (copy-[o]ut) in cpio's binary forma:

```
echo "{{path/to/file1 path/to/file2 ...}}" | cpio {{[-o|--create]}} > {{archive.cpio}}
```

- Copy all files and directories in a directory and add them onto an archive (copy-[o]ut), in verbose mode:

```
find {{path/to/directory}} | cpio {{[-ov|--create --verbose]}} > {{archive.cpio}}
```

- Pick all files from an archive (copy-[i]n), generating directories where needed, in verbose mode:

```
cpio < {{archive.cpio}} {{[-idv|--extract --make-directories --verbose]}}
```

cppcheck

A static analysis tool for C/C++ code.

Instead of syntax errors, it focuses on the types of bugs that compilers normally do not detect.

More information: https://cppcheck.sourceforge.net.

- Recursively check the current directory, showing progress on the screen and logging error messages to a file:

```
cppcheck . 2> cppcheck.log
```

- Recursively check a given directory, and don't print progress messages:

```
cppcheck {{[-q|--quiet]}} {{path/to/directory}}
```

- Check a given file, specifying which tests to perform (by default only errors are shown):

```
cppcheck --enable {{error|warning|style|performance|
portability|information|all}} {{path/to/file.cpp}}
```

- List available tests:

```
cppcheck --errorlist
```

- Check a given file, ignoring specific tests:

```
cppcheck --suppress {{test_id1}} --suppress {{test_id2}}
{{path/to/file.cpp}}
```

- Check the current directory, providing paths for include files located outside it (e.g. external libraries):

```
cppcheck -I {{include/directory_1}} -I {{include/
directory_2}} .
```

- Check a Microsoft Visual Studio project (`*.vcxproj`) or solution (`*.sln`):

```
cppcheck --project {{path/to/project.sln}}
```

cppclean

Find unused code in C++ projects.

More information: https://github.com/myint/cppclean.

- Run in a project's directory:

```
cppclean {{path/to/project}}
```

- Run on a project where the headers are in the `inc1/` and `inc2/` directories:

```
cppclean {{path/to/project}} --include-path {{inc1}} --include-path {{inc2}}
```

- Run on a specific file `main.cpp`:

```
cppclean {{main.cpp}}
```

- Run on the current directory, excluding the "build" directory:

```
cppclean {{.}} --exclude {{build}}
```

crackle

Crack and decrypt Bluetooth Low Energy (BLE) encryption.

More information: https://github.com/mikeryan/crackle.

- Check whether the recorded BLE communications contain the packets necessary for recovering temporary keys (TKs):

```
crackle -i {{path/to/input.pcap}}
```

- Use brute force to recover the TK of the recorded pairing events and use it to decrypt all subsequent communications:

```
crackle -i {{path/to/input.pcap}} -o {{path/to/
decrypted.pcap}}
```

- Use the specified long-term key (LTK) to decrypt the recorded communication:

```
crackle -i {{path/to/input.pcap}} -o {{path/to/
decrypted.pcap}} -l {{81b06facd90fe7a6e9bbd9cee59736a7}}
```

cradle deploy

Manage Cradle deployments.

More information: https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#deploy.

- Deploy Cradle to a server:

```
cradle deploy production
```

- Deploy static assets to Amazon S3:

```
cradle deploy s3
```

- Deploy static assets including the Yarn "components" directory:

```
cradle deploy s3 --include-yarn
```

- Deploy static assets including the "upload" directory:

```
cradle deploy s3 --include-upload
```

cradle elastic

Manage the Elasticsearch instances for a Cradle instance.

More information: https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#elastic.

- Truncate the Elasticsearch index:

  ```
  cradle elastic flush
  ```

- Truncate the Elasticsearch index for a specific package:

  ```
  cradle elastic flush {{package}}
  ```

- Submit the Elasticsearch schema:

  ```
  cradle elastic map
  ```

- Submit the Elasticsearch schema for a specific package:

  ```
  cradle elastic map {{package}}
  ```

- Populate the Elasticsearch indices for all packages:

  ```
  cradle elastic populate
  ```

- Populate the Elasticsearch indices for a specific package:

  ```
  cradle elastic populate {{package}}
  ```

cradle install

Install the Cradle PHP framework components.

More information: https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#install.

- Install Cradle's components (User will be prompted for further details):

```
cradle install
```

- Forcefully overwrite files:

```
cradle install {{[-f|--force]}}
```

- Skip running SQL migrations:

```
cradle install --skip-sql
```

- Skip running package updates:

```
cradle install --skip-versioning
```

- Use specific database details:

```
cradle install -h {{hostname}} -u {{username}} -p
{{password}}
```

cradle package

Manage packages for a Cradle instance.

More information: https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#package.

- Display a list of available packages:

```
cradle package list
```

- Search for a package:

```
cradle package search {{package}}
```

- Install a package from Packagist:

```
cradle package install {{package}}
```

- Install a specific version of a package:

```
cradle package install {{package}} {{version}}
```

- Update a package:

```
cradle package update {{package}}
```

- Update a package to a specific version:

```
cradle package update {{package}} {{version}}
```

- Remove a specific package:

```
cradle package remove {{package}}
```

cradle sql

Manage Cradle SQL databases.

More information: https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#sql.

- Rebuild the database schema:

```
cradle sql build
```

- Rebuild the database schema for a specific package:

```
cradle sql build {{package}}
```

- Empty the entire database:

```
cradle sql flush
```

- Empty the database tables for a specific package:

```
cradle sql flush {{package}}
```

- Populate the tables for all packages:

```
cradle sql populate
```

- Populate the tables for a specific package:

```
cradle sql populate {{package}}
```

cradle

The Cradle PHP framework.

Some subcommands such as **`install`** have their own usage documentation.

More information: https://cradlephp.github.io.

- Connect to a server:

  ```
  cradle connect {{server_name}}
  ```

- Execute a Cradle command:

  ```
  cradle {{command}}
  ```

- Display help:

  ```
  cradle help
  ```

- Display help for a specific command:

  ```
  cradle {{command}} help
  ```

crane append

Push an image based on an (optional) base image.

Appends layers containing the contents of the provided tarballs.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_append.md.

- Push image based on a base image:

```
crane append {{[-b|--base]}} {{image_name}}
```

- Push image with appended layer from tarball:

```
crane append {{[-f|--new_layer]}} {{layer_name1 layer_name2 ...}}
```

- Push image with appended layer with new tag:

```
crane append {{[-t|--new_tag]}} {{tag_name}}
```

- Push resulting image to new tarball:

```
crane append {{[-o|--output]}} {{path/to/tarball}}
```

- Use empty base image of type OCI media instead of Docker:

```
crane append --oci-empty-base
```

- Annotate resulting image as being based on the base image:

```
crane append --set-base-image-annotations
```

- Display help:

```
crane append {{[-h|--help]}}
```

crane auth

Log in or access credentials.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_auth.md.

- Execute `crane auth` subcommand:

```
crane auth {{subcommand}}
```

- Implement credential helper:

```
crane auth get {{registry_address}} {{[-h|--help]}}
```

- Log in to a registry:

```
crane auth login {{registry_address}} {{[-h|--help]}} {{[-p|--password]}} {{password}} {{-password-stdin}} {{[-u|--username]}} {{username}}
```

- Log out of a registry:

```
crane auth logout {{registry_address}} {{[-h|--help]}}
```

- Retrieve a token for a remote repository:

```
crane auth token {{registry_address}} {{[-H|--header]}} {{[-h|--help]}} {{[-m|--mount]}} {{scope1 scope2 ...}} --push
```

- Display help:

```
crane auth {{[-h|--help]}}
```

crane blob

Read a blob from a registry.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_blob.md.

- Read the blob from a registry:

```
crane blob {{blob_identifier}}
```

- Display help:

```
crane blob {{[-h|--help]}}
```

crane catalog

List the repositories in a registry.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_catalog.md.

- List the repositories in a registry:

```
crane catalog {{registry_address}}
```

- Print the full image reference:

```
crane catalog {{registry_address}} --full-ref
```

- Display help:

```
crane catalog {{[-h|--help]}}
```

crane config

Get the configuration of an image.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_config.md.

- Get the configuration of an image:

```
crane config {{image_name}}
```

- Display help:

```
crane config {{[-h|--help]}}
```

crane copy

Efficiently copy a remote image from source to target while retaining the digest value.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_copy.md.

- Copy an image from source to target:

```
crane copy {{source}} {{target}}
```

- Copy all tags:

```
crane copy {{source}} {{target}} {{[-a|--all-tags]}}
```

- Set the maximum number of concurrent copies, defaults to GOMAXPROCS:

```
crane copy {{source}} {{target}} {{[-j|--jobs]}} {{int}}
```

- Avoid overwriting existing tags in target:

```
crane copy {{source}} {{target}} {{[-n|--no-clobber]}}
```

- Display help:

```
crane copy {{[-h|--help]}}
```

crane cp

This command is an alias of **`crane copy`**.

- View documentation for the original command:

```
tldr crane copy
```

crane delete

Delete an image reference from its registry.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_delete.md.

- Delete an image reference from its registry:

```
crane delete {{image_name}}
```

- Display help:

```
crane delete {{[-h|--help]}}
```

crane digest

Get the digest of an image.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_digest.md.

- Get the digest of an image:

```
crane digest {{image_name}}
```

- Print the full image reference by digest:

```
crane digest {{image_name}} --full-ref
```

- Specify path to tarball containing the image:

```
crane digest {{image_name}} --tarball {{path/to/tarball}}
```

- Display help:

```
crane digest {{[-h|--help]}}
```

crane export

Export filesystem of a container image as a tarball.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_digest.md.

- Write tarball to `stdout`:

```
crane export {{image_name}} -
```

- Write tarball to file:

```
crane export {{image_name}} {{path/to/tarball}}
```

- Read image from `stdin`:

```
crane export - {{path/to/file}}
```

crane flatten

Flatten an image's layers into a single layer.

Pushes digest to original image repository if no tags are specified.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_digest.md.

- Flatten an image:

  ```
  crane flatten
  ```

- Apply new tag to flattened image:

  ```
  crane flatten {{[-t|--tag]}} {{tag_name}}
  ```

- Display help:

  ```
  crane flatten {{[-h|--help]}}
  ```

crane index append

Append manifest to a remote index.

This sub-command pushes an index based on an (optional) base index, with appended manifests.

The platform for appended manifests is inferred from the configuration file or omitted if that is infeasible.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_index_append.md.

- Append manifest to a remote index:

```
crane index append
```

- Reference to manifests to append to the base index:

```
crane index append {{[-m|--manifest]}} {{manifest_name1 manifest_name2 ...}}
```

- Tag to apply to resulting image:

```
crane index append {{[-t|--tag]}} {{tag_name}}
```

- Empty base index will have Docker media types instead of OCI:

```
crane index append --docker-empty-base
```

- Append each of its children rather than the index itself (defaults true):

```
crane index append --flatten
```

- Display help:

```
crane index append {{[-h|--help]}}
```

crane index filter

Modifies a remote index by filtering based on platform.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_index_filter.md.

- Modify remote index:

```
crane index filter
```

- Specify the platform(s) to keep from base in the form `os/arch/variant:osversion,platform`:

```
crane index filter --platform {{platform1 platform2 ...}}
```

- Tag to apply to resulting image:

```
crane index filter {{[-t|--tags]}} {{tag_name}}
```

- Display help:

```
crane index filter {{[-h|--help]}}
```

crane index

Modify an image index.

The subcommands **`append`** and **`filter`** have their own usage documentation.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_index.md.

- Modify an image index:

```
crane index
```

- Modify an image index with subcommand:

```
crane index {{subcommand}}
```

- Display help:

```
crane index {{[-h|--help]}}
```

crane ls

List the tags in a repository.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_ls.md.

- List the tags:

```
crane ls {{repository}}
```

- Print the full image reference:

```
crane ls {{repository}} --full-ref
```

- Omit digest tags:

```
crane ls {{[-o|--omit-digest-tags]}}
```

- Display help:

```
crane ls {{[-h|--help]}}
```

crane manifest

Get the manifest of an image.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_manifest.md.

- Get the manifest:

```
crane manifest {{image_name}}
```

- Display help:

```
crane manifest {{[-h|--help]}}
```

crane mutate

Modify image labels and annotations.

The container must be pushed to a registry, and the manifest is updated there.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_mutate.md.

- New annotations to set (default []):

```
crane mutate {{[-a|--annotation]}}/{{[-l|--label]}} {{annotation/label}}
```

- Path to tarball/command/entrypoint/environment variable/exposed-ports to append to image:

```
crane mutate {{--append}}/{{--cmd}}/{{--entrypoint}}/{{[-e|--env]}}/{{--exposed-ports}} {{var1 var2 ...}}
```

- Path to new tarball of resulting image:

```
crane mutate {{[-o|--output]}} {{path/to/tarball}}
```

- Repository in the form `os/arch/variant:osversion,platform` to push mutated image:

```
crane mutate --set-platform {{platform_name}}
```

- New tag reference to apply to mutated image:

```
crane mutate {{[-t|--tag]}} {{tag_name}}
```

- New user to set:

```
crane mutate {{[-u|--user]}} {{username}}
```

- New working dir to set:

```
crane mutate {{[-w|--workdir]}} {{path/to/work_directory}}
```

- Display help:

```
crane mutate {{[-h|--help]}}
```

crane pull

Pull remote images by reference and store their contents locally.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_pull.md.

- Pull remote image:

```
crane pull {{image_name}} {{path/to/tarball}}
```

- Preserve image reference used to pull as an annotation when used with `--format=oci`:

```
crane pull {{image_name}} {{path/to/tarball}} --annotate-ref
```

- Path to cache image layers:

```
crane pull {{image_name}} {{path/to/tarball}} {{[-c|--cache_path]}} {{path/to/cache}}
```

- Format in which to save images (default `tarball`):

```
crane pull {{image_name}} {{path/to/tarball}} {{-format}} {{format_name}}
```

- Display help:

```
crane pull {{[-h|--help]}}
```

crane push

Push local image contents to a remote registry.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_push.md.

- Push local image to remote registry:

```
crane push {{path/to/tarball}} {{image_name}}
```

- Path to file with list of published image references:

```
crane push {{path/to/tarball}} {{image_name}} --image-refs {{path/to/file}}
```

- Push a collection of images as a single index (required if path has multiple images):

```
crane push {{path/to/tarball}} {{image_name}} --index
```

- Display help:

```
crane push {{[-h|--help]}}
```

crane rebase

Rebase an image onto a new base image.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_rebase.md.

- Rebase image:

```
crane rebase
```

- New base image to insert:

```
crane rebase --new_base {{image_name}}
```

- Old base image to remove:

```
crane rebase --old_base {{image_name}}
```

- Tag to apply to rebased image:

```
crane rebase {{[-t|--tag]}} {{tag_name}}
```

- Display help:

```
crane rebase {{[-h|--help]}}
```

crane registry

This command serves a registry implementation on an automatically chosen port (:0), $PORT or --address.

The command blocks while the server accepts pushes and pulls and contents are can be stored in memory, and disk.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_registry_serve.md.

- Serve a registry implementation:

```
crane registry serve
```

- Address to listen on:

```
crane registry serve --address {{address_name}}
```

- Path to a directory where blobs will be stored:

```
crane registry serve --disk {{path/to/store_directory}}
```

- Display help for `crane registry`:

```
crane registry {{[-h|--help]}}
```

- Display help for `crane registry serve`:

```
crane registry serve {{[-h|--help]}}
```

crane tag

Efficiently tag a remote image without downloading it, which differs from the **copy** command.

It skips the layer existence checks because we know the manifest already exists making it slightly faster.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_tag.md.

- Tag remote image:

```
crane tag {{image_name}} {{tag_name}}
```

- Display help:

```
crane tag {{[-h|--help]}}
```

crane validate

Validate that an image is well-formed.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_validate.md.

- Validate an image:

```
crane validate
```

- Skip downloading/digesting layers:

```
crane validate --fast
```

- Name of remote image to validate:

```
crane validate --remote {{image_name}}
```

- Path to tarball to validate:

```
crane validate --tarball {{path/to/tarball}}
```

- Display help:

```
crane validate {{[-h|--help]}}
```

crane version

Print the version of a binary.

The version string is completely dependent on how the binary was built, so you should not depend on the version format. It may change without notice.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane_version.md.

- Display version:

  ```
  crane version
  ```

- Display help:

  ```
  crane version {{[-h|--help]}}
  ```

crane

Container images managing tool.

Some subcommands such as **`pull`**, **`push`**, **`copy`**, etc. have their own usage documentation.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/crane/doc/crane.md/.

- Log in to a registry:

```
crane auth login {{registry}} {{[-u|--username]}} {{user}} {{[-p|--password]}} {{password}}
```

- List the repos in a registry:

```
crane catalog {{registry}} --full-ref
```

- List the tags in a repository:

```
crane ls {{repository}} {{[-o|--omit-digest-tags]}}
```

- Pull remote images by reference and store their contents locally:

```
crane pull {{image}} {{tarball}}
```

- Push local image contents to a remote registry:

```
crane push {{path/to/directory_or_tarball}} {{image}}
```

- Efficiently tag a remote image:

```
crane tag {{image}} {{tag}}
```

- Efficiently copy a remote image from `src` to `dst` while retaining the digest value:

```
crane copy src dst {{[-a|--all-tags]}}
```

- Delete an image reference from its registry:

```
crane delete {{image}}
```

createdb

Create a PostgreSQL database.

More information: https://www.postgresql.org/docs/current/app-createdb.html.

- Create a database owned by the current user:

```
createdb {{database_name}}
```

- Create a database owned by a specific user with a description:

```
createdb {{[-O|--owner]}} {{username}} {{database_name}}
'{{description}}'
```

- Create a database from a template:

```
createdb {{[-T|--template]}} {{template_name}}
{{database_name}}
```

createuser

Create a PostgreSQL user (role).

More information: https://www.postgresql.org/docs/current/app-createuser.html.

- Create a user interactively:

```
createuser --interactive {{username}}
```

- Create a user with no special rights:

```
createuser {{username}}
```

- Create a superuser:

```
createuser {{[-s|--superuser]}} {{username}}
```

- Create a user allowed to create databases, manage roles, and prompt for a password:

```
createuser {{[-d|--createdb]}} {{[-r|--createrole]}} {{[-P|--pwprompt]}} {{username}}
```

- Create a user without the ability to create databases or manage roles:

```
createuser {{[-D|--no-createdb]}} {{[-R|--no-createrole]}} {{username}}
```

crictl

Manage CRI-compatible container runtimes.

More information: https://github.com/kubernetes-sigs/cri-tools/blob/master/docs/crictl.md.

- List all kubernetes pods (Ready and NotReady):

```
crictl pods
```

- List all containers (Running and Exited):

```
crictl ps {{[-a|--all]}}
```

- List all images:

```
crictl images
```

- Print information about specific containers:

```
crictl inspect {{container_id1 container_id2 ...}}
```

- Open a specific shell inside a running container:

```
crictl exec {{[-it|--interactive --tty]}} {{container_id}} {{sh}}
```

- Pull a specific image from a registry:

```
crictl pull {{image:tag}}
```

- Print and follow logs of a specific container:

```
crictl logs {{[-f|--follow]}} {{container_id}}
```

- Remove one or more images:

```
crictl rmi {{image_id1 image_id2 ...}}
```

croc

Send and receive files easily and securely over any network.

More information: https://github.com/schollz/croc.

- Send a file or directory:

```
croc send {{path/to/file_or_directory}}
```

- Send a file or directory with a specific passphrase:

```
croc send {{[-c|--code]}} {{passphrase}} {{path/to/
file_or_directory}}
```

- Receive a file or directory on receiving machine:

```
croc {{passphrase}}
```

- Send and connect over a custom relay:

```
croc --relay {{ip_to_relay}} send {{path/to/
file_or_directory}}
```

- Receive and connect over a custom relay:

```
croc --relay {{ip_to_relay}} {{passphrase}}
```

- Host a croc relay on the default ports:

```
croc relay
```

- Display parameters and options for a croc command:

```
croc {{send|relay}} --help
```

cron

A system scheduler for running jobs or tasks unattended.

The command to submit, edit or delete entries to **`cron`** is called **`crontab`**.

- View documentation for managing `cron` entries:

  ```
  tldr crontab
  ```

cronic

Bash script for wrapping cron jobs to prevent excess email sending.

More information: https://manned.org/cronic.

- Call a command and display its output if it returns a non-zero exit code:

```
cronic {{command}}
```

crontab

Schedule cron jobs to run on a time interval for the current user.

More information: https://manned.org/crontab.

- Edit the crontab file for the current user:

```
crontab -e
```

- Edit the crontab file for a specific user:

```
sudo crontab -e -u {{user}}
```

- Replace the current crontab with the contents of the given file:

```
crontab {{path/to/file}}
```

- View a list of existing cron jobs for current user:

```
crontab -l
```

- Remove all cron jobs for the current user:

```
crontab -r
```

- Sample job which runs at 10:00 every day (* means any value):

```
0 10 * * * {{command_to_execute}}
```

- Sample crontab entry, which runs a command every 10 minutes:

```
*/10 * * * * {{command_to_execute}}
```

- Sample crontab entry, which runs a certain script at 02:30 every Friday:

```
30 2 * * Fri /{{path/to/script.sh}}
```

crunch

Wordlist generator.

More information: https://manned.org/crunch.

- Output a list of words of length 1 to 3 with only lowercase characters:

```
crunch {{1}} {{3}}
```

- Output a list of hexadecimal words of length 8:

```
crunch {{8}} {{8}} {{0123456789abcdef}}
```

- Output a list of all permutations of abc (lengths are not processed):

```
crunch {{1}} {{1}} -p {{abc}}
```

- Output a list of all permutations of the given strings (lengths are not processed):

```
crunch {{1}} {{1}} -p {{abc}} {{def}} {{ghi}}
```

- Output a list of words generated according to the given pattern and a maximum number of duplicate letters:

```
crunch {{5}} {{5}} {{abcde123}} -t {{@@@12}} -d 2@
```

- Write a list of words in chunk files of a given size, starting with the given string:

```
crunch {{3}} {{5}} -o {{START}} -b {{10kb}} -s {{abc}}
```

- Write a list of words stopping with the given string and inverting the wordlist:

```
crunch {{1}} {{5}} -o {{START}} -e {{abcde}} -i
```

- Write a list of words in compressed chunk files with a specified number of words:

```
crunch {{1}} {{5}} -o {{START}} -c {{1000}} -z {{gzip|bzip2|lzma|7z}}
```

crush

AI-powered terminal assistant for software development tasks.

Provides interactive chat interface with AI capabilities, code analysis, and LSP integration.

More information: https://github.com/charmbracelet/crush.

- Start interactive mode:

```
crush
```

- Run with debug logging:

```
crush {{[-d|--debug]}}
```

- Run with debug logging in a specific directory:

```
crush {{[-d|--debug]}} {{[-c|--cwd]}} {{path/to/project}}
```

- Run a single non-interactive prompt:

```
crush run "{{Explain the use of context in Go}}"
```

- Run in dangerous mode (auto-accept all permissions):

```
crush {{[-y|--yolo]}}
```

- Display version:

```
crush {{[-v|--version]}}
```

cryfs

A cryptographic filesystem for the cloud.

More information: https://manned.org/cryfs.

- Mount an encrypted filesystem. The initialization wizard will be started on the first execution:

```
cryfs {{path/to/cipher_directory}} {{path/to/mount_point}}
```

- Unmount an encrypted filesystem:

```
cryfs-unmount {{path/to/mount_point}}
```

- Automatically unmount after ten minutes of inactivity:

```
cryfs --unmount-idle {{10}} {{path/to/cipher_directory}} {{path/to/mount_point}}
```

- List supported ciphers:

```
cryfs --show-ciphers
```

crystal

Manage Crystal source code.

More information: https://crystal-lang.org/reference/using_the_compiler.

- Run a Crystal file:

```
crystal {{path/to/file.cr}}
```

- Compile a file and all dependencies to a single executable:

```
crystal build {{path/to/file.cr}}
```

- Read Crystal source code from the command-line or `stdin`, and execute it:

```
crystal eval '{{code}}'
```

- Generate API documentation from inline docstrings in Crystal files:

```
crystal docs
```

- Compile and run a Crystal specification suite:

```
crystal spec
```

- Start a local interactive server for testing the language:

```
crystal play
```

- Create a project directory for a Crystal application:

```
crystal init app {{application_name}}
```

- Display help:

```
crystal help
```

cs complete-dep

Search for libraries without doing it directly on the web.

More information: https://get-coursier.io/docs/cli-complete.

- Print which artifacts are published under a specific Maven group identifier:

```
cs complete-dep {{group_id}}
```

- List published library versions under a specific Maven group identifier and an artifact one:

```
cs complete-dep {{group_id}}:{{artifact_id}}
```

- Print which artifacts are pubblished under a given Maven groupId searching in the ivy2local:

```
cs complete-dep {{group_id}} --repository ivy2local
```

- List published artifacts under a Maven group identifier searching in a specific repository and credentials:

```
cs complete-dep {{group_id}}:{{artifact_id}} --repository
{{repository_url}} --credentials {{user}}:{{password}}
```

cs fetch

Fetch fetches the JARs of dependencies.

More information: https://get-coursier.io/docs/cli-fetch.

- Fetch a specific version of a jar:

```
cs fetch {{group_id}}:{{artifact_id}}:{{artifact_version}}
```

- Fetch a package and evaluate the classpath corresponding to the selected package in an env var:

```
CP="$(cs fetch --classpath org.scalameta::scalafmt-cli:latest.release)"
```

- Fetch a source of a specific jar:

```
cs fetch --sources {{group_id}}:{{artifact_id}}:{{artifact_version}}
```

- Fetch the javadoc jars:

```
cs fetch --javadoc {{group_id}}:{{artifact_id}}:{{artifact_version}}
```

- Fetch dependency with javadoc jars and source jars:

```
cs fetch --default={{true}} --sources --javadoc {{group_id}}:{{artifact_id}}:{{artifact_version}}
```

- Fetch jars coming from dependency files:

```
cs fetch {{--dependency-file path/to/file1 --dependency-file path/to/file2 ...}}
```

cs install

Install an application in the installation directory onfigured when installing **`cs`** (to enable the binary to be loaded add to your **`.bash_profile`** the **`$ eval "$(cs install --env)"`** command).

More information: https://get-coursier.io/docs/cli-install.

- Install a specific application:

```
cs install {{application_name}}
```

- Install a specific version of an application:

```
cs install {{application_name}}:{{application_version}}
```

- Search an application by a specific name:

```
cs search {{application_partial_name}}
```

- Update a specific application if available:

```
cs update {{application_name}}
```

- Update all the installed applications:

```
cs update
```

- Uninstall a specific application:

```
cs uninstall {{application_name}}
```

- List all installed applications:

```
cs list
```

- Pass specific Java options to an installed application:

```
{{application_name}} {{-Jjava_option_name1=value1 -
Jjava_option_name2=value2 ...}}
```

cs java

The **`java`** and **`java-home`** commands fetch and install JVMs. The **`java`** command runs them too.

More information: https://get-coursier.io/docs/cli-java.

- Display Java version by using coursier:

```
cs java -version
```

- Call a specific Java version with custom properties using coursier:

```
cs java --jvm {{jvm_name}}:{{jvm_version}} -Xmx32m -
X{{another_jvm_opt}} -jar {{path/to/jar_name.jar}}
```

- List all the available JVM in the coursier default index:

```
cs java --available
```

- List all the installed JVM in the system with his own location:

```
cs java --installed
```

- Set a specific JVM as one-off default for the shell instance:

```
cs java --jvm {{jvm_name}}:{{jvm_version}} --env
```

- Revert the changes for the default JVM settings:

```
eval "$(cs java --disable)"
```

- Set a specific JVM as default for the whole system:

```
cs java --jvm {{jvm_name}}:{{jvm_version}} --setup
```

cs launch

Launch an application from the name directly from Maven dependencies without the need of installing it.

More information: https://get-coursier.io/docs/cli-launch.

- Launch a specific application with arguments:

```
cs launch {{application_name}} -- {{argument1 argument2 ...}}
```

- Launch a specific application version with arguments:

```
cs launch {{application_name}}:{{application_version}} --
{{argument1 argument2 ...}}
```

- Launch a specific version of an application specifying which is the main file:

```
cs launch {{group_id}}:{{artifact_id}}:{{artifact_version}}
--main-class {{path/to/main_class_file}}
```

- Launch an application with specific Java options and JVM memory ones:

```
cs launch --java-opt {{-Doption_name1:option_value1 -
Doption_name2:option_value2 ...}} --java-opt {{-Xjvm_option1
-Xjvm_option2 ...}} {{application_name}}
```

cs resolve

Resolve lists the transitive dependencies of other dependencies.

More information: https://get-coursier.io/docs/cli-resolve.

- Resolve lists of transitive dependencies of two dependencies:

```
cs resolve {{group_id1}}:{{artifact_id1}}:
{{artifact_version1}} {{group_id2}}:{{artifact_id2}}:
{{artifact_version2}}
```

- Resolve lists of transitive dependencies of a package by the dependency tree:

```
cs resolve --tree {{group_id}}:{{artifact_id}}:
{{artifact_version}}
```

- Resolve dependency tree in a reverse order (from a dependency to its dependencies):

```
cs resolve --reverse-tree {{group_id}}:{{artifact_id}}:
{{artifact_version}}
```

- Print all the libraries that depends on a specific library:

```
cs resolve {{group_id}}:{{artifact_id}}:{{artifact_version}}
--what-depends-on {{searched_group_id}}:
{{searched_artifact_id}}
```

- Print all the libraries that depends on a specific library version:

```
cs resolve {{group_id}}:{{artifact_id}}:{{artifact_version}}
--what-depends-on {{searched_group_id}}:
{{searched_artifact_id}}{{searched_artifact_version}}
```

- Print eventual conflicts between a set of packages:

```
cs resolve --conflicts
{{group_id1:artifact_id1:artifact_version1
group_id2:artifact_id2:artifact_version2 ...}}
```

cs

Application and artifact manager for the Scala language.

Installs Scala applications and sets up a Scala development environment.

Some subcommands such as **`install`**, **`launch`**, **`java`**, **`fetch`**, **`resolve`**, **`complete-dep`**, etc. have their own usage documentation.

More information: https://get-coursier.io/docs/overview.

- List installed applications:

```
cs list
```

- Install a specific application:

```
cs install {{application_name}}
```

- Uninstall a specific application:

```
cs uninstall {{application_name}}
```

- Setup machine for the Scala development:

```
cs setup
```

- Update all the installed applications:

```
cs update
```

- Display version:

```
cs version
```

csc

The Microsoft C# Compiler.

More information: https://learn.microsoft.com/dotnet/csharp/language-reference/compiler-options/command-line-building-with-csc-exe.

- Compile one or more C# files to a CIL executable:

```
csc {{path/to/input_file_a.cs}} {{path/to/input_file_b.cs}}
```

- Specify the output filename:

```
csc /out:{{path/to/file}} {{path/to/input_file.cs}}
```

- Compile into a `.dll` library instead of an executable:

```
csc /target:library {{path/to/input_file.cs}}
```

- Reference another assembly:

```
csc /reference:{{path/to/library.dll}} {{path/to/
input_file.cs}}
```

- Embed a resource:

```
csc /resource:{{path/to/resource_file}} {{path/to/
input_file.cs}}
```

- Automatically generate XML documentation:

```
csc /doc:{{path/to/output.xml}} {{path/to/input_file.cs}}
```

- Specify an icon:

```
csc /win32icon:{{path/to/icon.ico}} {{path/to/input_file.cs}}
```

- Strongly-name the resulting assembly with a keyfile:

```
csc /keyfile:{{path/to/keyfile}} {{path/to/input_file.cs}}
```

csh

The shell (command interpreter) with C-like syntax.

See also: **`tcsh`**.

More information: https://www.mkssoftware.com/docs/man1/csh.1.asp.

- Start an interactive shell session:

```
csh
```

- Start an interactive shell session [f]aster without loading startup configs:

```
csh -f
```

- Execute specific [c]ommands:

```
csh -c "{{echo 'csh is executed'}}"
```

- Execute a specific script:

```
csh {{path/to/script.csh}}
```

csslint

Lint CSS code.

More information: https://github.com/CSSLint/csslint/wiki/Command-line-interface.

- Lint a single CSS file:

```
csslint {{file.css}}
```

- Lint multiple CSS files:

```
csslint {{file1.css file2.css ...}}
```

- List all possible style rules:

```
csslint --list-rules
```

- Treat certain rules as errors (which results in a non-zero exit code):

```
csslint --errors={{errors,universal-selector,imports}}
{{file.css}}
```

- Treat certain rules as warnings:

```
csslint --warnings={{box-sizing,selector-max,floats}}
{{file.css}}
```

- Ignore specific rules:

```
csslint --ignore={{ids,rules-count,shorthand}} {{file.css}}
```

csv-diff

View differences between two CSV, TSV or JSON files.

More information: https://github.com/simonw/csv-diff.

- Display a human-readable summary of differences between files using a specific column as a unique identifier:

```
csv-diff {{path/to/file1.csv}} {{path/to/file2.csv}} --key {{column_name}}
```

- Display a human-readable summary of differences between files that includes unchanged values in rows with at least one change:

```
csv-diff {{path/to/file1.csv}} {{path/to/file2.csv}} --key {{column_name}} --show-unchanged
```

- Display a summary of differences between files in JSON format using a specific column as a unique identifier:

```
csv-diff {{path/to/file1.csv}} {{path/to/file2.csv}} --key {{column_name}} --json
```

csv2tsv

Convert CSV (comma-separated) text to TSV (tab-separated) format.

More information: https://github.com/eBay/tsv-utils/blob/master/README.md#csv2tsv.

- Convert from CSV to TSV:

```
csv2tsv {{path/to/input_csv1 path/to/input_csv2 ...}} > {{path/to/output_tsv}}
```

- Convert field delimiter separated CSV to TSV:

```
csv2tsv -c'{{field_delimiter}}' {{path/to/input_csv}}
```

- Convert semicolon separated CSV to TSV:

```
csv2tsv -c';' {{path/to/input_csv}}
```

csvclean

Find and clean common syntax errors in CSV files.

Included in csvkit.

More information: https://csvkit.readthedocs.io/en/latest/scripts/csvclean.html.

- Clean a CSV file:

```
csvclean {{bad.csv}}
```

- List locations of syntax errors in a CSV file:

```
csvclean -n {{bad.csv}}
```

csvcut

Filter and truncate CSV files. Like Unix's **`cut`** command, but for tabular data.

Included in csvkit.

More information: https://csvkit.readthedocs.io/en/latest/scripts/csvcut.html.

- Print indices and names of all columns:

```
csvcut {{[-n|--names]}} {{data.csv}}
```

- Extract the first and third columns:

```
csvcut {{[-c|--columns]}} {{1,3}} {{data.csv}}
```

- Extract all columns except the fourth one:

```
csvcut {{[-C|--not-columns]}} {{4}} {{data.csv}}
```

- Extract the columns named "id" and "first name" (in that order):

```
csvcut {{[-c|--columns]}} {{id,"first name"}} {{data.csv}}
```

csvformat

Convert a CSV file to a custom output format.

Included in csvkit.

More information: https://csvkit.readthedocs.io/en/latest/scripts/csvformat.html.

- Convert to a tab-delimited file (TSV):

```
csvformat {{[-T|--out-tabs]}} {{data.csv}}
```

- Convert delimiters to a custom character:

```
csvformat {{[-D|--out-delimiter]}}
"{{custom_character}}" {{data.csv}}
```

- Convert line endings to carriage return (^M) + line feed:

```
csvformat {{[-M|--out-lineterminator]}}
"{{\r\n}}" {{data.csv}}
```

- Minimize use of quote characters:

```
csvformat {{[-U|--out-quoting]}} 0 {{data.csv}}
```

- Maximize use of quote characters:

```
csvformat {{[-U|--out-quoting]}} 1 {{data.csv}}
```

csvgrep

Filter CSV rows with string and pattern matching.

Included in csvkit.

More information: https://csvkit.readthedocs.io/en/latest/scripts/csvgrep.html.

- Find rows that have a certain string in column 1:

```
csvgrep {{[-c|--columns]}} {{1}} {{[-m|--match]}} {{string_to_match}} {{data.csv}}
```

- Find rows in which columns 3 or 4 match a certain `regex`:

```
csvgrep {{[-c|--columns]}} {{3,4}} {{[-r|--regex]}} {{regex}} {{data.csv}}
```

- Find rows in which the "name" column does NOT include the string "John Doe":

```
csvgrep {{[-i|--invert-match]}} {{[-c|--columns]}} {{name}} {{[-m|--match]}} "{{John Doe}}" {{data.csv}}
```

csvlook

Render a CSV file in the console as a fixed-width table.

Included in csvkit.

More information: https://csvkit.readthedocs.io/en/latest/scripts/csvlook.html.

- View a CSV file:

```
csvlook {{data.csv}}
```

csvpy

Load a CSV file into a Python shell.

Included in csvkit.

More information: https://csvkit.readthedocs.io/en/latest/scripts/csvpy.html.

- Load a CSV file into a `CSVKitReader` object:

  ```
  csvpy {{data.csv}}
  ```

- Load a CSV file into a `CSVKitDictReader` object:

  ```
  csvpy --dict {{data.csv}}
  ```

csvsort

Sort CSV files.

Included in csvkit.

More information: https://csvkit.readthedocs.io/en/latest/scripts/csvsort.html.

- Sort a CSV file by column 9:

```
csvsort {{[-c|--columns]}} {{9}} {{data.csv}}
```

- Sort a CSV file by the "name" column in descending order:

```
csvsort {{[-r|--reverse]}} {{[-c|--columns]}} {{name}}
{{data.csv}}
```

- Sort a CSV file by column 2, then by column 4:

```
csvsort {{[-c|--columns]}} {{2,4}} {{data.csv}}
```

- Sort a CSV file without inferring data types:

```
csvsort {{[-I|--no-inference]}} {{[-c|--columns]}}
{{columns}} {{data.csv}}
```

csvsql

Generate SQL statements for a CSV file or execute those statements directly on a database.

Included in csvkit.

More information: https://csvkit.readthedocs.io/en/latest/scripts/csvsql.html.

- Generate a `CREATE TABLE` SQL statement for a CSV file:

```
csvsql {{path/to/data.csv}}
```

- Import a CSV file into an SQL database:

```
csvsql --insert --db "{{mysql://user:password@host/
database}}" {{data.csv}}
```

- Run an SQL query on a CSV file:

```
csvsql --query "{{select * from 'data'}}" {{data.csv}}
```

csvstat

Print descriptive statistics for all columns in a CSV file.

Included in csvkit.

More information: https://csvkit.readthedocs.io/en/latest/scripts/csvstat.html.

- Show all stats for all columns:

```
csvstat {{data.csv}}
```

- Show all stats for columns 2 and 4:

```
csvstat {{[-c|--columns]}} {{2,4}} {{data.csv}}
```

- Show sums for all columns:

```
csvstat --sum {{data.csv}}
```

- Show the max value length for column 3:

```
csvstat {{[-c|--columns]}} {{3}} --len {{data.csv}}
```

- Show the number of unique values in the "name" column:

```
csvstat {{[-c|--columns]}} {{name}} --unique {{data.csv}}
```

csvtool

Utility to filter and extract data from CSV formatted sources.

More information: https://github.com/maroofi/csvtool.

- Extract the second column from a CSV file:

```
csvtool {{[-c|--column]}} {{2}} {{path/to/file.csv}}
```

- Extract the second and fourth columns from a CSV file:

```
csvtool {{[-c|--column]}} {{2,4}} {{path/to/file.csv}}
```

- Extract lines from a CSV file where the second column exactly matches `String`:

```
csvtool {{[-c|--column]}} {{2}} {{[-s|--search]}}
'{{^String$}}' {{path/to/file.csv}}
```

- Extract lines from a CSV file where the second column starts with `Bar`:

```
csvtool {{[-c|--column]}} {{2}} {{[-s|--search]}}
'{{^Bar}}' {{path/to/file.csv}}
```

- Find lines in a CSV file where the second column ends with `Baz` and then extract the third and sixth columns:

```
csvtool {{[-c|--column]}} {{2}} {{[-s|--search]}}
'{{Baz$}}' {{path/to/file.csv}} | csvtool {{[-e|--no-
header]}} {{[-c|--column]}} {{3,6}}
```

ctags

Generate an index (or tag) file of language objects found in source files for many popular programming languages.

More information: https://docs.ctags.io/en/latest/man/ctags.1.html.

- Generate tags for a single file, and output them to a file named "tags" in the current directory, overwriting the file if it exists:

```
ctags {{path/to/file}}
```

- Generate tags for all files in the current directory, and output them to a specific file, overwriting the file if it exists:

```
ctags -f {{path/to/file}} *
```

- Generate tags for all files in the current directory and all subdirectories:

```
ctags --recurse
```

- Generate tags for a single file, and output them with start line number and end line number in JSON format:

```
ctags --fields=+ne --output-format=json {{path/to/file}}
```

ctest

CMake test driver program.

More information: https://gitlab.kitware.com/cmake/community/wikis/doc/ctest/Testing-With-CTest.

- Run all tests defined in the CMake project, executing 4 [j]obs at a time in parallel:

```
ctest {{[-j|--parallel]}} {{4}} --output-on-failure
```

- List available tests:

```
ctest {{[-N|--show-only]}}
```

- Run a single test based on its name, or filter on a `regex`:

```
ctest --output-on-failure {{[-R|--tests-regex]}}
'^{{test_name}}$'
```

cups-config

Show technical information about your CUPS print server installation.

More information: https://openprinting.github.io/cups/doc/man-cups-config.html.

- Show where CUPS is currently installed:

```
cups-config --serverbin
```

- Show the location of CUPS' configuration directory:

```
cups-config --serverroot
```

- Show the location of CUPS' data directory:

```
cups-config --datadir
```

- Display help:

```
cups-config --help
```

- Display CUPS version:

```
cups-config --version
```

CUPS

Open source printing system.

CUPS isn't a single command, but a set of commands.

More information: https://www.cups.org/index.html.

- View documentation for running the CUPS daemon:

```
tldr cupsd
```

- View documentation for managing printers:

```
tldr lpadmin
```

- View documentation for printing files:

```
tldr lp
```

- View documentation for checking status information about the current classes, jobs, and printers:

```
tldr lpstat
```

- View documentation for cancelling print jobs:

```
tldr lprm
```

cupsaccept

Accept jobs sent to destinations.

Note: Destination is referred as a printer or a class of printers.

See also: **`cupsreject`**, **`cupsenable`**, **`cupsdisable`**, **`lpstat`**.

More information: https://www.cups.org/doc/man-cupsaccept.html.

- Accept print jobs to the specified destinations:

```
cupsaccept {{destination1 destination2 ...}}
```

- Specify a different server:

```
cupsaccept -h {{server}} {{destination1 destination2 ...}}
```

cupsctl

Update or query a server's **`cupsd.conf`**.

More information: https://openprinting.github.io/cups/doc/man-cupsctl.html.

- Display the current configuration values:

```
cupsctl
```

- Display the configuration values of a specific server:

```
cupsctl -h {{server[:port]}}
```

- Enable encryption on the connection to the scheduler:

```
cupsctl -E
```

- Enable or disable debug logging to the `error_log` file:

```
cupsctl {{--debug-logging|--no-debug-logging}}
```

- Enable or disable remote administration:

```
cupsctl {{--remote-admin|--no-remote-admin}}
```

- Parse the current debug logging state:

```
cupsctl | grep '^_debug_logging' | awk -F= '{print $2}'
```

cupsd

Server daemon for the CUPS print server.

More information: https://openprinting.github.io/cups/doc/man-cupsd.html.

- Start `cupsd` in the background, aka. as a daemon:

  `cupsd`

- Start `cupsd` on the [f]oreground:

  `cupsd -f`

- [l]aunch `cupsd` on-demand (commonly used by `launchd` or `systemd`):

  `cupsd -l`

- Start `cupsd` using the specified `cupsd.conf` [c]onfiguration file:

  `cupsd -c {{path/to/cupsd.conf}}`

- Start `cupsd` using the specified `cups-files.conf` configuration file:

  `cupsd -s {{path/to/cups-files.conf}}`

- [t]est the `cupsd.conf` [c]onfiguration file for errors:

  `cupsd -t -c {{path/to/cupsd.conf}}`

- [t]est the `cups-files.conf` configuration file for errors:

  `cupsd -t -s {{path/to/cups-files.conf}}`

- Display [h]elp:

  `cupsd -h`

cupsdisable

Stop printers and classes.

Note: Destination is referred as a printer or a class of printers.

See also: **`cupsenable`**, **`cupsaccept`**, **`cupsreject`**, **`lpstat`**.

More information: https://openprinting.github.io/cups/doc/man-cupsenable.html.

- Stop one or more destination(s):

```
cupsdisable {{destination1 destination2 ...}}
```

- Cancel all jobs of the specified destination(s):

```
cupsdisable -c {{destination1 destination2 ...}}
```

cupsenable

Start printers and classes.

Note: Destination is referred as a printer or a class of printers.

See also: **`cupsdisable`**, **`cupsaccept`**, **`cupsreject`**, **`lpstat`**.

More information: https://www.cups.org/doc/man-cupsenable.html.

- Start one or more destination(s):

```
cupsenable {{destination1 destination2 ...}}
```

- Resume printing of pending jobs of a destination (use after `cupsdisable` with `--hold`):

```
cupsenable --release {{destination}}
```

- Cancel all jobs of the specified destination(s):

```
cupsenable -c {{destination1 destination2 ...}}
```

cupsreject

Reject jobs sent to printers.

Note: Destination is referred as a printer or a class of printers.

See also: **`cupsaccept`**, **`cupsenable`**, **`cupsdisable`**, **`lpstat`**.

More information: https://www.cups.org/doc/man-cupsaccept.html.

- Reject print jobs to the specified destinations:

```
cupsreject {{destination1 destination2 ...}}
```

- Specify a different server:

```
cupsreject -h {{server}} {{destination1 destination2 ...}}
```

- Specify a reason string ("Reason Unknown" by default):

```
cupsreject -r {{reason}} {{destination1 destination2 ...}}
```

cupstestppd

Test conformance of PPD files to the version 4.3 of the specification.

Error codes (1, 2, 3 and 4, respectively): bad CLI arguments, unable to open file, unskippable format errors and non-conformance with PPD specification.

Note: This command is deprecated.

See also: **`lpadmin`**.

More information: https://openprinting.github.io/cups/doc/man-cupstestppd.html.

- Test the conformance of one or more files in quiet mode:

```
cupstestppd -q {{path/to/file1.ppd path/to/file2.ppd ...}}
```

- Get the PPD file from `stdin`, showing detailed conformance testing results:

```
cupstestppd < {{path/to/file.ppd}} -v -
```

- Test all PPD files under the current directory, printing the names of each file that does not conform:

```
find . -name \*.ppd \! -execdir cupstestppd -q '{}' \; -print
```

curl

Transfers data from or to a server.

Supports most protocols, including HTTP, HTTPS, FTP, SCP, etc.

See also: **`wget`**.

More information: https://curl.se/docs/manpage.html.

- Make an HTTP GET request and dump the contents in `stdout`:

```
curl {{https://example.com}}
```

- Make an HTTP GET request, follow any `3xx` redirects, and dump the reply headers and contents to `stdout`:

```
curl {{[-L|--location]}} {{[-D|--dump-header]}} - {{https://example.com}}
```

- Download a file, saving the output under the filename indicated by the URL:

```
curl {{[-O|--remote-name]}} {{https://example.com/filename.zip}}
```

- Send form-encoded data (POST request of type `application/x-www-form-urlencoded`). Use `--data @file_name` or `--data @'-'` to read from `stdin`:

```
curl {{[-X|--request]}} POST {{[-d|--data]}} '{{name=bob}}' {{http://example.com/form}}
```

- Send a request with an extra header, using a custom HTTP method and over a proxy (such as BurpSuite), ignoring insecure self-signed certificates:

```
curl {{[-k|--insecure]}} {{[-x|--proxy]}} {{http://127.0.0.1:8080}} {{[-H|--header]}} '{{Authorization: Bearer token}}' {{[-X|--request]}} {{GET|PUT|POST|DELETE|PATCH|...}} {{https://example.com}}
```

- Send data in JSON format, specifying the appropriate Content-Type header:

```
curl {{[-d|--data]}} '{{{"name":"bob"}}}' {{[-H|--header]}}
'{{Content-Type: application/json}}' {{http://example.com/
users/1234}}
```

- Pass client certificate and private key for the request, skipping certificate validation:

  ```
  curl {{[-E|--cert]}} {{client.pem}} --key {{key.pem}} {{[-
  k|--insecure]}} {{https://example.com}}
  ```

- Resolve a hostname to a custom IP address, with verbose output (similar to editing the `/etc/hosts` file for custom DNS resolution):

  ```
  curl {{[-v|--verbose]}} --resolve {{example.com}}:{{80}}:
  {{127.0.0.1}} {{http://example.com}}
  ```

cut

Cut out fields from **`stdin`** or files.

More information: https://www.gnu.org/software/coreutils/manual/html_node/cut-invocation.html.

- Print a specific [c]haracter/[f]ield range of each line:

```
{{command}} | cut --{{characters|fields}} {{1|1,10|1-10|1-|-10}}
```

- Print a field range of each line with a specific delimiter:

```
{{command}} | cut {{[-d|--delimiter]}} "{{delimiter}}" {{[-f|--fields]}} {{1|1,10|1-10|1-|-10}}
```

- Print a character range of each line of the specific file:

```
cut {{[-c|--characters]}} {{1}} {{path/to/file}}
```

- Print specific fields of `NUL` terminated lines (e.g. as in `find . -print0`) instead of newlines:

```
{{command}} | cut {{[-z|--zero-terminated]}} {{[-f|--fields]}} {{1}}
```

CVS

Concurrent Versions System, a revision control system.

More information: https://manned.org/cvs.

- Create a new repository (requires the `CVSROOT` environment variable to be set externally):

```
cvs -d {{path/to/repository}} init
```

- Add a project to the repository:

```
cvs import -m "{{message}}" {{project_name}} {{version}} {{vendor}}
```

- Checkout a project:

```
cvs checkout {{project_name}}
```

- Show changes made to files:

```
cvs diff {{path/to/file}}
```

- Add a file:

```
cvs add {{path/to/file}}
```

- Commit a file:

```
cvs commit -m "{{message}}" {{path/to/file}}
```

- Update the working directory from the remote repository:

```
cvs update
```

cwebp

Compress an image file to a WebP file.

More information: https://developers.google.com/speed/webp/docs/cwebp.

- Compress a WebP file with default settings (q = 75) to the [o]utput file:

```
cwebp {{path/to/image_file}} -o {{path/to/output.webp}}
```

- Compress a WebP file with the best [q]uality and largest file size:

```
cwebp {{path/to/image_file}} -o {{path/to/output.webp}} -q
{{100}}
```

- Compress a WebP file with the worst [q]uality and smallest file size:

```
cwebp {{path/to/image_file}} -o {{path/to/output.webp}} -q
{{0}}
```

- Compress a WebP file and apply resize to image:

```
cwebp {{path/to/image_file}} -o {{path/to/output.webp}} -
resize {{width}} {{height}}
```

- Compress a WebP file and drop alpha channel information:

```
cwebp {{path/to/image_file}} -o {{path/to/output.webp}} -
noalpha
```

cypher-shell

Open an interactive session and run Cypher queries against a Neo4j instance.

See also: **`neo4j-admin`**, **`mysql`**.

More information: https://neo4j.com/docs/operations-manual/current/tools/cypher-shell/.

- Connect to a local instance on the default port (`neo4j://localhost:7687`):

```
cypher-shell
```

- Connect to a remote instance:

```
cypher-shell --address neo4j://{{host}}:{{port}}
```

- Connect and supply security credentials:

```
cypher-shell --username {{username}} --password {{password}}
```

- Connect to a specific database:

```
cypher-shell --database {{database_name}}
```

- Execute Cypher statements in a file and close:

```
cypher-shell --file {{path/to/file.cypher}}
```

- Enable logging to a file:

```
cypher-shell --log {{path/to/file.log}}
```

- Display help:

```
cypher-shell --help
```

cython

A compiler that converts .pyx files into C or C++ source files.

More information: https://docs.cython.org/en/latest/.

- Compile into C code:

```
cython {{path/to/file}}
```

- Compile into C++ code:

```
cython --cplus {{path/to/file}}
```

- Specify an output file:

```
cython {{[-o|--output-file]}} {{path/to/output_file}} {{path/to/file}}
```

- Display version:

```
cython {{[-V|--version]}}
```

czkawka-cli

Command-line version of **czkawka** a multi-functional app to find duplicates, empty folders, similar images and much more.

More information: https://github.com/qarmin/czkawka.

- List duplicate or similar files in specific directories:

```
czkawka-cli {{dup|image}} --directories {{path/to/directory1 path/to/directory2 ...}}
```

- Find duplicate files in specific directories and delete them (default: NONE):

```
czkawka-cli dup --directories {{path/to/directory1 path/to/directory2 ...}} --delete-method {{AEN|AEO|ON|OO|HARD|NONE}}
```

d2

A modern diagram scripting language that turns text to diagrams.

Note: The output file supports SVG and PNG file formats.

More information: https://d2lang.com/tour/man.

- Compile and render a D2 source file into an output file:

  ```
  d2 {{path/to/input_file.d2}} {{path/to/output_file.ext}}
  ```

- Watch live changes made to a D2 source file in the default web browser:

  ```
  d2 {{[-w|--watch]}} {{path/to/input_file.d2}} {{path/to/output_file.ext}}
  ```

- Format a D2 source file:

  ```
  d2 fmt {{path/to/input_file.d2}}
  ```

- List available themes:

  ```
  d2 themes
  ```

- Use a different theme for the output file (list available themes first to get the desired `theme_id`):

  ```
  d2 {{[-t|--theme]}} {{theme_id}} {{path/to/input_file.d2}} {{path/to/output_file.ext}}
  ```

- Make rendered diagrams look like hand sketches:

  ```
  d2 {{[-s|--sketch]}} true {{path/to/input_file.d2}} {{path/to/output_file.ext}}
  ```

d8

Developer shell for the V8 JavaScript engine.

More information: https://v8.dev/docs/d8.

- Start a REPL (interactive shell):

```
d8
```

- Run a JavaScript file:

```
d8 {{path/to/file.js}}
```

- Evaluate a JavaScript expression:

```
d8 -e "{{code}}"
```

dalfox

A powerful open-source XSS scanner focused on automation.

More information: https://dalfox.hahwul.com/page/usage/.

- Scan a single URL for XSS vulnerabilities:

```
dalfox url {{https://example.com}}
```

- Scan a URL using a header for authentication:

```
dalfox url {{https://example.com}} {{[-H|--header]}} '{{X-My-Header: 123}}'
```

- Scan a list of URLs from a file:

```
dalfox file {{path/to/file}}
```

- Start Dalfox as a REST API server:

```
dalfox server --host {{0.0.0.0}} --port {{8080}}
```

daps

An open source program for transforming DocBook XML into output formats such as HTML or PDF.

More information: https://opensuse.github.io/daps/doc/index.html.

- Check if a DocBook XML file is valid:

```
daps {{[-d|--docconfig]}} {{path/to/file.xml}} validate
```

- Convert a DocBook XML file into PDF:

```
daps {{[-d|--docconfig]}} {{path/to/file.xml}} pdf
```

- Convert a DocBook XML file into a single HTML file:

```
daps {{[-d|--docconfig]}} {{path/to/file.xml}} html --single
```

- Display help:

```
daps {{[-h|--help]}}
```

- Display version:

```
daps --version
```

darkhttpd

Darkhttpd web server.

More information: https://github.com/emikulic/darkhttpd#how-to-run-darkhttpd.

- Start server serving the specified document root:

```
darkhttpd {{path/to/docroot}}
```

- Start server on specified port (port 8080 by default if running as non-root user):

```
darkhttpd {{path/to/docroot}} --port {{port}}
```

- Listen only on specified IP address (by default, the server listens on all interfaces):

```
darkhttpd {{path/to/docroot}} --addr {{ip_address}}
```

dart

Manage Dart projects.

More information: https://dart.dev/tools/dart-tool.

- Initialize a new Dart project in a directory of the same name:

```
dart create {{project_name}}
```

- Run a Dart file:

```
dart run {{path/to/file.dart}}
```

- Download dependencies for the current project:

```
dart pub get
```

- Run unit tests for the current project:

```
dart test
```

- Update an outdated project's dependencies to support null-safety:

```
dart pub upgrade --null-safety
```

- Compile a Dart file to a native binary:

```
dart compile exe {{path/to/file.dart}}
```

- Apply automated fixes to the current project:

```
dart fix --apply
```

dash

Debian Almquist Shell, a modern, POSIX-compliant implementation of **sh** (not Bash-compatible).

More information: https://manned.org/dash.

- Start an interactive shell session:

```
dash
```

- Execute specific [c]ommands:

```
dash -c "{{echo 'dash is executed'}}"
```

- Execute a specific script:

```
dash {{path/to/script.sh}}
```

- Check a specific script for syntax errors:

```
dash -n {{path/to/script.sh}}
```

- Execute a specific script while printing each command before executing it:

```
dash -x {{path/to/script.sh}}
```

- Execute a specific script and stop at the first [e]rror:

```
dash -e {{path/to/script.sh}}
```

- Execute specific commands from `stdin`:

```
{{echo "echo 'dash is executed'"}} | dash
```

datashader_cli

Quick visualization of large datasets using CLI based on datashader.

More information: https://github.com/wybert/datashader-cli.

- Create a shaded scatter plot of points and save it to a PNG file and set the background color:

```
datashader_cli points {{path/to/input.parquet}} --x
{{pickup_x}} --y {{pickup_y}} {{path/to/output.png}} --
background {{black|white|#rrggbb}}
```

- Visualize the geospatial data (supports Geoparquet, shapefile, geojson, geopackage, etc.):

```
datashader_cli points {{path/to/input_data.geo.parquet}}
{{path/to/output_data.png}} --geo true
```

- Use matplotlib to render the image:

```
datashader_cli points {{path/to/input_data.geo.parquet}}
{{path/to/output_data.png}} --geo {{true}} --matplotlib true
```

date

Set or display the system date.

More information: https://www.gnu.org/software/coreutils/manual/html_node/date-invocation.html.

- Display the current date using the default locale's format:

```
date +%c
```

- Display the current date in UTC, using the ISO 8601 format:

```
date {{[-u|--utc]}} +%Y-%m-%dT%H:%M:%S%Z
```

- Display the current date as a Unix timestamp (seconds since the Unix epoch):

```
date +%s
```

- Convert a date specified as a Unix timestamp to the default format:

```
date {{[-d|--date]}} @{{1473305798}}
```

- Convert a given date to the Unix timestamp format:

```
date {{[-d|--date]}} "{{2018-09-01 00:00}}" +%s {{[-u|--utc]}}
```

- Display the current date using the RFC-3339 format (`YYYY-MM-DD hh:mm:ss TZ`):

```
date --rfc-3339 s
```

- Set the current date using the format `MMDDhhmmYYYY.ss` (`YYYY` and `.ss` are optional):

```
date {{093023592021.59}}
```

- Display the current ISO week number:

```
date +%V
```

dbeaver

A GUI SQL client supporting many databases.

More information: https://dbeaver.com/docs/dbeaver/Command-Line/.

- Open DBeaver:

```
dbeaver
```

- Open DBeaver connecting to a specific database:

```
dbeaver {{[-con|--connect]}} {{database}}
```

- Force the creation of a new instance:

```
dbeaver --newInstance
```

- Stop the running instance:

```
dbeaver --quit
```

- Close all tabs within DBeaver:

```
dbeaver --closeTabs
```

- Bring DBeaver to the top of applications:

```
dbeaver --bringToFront
```

- Display help:

```
dbeaver --help
```

dbt

A tool to model transformations in data warehouses.

More information: https://github.com/dbt-labs/dbt-core.

- Debug the dbt project and the connection to the database:

  ```
  dbt debug
  ```

- Run all models of the project:

  ```
  dbt run
  ```

- Run all tests of `example_model`:

  ```
  dbt test --select example_model
  ```

- Build (load seeds, run models, snapshots, and tests associated with) `example_model` and its downstream dependents:

  ```
  dbt build --select example_model+
  ```

- Build all models, except the ones with the tag `not_now`:

  ```
  dbt build --exclude "tag:not_now"
  ```

- Build all models with tags `one` and `two`:

  ```
  dbt build --select "tag:one,tag:two"
  ```

- Build all models with tags `one` or `two`:

  ```
  dbt build --select "tag:one tag:two"
  ```

dbx

Interact with the Databricks platform.

Note: This tool has been retired and it is recommended to use Databricks Asset Bundles instead.

More information: https://dbx.readthedocs.io/en/latest/reference/cli/#dbx.

- Create a new `dbx` project in the current working directory:

  ```
  dbx configure --profile {{DEFAULT}}
  ```

- Sync local files from the specified path to DBFS and watch for changes:

  ```
  dbx sync dbfs --source {{path/to/directory}} --dest {{path/to/remote_directory}}
  ```

- Deploy the specified workflow to artifact storage:

  ```
  dbx deploy {{workflow_name}}
  ```

- Launch the specified workflow after deploying it:

  ```
  dbx launch {{workflow_name}}
  ```

dc

An arbitrary precision calculator. Uses reverse polish notation (RPN).

See also: **`bc`**, **`qalc`**.

More information: https://www.gnu.org/software/bc/manual/dc-1.05/html_mono/dc.html.

- Start an interactive session:

```
dc
```

- Execute a script:

```
dc {{path/to/script.dc}}
```

- Calculate an expression with the specified scale:

```
dc {{[-e|--expression]}} '{{10}} k {{5 3 /}} p'
```

- Calculate 4 times 5 (4 5 *), subtract 17 (17 -), and [p]rint the output:

```
dc {{[-e|--expression]}} '4 5 * 17 - p'
```

- Specify the number of decimal places to 7 (7 k), calculate 5 divided by -3 (5 _3 /) and [p]rint:

```
dc {{[-e|--expression]}} '7 k 5 _3 / p'
```

- Calculate the golden ratio, phi: set number of decimal places to 100 (100 k), square root of 5 (5 v) plus 1 (1 +), divided by 2 (2 /), and [p]rint result:

```
dc {{[-e|--expression]}} '100 k 5 v 1 + 2 / p'
```

dcfldd

Enhanced version of dd for forensics and security.

More information: https://manned.org/dcfldd.

- Copy a disk to a raw image file and hash the image using SHA256:

```
dcfldd if={{/dev/disk_device}} of={{file.img}} hash=sha256
hashlog={{file.hash}}
```

- Copy a disk to a raw image file, hashing each 1 GB chunk:

```
dcfldd if={{/dev/disk_device}} of={{file.img}} hash={{sha512|
sha384|sha256|sha1|md5}} hashlog={{file.hash}}
hashwindow={{1G}}
```

dcg

Drupal code generator.

More information: https://github.com/Chi-teck/drupal-code-generator.

- Start a wizard to choose what kind of code (e.g. module, service, form, etc.) to generate:

  ```
  dcg
  ```

- Directly specify the kind of code to generate:

  ```
  dcg {{service|plugin|theme|module|form}}
  ```

- Generate the code in a specific directory:

  ```
  dcg --directory {{path/to/directory}}
  ```

dcgmi

Manage and monitor NVIDIA Data Center GPUs.

More information: https://developer.nvidia.com/dcgm.

- Display information on all available GPUs and processes using them:

```
dcgmi discovery {{[-l|--list]}}
```

- List created groups:

```
dcgmi group {{[-l|--list]}}
```

- Display current usage statistics for device 0:

```
dcgmi dmon {{[-e|--field-id]}}{{1001,1002,1003,1004,1005}}
```

- Display help:

```
dcgmi {{[-h|--help]}}
```

- Display help for a subcommand:

```
dcgmi {{subcommand}} {{[-h|--help]}}
```

dcode

Recursively detect and decode strings, supporting hex, decimal, binary, base64, URL, FromChar encodings, Caesar ciphers, and MD5, SHA1, and SHA2 hashes.

Warning: uses 3rd-party web services for MD5, SHA1 and SHA2 hash lookups. For sensitive data, use **`-s`** to avoid these services.

More information: https://github.com/s0md3v/Decodify.

- Recursively detect and decode a string:

```
dcode "{{NjM3YTQyNzQ1YTQ0NGUzMg==}}"
```

- Rotate a string by the specified offset:

```
dcode -rot {{11}} "{{spwwz hzcwo}}"
```

- Rotate a string by all 26 possible offsets:

```
dcode -rot {{all}} "{{bpgkta xh qtiitg iwpc sr}}"
```

- Reverse a string:

```
dcode -rev "{{hello world}}"
```

dd

Convert and copy a file.

More information: https://manned.org/dd.1p.

- Make a bootable USB drive from an isohybrid file (such as `archlinux-xxx.iso`) and show the progress:

```
dd if={{path/to/file.iso}} of={{/dev/usb_drive}} status=progress
```

- Clone a drive to another drive with 4 MiB block size and flush writes before the command terminates:

```
dd bs=4194304 conv=fsync if={{/dev/source_drive}} of={{/dev/dest_drive}}
```

- Generate a file with a specific number of random bytes by using kernel random driver:

```
dd bs={{100}} count={{1}} if=/dev/urandom of={{path/to/random_file}}
```

- Benchmark the sequential write performance of a disk:

```
dd bs={{1024}} count={{1000000}} if=/dev/zero of={{path/to/file_1GB}}
```

- Create a system backup, save it into an IMG file (can be restored later by swapping `if` and `of`), and show the progress:

```
dd if={{/dev/drive_device}} of={{path/to/file.img}} status=progress
```

ddev

Container based local development tool for PHP environments.

More information: https://ddev.readthedocs.io/en/stable/users/usage/cli/.

- Start up a project:

```
ddev start
```

- Configure a project's type and docroot:

```
ddev config
```

- Follow the log trail:

```
ddev logs {{[-f|--follow]}}
```

- Run composer within the container:

```
ddev composer
```

- Install a specific Node.js version:

```
ddev nvm install {{version}}
```

- Export a database:

```
ddev export-db {{[-f|--file]}} {{/tmp/db.sql.gz}}
```

- Run a specific command within a container:

```
ddev exec {{echo 1}}
```

ddgr

Search DuckDuckGo (HTML version) from the terminal.

More information: https://github.com/jarun/ddgr.

- Start in interactive mode:

```
ddgr
```

- Search DuckDuckGo for a keyword:

```
ddgr {{keyword}}
```

- Limit the number of search results to n:

```
ddgr {{[-n|--num]}} {{n}} {{keyword}}
```

- Display the complete URL in search results:

```
ddgr {{[-x|--expand]}} {{keyword}}
```

- Search DuckDuckGo for a keyword and open the first result in the browser:

```
ddgr !w {{keyword}}
```

- Perform a website-specific search:

```
ddgr {{[-w|--site]}} {{site}} {{keyword}}
```

- Search for a specific file type:

```
ddgr {{keyword}} filetype:{{filetype}}
```

- Display help in interactive mode:

```
<?>
```

decaffeinate

Move your CoffeeScript source to modern JavaScript.

More information: https://decaffeinate-project.org.

- Convert a CoffeeScript file to JavaScript:

  ```
  decaffeinate {{path/to/file.coffee}}
  ```

- Convert a CoffeeScript v2 file to JavaScript:

  ```
  decaffeinate --use-cs2 {{path/to/file.coffee}}
  ```

- Convert require and `module.exports` to import and export:

  ```
  decaffeinate --use-js-modules {{path/to/file.coffee}}
  ```

- Convert a CoffeeScript, allowing named exports:

  ```
  decaffeinate --loose-js-modules {{path/to/file.coffee}}
  ```

declare

Declare variables and give them attributes.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-declare.

- Declare a string variable with the specified value:

```
declare {{variable}}="{{value}}"
```

- Declare an integer variable with the specified value:

```
declare -i {{variable}}="{{value}}"
```

- Declare an array variable with the specified value:

```
declare -a {{variable}}=({{item_a item_b item_c}})
```

- Declare an associative array variable with the specified value:

```
declare -A {{variable}}=({{[key_a]=item_a [key_b]=item_b [key_c]=item_c}})
```

- Declare a readonly string variable with the specified value:

```
declare -r {{variable}}="{{value}}"
```

- Declare a global variable within a function with the specified value:

```
declare -g {{variable}}="{{value}}"
```

- Print a function definition:

```
declare -f {{function_name}}
```

- Print a variable definition:

```
declare -p {{variable_name}}
```

deemix

A barebone deezer downloader library built from the ashes of Deezloader Remix.

It can be used as a standalone CLI app or implemented in a UI using the API.

More information: https://gitlab.com/RemixDev/deemix-py.

- Download a track or playlist:

  ```
  deemix {{https://www.deezer.com/us/track/00000000}}
  ```

- Download track/playlist at a specific bitrate:

  ```
  deemix --bitrate {{FLAC|MP3}} {{url}}
  ```

- Download to a specific path:

  ```
  deemix --bitrate {{bitrate}} --path {{path}} {{url}}
  ```

- Create a portable deemix configuration file in the current directory:

  ```
  deemix --portable --bitrate {{bitrate}} --path {{path}}
  {{url}}
  ```

delta

A viewer for Git and diff output.

More information: https://dandavison.github.io/delta/full---help-output.html.

- Compare files or directories:

```
delta {{path/to/old_file_or_directory}} {{path/to/
new_file_or_directory}}
```

- Compare files or directories, showing the line numbers:

```
delta {{[-n|--line-numbers]}} {{path/to/
old_file_or_directory}} {{path/to/new_file_or_directory}}
```

- Compare files or directories, showing the differences side by side:

```
delta {{[-s|--side-by-side]}} {{path/to/
old_file_or_directory}} {{path/to/new_file_or_directory}}
```

- Compare files or directories, ignoring any Git configuration settings:

```
delta --no-gitconfig {{path/to/old_file_or_directory}}
{{path/to/new_file_or_directory}}
```

- Compare, rendering commit hashes, file names, and line numbers as hyperlinks, according to the hyperlink spec for terminal emulators:

```
delta --hyperlinks {{path/to/old_file_or_directory}} {{path/
to/new_file_or_directory}}
```

- Display the current settings:

```
delta --show-config
```

- Display supported languages and associated file extensions:

```
delta --list-languages
```

deluge-console

An interactive interface for the Deluge BitTorrent client.

More information: https://deluge-torrent.org/userguide/thinclient/.

- Start the interactive console interface:

```
deluge-console
```

- Connect to a Deluge daemon instance:

```
connect {{hostname}}:{{port}}
```

- Add a torrent to the daemon:

```
add {{url|magnet|path/to/file}}
```

- Display information about all torrents:

```
info
```

- Display information about a specific torrent:

```
info {{torrent_id}}
```

- Pause a torrent:

```
pause {{torrent_id}}
```

- Resume a torrent:

```
resume {{torrent_id}}
```

- Remove a torrent from the daemon:

```
rm {{torrent_id}}
```

deluge

A BitTorrent client.

More information: https://manned.org/deluge.

- Download a torrent:

```
deluge {{url|magnet|path/to/file}}
```

- Download a torrent using a specific configuration file:

```
deluge {{[-c|--config]}} {{path/to/configuration_file}}
{{url|magnet|path/to/file}}
```

- Download a torrent and launch the specified user interface:

```
deluge -u {{gtk|web|console}} {{url|magnet|path/to/file}}
```

- Download a torrent and output the log to a file:

```
deluge {{[-l|--logfile]}} {{path/to/log_file}} {{url|magnet|
path/to/file}}
```

deluged

A daemon process for the Deluge BitTorrent client.

More information: https://manned.org/deluged.

- Start the Deluge daemon:

```
deluged
```

- Start the Deluge daemon on a specific port:

```
deluged {{[-p|--port]}} {{port}}
```

- Start the Deluge daemon using a specific configuration file:

```
deluged {{[-c|--config]}} {{path/to/configuration_file}}
```

- Start the Deluge daemon and output the log to a file:

```
deluged {{[-l|--logfile]}} {{path/to/log_file}}
```

deno

A secure runtime for JavaScript, TypeScript, and WebAssembly.

Includes dependency management using **`npm`** or **`jsr`**, and tooling like bench, bundle, doc, and coverage.

More information: https://docs.deno.com/runtime/reference/cli.

- Start a REPL (interactive shell, also known as Read-Eval-Print Loop):

  ```
  deno
  ```

- Start a new project named sample and test it:

  ```
  deno init sample && cd sample && deno test
  ```

- Run a file securely. It will ask (if needed) to allow net, read, etc:

  ```
  deno run {{path/to/file.ts}}
  ```

- Run a file with explicit permissions or allow all (only if you trust the source):

  ```
  deno run {{[--allow-env|--allow-net|--allow-write|--allow-all]}} {{jsr:@deno/deployctl}}
  ```

- List and run tasks from `deno.json` or scripts from `package.json`:

  ```
  deno task
  ```

- Install dependencies listed in `deno.json` or `package.json` (also lock files):

  ```
  deno install
  ```

- Check types, format, and lint (fix if possible):

  ```
  deno check && deno fmt && deno lint --fix
  ```

- Compile the script, imported dependencies, and runtime into a self contained executable:

  ```
  deno compile {{path/to/file.ts}}
  ```

dep

Deploy PHP applications.

Note: The Go command **dep** with the same name is deprecated and archived.

More information: https://deployer.org.

- Interactively initialize deployer in the local path (use a framework template with `--template=template`):

  ```
  dep init
  ```

- Deploy an application to a remote host:

  ```
  dep deploy {{hostname}}
  ```

- Rollback to the previous working release:

  ```
  dep rollback
  ```

- Connect to a remote host via SSH:

  ```
  dep ssh {{hostname}}
  ```

- List commands:

  ```
  dep list
  ```

- Run any arbitrary command on the remote hosts:

  ```
  dep run "{{command}}"
  ```

- Display help for a command:

  ```
  dep help {{command}}
  ```

detox

Renames files to make them easier to work with.

It removes spaces and other such annoyances like duplicate underline characters.

More information: https://manned.org/detox.

- Remove spaces and other undesirable characters from a file's name:

```
detox {{path/to/file}}
```

- Show how detox would rename all the files in a directory tree:

```
detox {{[-n|--dry-run]}} -r {{path/to/directory}}
```

- Remove spaces and other undesirable characters from all files in a directory tree:

```
detox -r {{path/to/directory}}
```

devcontainer

Use a Docker container as a development environment.

More information: https://containers.dev/implementors/reference/.

- Create and run a Dev Container:

  ```
  devcontainer up
  ```

- Apply a Dev Container Template to a workspace:

  ```
  devcontainer templates apply {{[-t|--template-id]}} {{template_id}} {{[-a|--template-args]}} {{template_args}} {{[-w|--workspace-folder]}} {{path/to/workspace}}
  ```

- Execute a command on a running Dev Container in the current workspace:

  ```
  devcontainer exec {{command}}
  ```

- Build a Dev Container image from `devcontainer.json`:

  ```
  devcontainer build {{path/to/workspace}}
  ```

- Open a Dev Container in VS Code (the path is optional):

  ```
  devcontainer open {{path/to/workspace}}
  ```

- Read and print the configuration of a Dev Container from `devcontainer.json`:

  ```
  devcontainer read-configuration
  ```

devenv

Fast, Declarative, Reproducible and Composable Developer Environments using Nix.

More information: https://devenv.sh/getting-started/#commands.

- Initialize the environment:

  ```
  devenv init
  ```

- Enter the Development Environment with relaxed hermeticity (state of being airtight):

  ```
  devenv shell --impure
  ```

- Get detailed information about the current environment:

  ```
  devenv info --verbose
  ```

- Start processes with `devenv`:

  ```
  devenv up --config {{path/to/file}}
  ```

- Clean the environment variables and re-enter the shell in offline mode:

  ```
  devenv --clean --offline
  ```

- Delete the previous shell generations:

  ```
  devenv gc
  ```

devpod

Launch reproducible development environments using Docker, Kubernetes, or SSH.

More information: https://devpod.sh/docs/quickstart/devpod-cli/.

- Add a provider such as Docker or Kubernetes:

  ```
  devpod provider add {{provider_name}}
  ```

- List all available providers:

  ```
  devpod provider list-available
  ```

- Start a workspace from a GitHub repository with a specific IDE:

  ```
  devpod up {{github.com/user/repo}} {{[-i|--ide]}} {{vscode}}
  ```

- Start a workspace from a local directory:

  ```
  devpod up {{path/to/project}}
  ```

- Recreate an existing workspace:

  ```
  devpod up {{workspace_name}} {{[-r|--recreate]}}
  ```

- Reset a workspace to a clean state:

  ```
  devpod up {{workspace_name}} {{[-x|--reset]}}
  ```

- Add a custom provider from a GitHub repository:

  ```
  devpod provider add {{org/provider-repo}}
  ```

devspace

Develop, deploy, and debug applications inside Kubernetes.

More information: https://devspace.sh/docs/cli.

- Initialize a new DevSpace project in the current directory:

  ```
  devspace init
  ```

- Start development mode with port forwarding, file synchronization, and terminal access:

  ```
  devspace dev
  ```

- Start development mode in a specific namespace:

  ```
  devspace dev {{[-n|--namespace]}} {{namespace}}
  ```

- Deploy the project to Kubernetes:

  ```
  devspace deploy
  ```

- Deploy the project with a specific profile:

  ```
  devspace deploy {{[-p|--profile]}} {{profile-name}}
  ```

- Build all defined images:

  ```
  devspace build
  ```

- Follow logs from a pod:

  ```
  devspace logs {{[-f|--follow]}}
  ```

- Open the DevSpace UI in the browser:

  ```
  devspace ui
  ```

dexdump

Display information about Android DEX files.

More information: https://manned.org/man/debian-stretch/dexdump.

- Extract classes and methods from an APK file:

```
dexdump {{path/to/file.apk}}
```

- Display header information of DEX files contained in an APK file:

```
dexdump -f {{path/to/file.apk}}
```

- Display the dis-assembled output of executable sections:

```
dexdump -d {{path/to/file.apk}}
```

- Output results to a file:

```
dexdump -o {{path/to/file}} {{path/to/file.apk}}
```

dexter

Authenticate the **`kubectl`** users with OpenId Connect.

More information: https://github.com/gini/dexter.

- Create and authenticate a user with Google OIDC:

```
dexter auth {{[-i|--client-id]}} {{client_id}} {{[-s|--
client-secret]}} {{client_secret}}
```

- Override the default kube configuration file location:

```
dexter auth {{[-i|--client-id]}} {{client_id}} {{[-s|--
client-secret]}} {{client_secret}} {{[-k|--kube-config]}}
{{sample/config}}
```

df

Display an overview of the filesystem disk space usage.

More information: https://manned.org/df.1posix.

- Display all filesystems and their disk usage (using 512-byte units):

```
df
```

- Display the filesystem containing the specified file or directory:

```
df {{path/to/file_or_directory}}
```

- Use [k]ibibyte (1024 byte) units when showing size figures:

```
df -k
```

- Display information in a portable way:

```
df -P
```

dfc

Get an overview of the filesystem disk space usage with colors and graphs.

More information: https://github.com/Rolinh/dfc.

- Display filesystems and their disk usage in human-readable form with colors and graphs:

```
dfc
```

- Display all filesystems including pseudo, duplicate and inaccessible filesystems:

```
dfc -a
```

- Display filesystems without color:

```
dfc -c never
```

- Display filesystems containing "ext" in the filesystem type:

```
dfc -t ext
```

dhclient

DHCP client.

More information: https://manned.org/dhclient.

- Get an IP address for the `eth0` interface:

```
sudo dhclient {{eth0}}
```

- Release an IP address for the `eth0` interface:

```
sudo dhclient -r {{eth0}}
```

dhcpig

Initiates an advanced DHCP exhaustion attack and stress test.

DHCPig needs to be run with root privileges.

More information: https://github.com/kamorin/DHCPig.

- Exhaust all of the available DHCP addresses using the specified interface:

```
sudo ./pig.py {{eth0}}
```

- Exhaust IPv6 addresses using eth1 interface:

```
sudo ./pig.py {{[-6|--ipv6]}} {{eth1}}
```

- Send fuzzed/malformed data packets using the interface:

```
sudo ./pig.py {{[-f|--fuzz]}} {{eth1}}
```

- Enable color output:

```
sudo ./pig.py {{[-c|--color]}} {{eth1}}
```

- Enable minimal verbosity and color output:

```
sudo ./pig.py {{[-c|--color]}} {{[-v|--verbosity]}} 1
{{eth1}}
```

- Use a debug verbosity of 100 and scan network of neighboring devices using ARP packets:

```
sudo ./pig.py {{[-c|--color]}} {{[-v|--verbosity]}} 100 {{[-
n|--neighbors-scan-arp]}} {{eth1}}
```

- Enable printing lease information, attempt to scan and release all neighbor IP addresses:

```
sudo ./pig.py {{[-n|--neighbors-scan-arp]}} {{[-r|--
neighbors-attack-release]}} {{[-o|--show-options]}} {{eth1}}
```

dhcpwn

Test DHCP IP exhaustion attacks and sniff local DHCP traffic.

More information: https://github.com/mschwager/dhcpwn.

- Flood the network with IP requests:

```
dhcpwn {{[-i|--interface]}} {{network_interface}} flood {{[-c|--count]}} {{number_of_requests}}
```

- Sniff local DHCP traffic:

```
dhcpwn {{[-i|--interface]}} {{network_interface}} sniff
```

dict

Command line dictionary using the DICT protocol.

More information: https://github.com/cheusov/dictd.

- List available databases:

```
dict {{[-D|--dbs]}}
```

- Get information about a database:

```
dict {{[-i|--info]}} {{database_name}}
```

- Look up a word in a specific database:

```
dict {{[-d|--database]}} {{database_name}} {{word}}
```

- Look up a word in all available databases:

```
dict {{word}}
```

- Show information about the DICT server:

```
dict {{[-I|--serverinfo]}}
```

diff-pdf

Compare two PDFs.

More information: https://github.com/vslavik/diff-pdf.

- Compare PDFs, indicating changes using return codes (`0` = no difference, `1` = PDFs differ):

```
diff-pdf {{path/to/a.pdf}} {{path/to/b.pdf}}
```

- Compare PDFs, outputting a PDF with visually highlighted differences:

```
diff-pdf --output-diff={{path/to/diff.pdf}} {{path/to/a.pdf}} {{path/to/b.pdf}}
```

- Compare PDFs, viewing differences in a simple GUI:

```
diff-pdf --view {{path/to/a.pdf}} {{path/to/b.pdf}}
```

diff

Compare files and directories.

More information: https://manned.org/diff.

- Compare files (lists changes to turn `old_file` into `new_file`):

```
diff {{old_file}} {{new_file}}
```

- Compare files, ignoring white spaces:

```
diff {{[-w|--ignore-all-space]}} {{old_file}} {{new_file}}
```

- Compare files, showing the differences side by side:

```
diff {{[-y|--side-by-side]}} {{old_file}} {{new_file}}
```

- Compare files, showing the differences in unified format (as used by `git diff`):

```
diff {{[-u|--unified]}} {{old_file}} {{new_file}}
```

- Compare directories recursively (shows names for differing files/directories as well as changes made to files):

```
diff {{[-r|--recursive]}} {{old_directory}} {{new_directory}}
```

- Compare directories, only showing the names of files that differ:

```
diff {{[-r|--recursive]}} {{[-q|--brief]}} {{old_directory}}
{{new_directory}}
```

- Create a patch file for Git from the differences of two text files, treating nonexistent files as empty:

```
diff {{[-a|--text]}} {{[-u|--unified]}} {{[-N|--new-file]}}
{{old_file}} {{new_file}} > {{diff.patch}}
```

- Compare files, showing output in color and try hard to find smaller set of changes:

```
diff {{[-d|--minimal]}} --color=always {{old_file}}
{{new_file}}
```

diffimg

Generate an image where each pixel is the difference between corresponding pixels in the 2 source images.

More information: https://graphviz.org/pdf/diffimg.1.pdf.

- Compare two images and output the difference image to the specified file:

```
diffimg {{path/to/image1}} {{path/to/image2}} {{path/to/output_image}}
```

diffoscope

Compare files, archives, and directories.

More information: https://diffoscope.org.

- Compare two files:

```
diffoscope {{path/to/file1}} {{path/to/file2}}
```

- Compare two files without displaying a progress bar:

```
diffoscope --no-progress {{path/to/file1}} {{path/to/file2}}
```

- Compare two files and write an HTML-report to a file (use `-` for `stdout`):

```
diffoscope --html {{path/to/outfile|-}} {{path/to/file1}}
{{path/to/file2}}
```

- Compare two directories excluding files with a name matching a specified pattern:

```
diffoscope --exclude {{pattern}} {{path/to/directory1}}
{{path/to/directory2}}
```

- Compare two directories and control whether directory metadata is considered:

```
diffoscope --exclude-directory-metadata {{auto|yes|no|
recursive}} {{path/to/directory1}} {{path/to/directory2}}
```

diffstat

Create a histogram from the output of the **`diff`** command.

More information: https://manned.org/diffstat.

- Display changes in a histogram:

```
diff {{path/to/file1}} {{path/to/file2}} | diffstat
```

- Display inserted, deleted and modified changes as a table:

```
diff {{path/to/file1}} {{path/to/file2}} | diffstat -t
```

difft

Compare files or directories based on the syntax of the programming language.

See also: **`delta`**, **`diff`**.

More information: https://difftastic.wilfred.me.uk/introduction.html.

- Compare two files or directories:

```
difft {{path/to/file_or_directory1}} {{path/to/
file_or_directory2}}
```

- Only report the presence of differences between the files:

```
difft --check-only {{path/to/file1}} {{path/to/file2}}
```

- Specify the display mode (default is `side-by-side`):

```
difft --display {{side-by-side|side-by-side-show-both|inline|
json}} {{path/to/file1}} {{path/to/file2}}
```

- Ignore comments when comparing:

```
difft --ignore-comments {{path/to/file1}} {{path/to/file2}}
```

- Enable/Disable syntax highlighting of source code (default is `on`):

```
difft --syntax-highlight {{on|off}} {{path/to/file1}} {{path/
to/file2}}
```

- Do not output anything at all if there are no differences between files:

```
difft --skip-unchanged {{path/to/file_or_directory1}} {{path/
to/file_or_directory2}}
```

- Print all programming languages supported by the tool, along with their extensions:

```
difft --list-languages
```

dig

DNS lookup utility.

More information: https://manned.org/dig.

- Lookup the IP(s) associated with a hostname (A records):

```
dig +short {{example.com}}
```

- Get a detailed answer for a given domain (A records):

```
dig +noall +answer {{example.com}}
```

- Query a specific DNS record type associated with a given domain name:

```
dig +short {{example.com}} {{A|MX|TXT|CNAME|NS}}
```

- Specify an alternate DNS server to query and optionally use DNS over TLS (DoT):

```
dig {{+tls}} @{{1.1.1.1|8.8.8.8|9.9.9.9|...}} {{example.com}}
```

- Perform a reverse DNS lookup on an IP address (PTR record):

```
dig -x {{8.8.8.8}}
```

- Find authoritative name servers for the zone and display SOA records:

```
dig +nssearch {{example.com}}
```

- Perform iterative queries and display the entire trace path to resolve a domain name:

```
dig +trace {{example.com}}
```

- Query a DNS server over a non-standard [p]ort using the TCP protocol:

```
dig +tcp -p {{port}} @{{dns_server_ip}} {{example.com}}
```

dijkstra

Compute shortest-path distances from a single source node in a graph.

More information: https://graphviz.org/pdf/dijkstra.1.pdf.

- Compute distances from a given source node in a graph file:

```
dijkstra {{source_node_file}}
```

- Treat the graph as [d]irected when computing distances:

```
dijkstra -d {{source_node_file}}
```

- Record the [p]revious closest node for each node on the shortest path:

```
dijkstra -p {{source_node_file}}
```

- [a]ssign large distance values to unreachable nodes:

```
dijkstra -a {{source_node_file}}
```

dillo

A lightweight web browser intended for slow computers.

More information: https://manned.org/dillo.

- Launch Dillo:

```
dillo
```

- Launch Dillo with a specific window size and screen location:

```
dillo {{[-g|--geometry]}} {{width}}x{{height}}+{{x_position}}
+{{y_position}}
```

- Launch Dillo and open a specific URL:

```
dillo {{duckduckgo.com}}
```

- Launch Dillo and open a file or directory:

```
dillo {{path/to/file_or_directory}}
```

- Launch Dillo in full-screen mode:

```
dillo {{[-f|--fullwindow]}}
```

- Display version:

```
dillo {{[-v|--version]}}
```

- Display help:

```
dillo {{[-h|--help]}}
```

dircolors

Output commands to set the LS_COLOR environment variable and style **`ls`**, **`dir`**, etc.

More information: https://www.gnu.org/software/coreutils/manual/html_node/dircolors-invocation.html.

- Output commands to set LS_COLOR using default colors:

```
dircolors
```

- Display each filetype with the color they would appear in `ls`:

```
dircolors --print-ls-colors
```

- Output commands to set LS_COLOR using colors from a file:

```
dircolors {{path/to/file}}
```

- Output commands for Bourne shell:

```
dircolors {{[-b|--bourne-shell]}}
```

- Output commands for C shell:

```
dircolors {{[-c|--c-shell]}}
```

- View the default colors for file types and extensions:

```
dircolors {{[-p|--print-database]}}
```

direnv

Shell extension to load and unload environment variables depending on the current directory.

More information: https://github.com/direnv/direnv.

- Grant direnv permission to load the `.envrc` present in the current directory:

```
direnv allow {{.}}
```

- Revoke the authorization to load the `.envrc` present in the current directory:

```
direnv deny {{.}}
```

- Edit the `.envrc` file in the default text editor and reload the environment on exit:

```
direnv edit {{.}}
```

- Trigger a reload of the environment:

```
direnv reload
```

- Print some debug status information:

```
direnv status
```

dirname

Calculates the parent directory of a file or directory path.

More information: https://www.gnu.org/software/coreutils/manual/html_node/dirname-invocation.html.

- Calculate the parent directory of a given path:

```
dirname {{path/to/file_or_directory}}
```

- Calculate the parent directory of multiple paths:

```
dirname {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Delimit output with a NUL character instead of a newline (useful when combining with `xargs`):

```
dirname {{[-z|--zero]}} {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

dirs

Display or manipulate the directory stack.

The directory stack is a list of recently visited directories that can be manipulated with the **pushd** and **popd** commands.

See also: **pushd**, **popd**.

More information: https://www.gnu.org/software/bash/manual/bash.html#Directory-Stack-Builtins.

- Display the directory stack with a space between each entry:

```
dirs
```

- Display the directory stack with one entry per line:

```
dirs -p
```

- Display a numbered list of entries in the directory stack:

```
dirs -v
```

- Display the directory stack without the tilde-prefix (~):

```
dirs -l
```

- Display only the nth entry in the directory stack, starting at 0 (Bash only):

```
dirs +{{n}}
```

- Display only the nth entry in the directory stack from the last, starting at 0 (Bash only):

```
dirs -{{n}}
```

- Clear the directory stack:

```
dirs -c
```

dirsearch

Web path scanner.

More information: https://github.com/maurosoria/dirsearch.

- Scan a web server for common paths with common extensions:

```
dirsearch {{[-u|--url]}} {{url}} --extensions-list
```

- Scan a list of web servers for common paths with given file extensions:

```
dirsearch {{[-l|--url-list]}} {{path/to/url-list.txt}} {{[-e|--extensions]}} {{php,jsp,aspx,...}}
```

- Scan a web server for user-defined paths with common extensions:

```
dirsearch {{[-u|--url]}} {{url}} --extensions-list {{[-w|--wordlists]}} {{path/to/url-paths1.txt,path/to/url-paths2.txt,...}}
```

- Scan a web server using a cookie:

```
dirsearch {{[-u|--url]}} {{url}} {{[-e|--extensions]}} {{php}} --cookie {{cookie}}
```

- Scan a web server using the `HEAD` HTTP method:

```
dirsearch {{[-u|--url]}} {{url}} {{[-e|--extensions]}} {{php}} {{[-m|--http-method]}} {{HEAD}}
```

- Scan a web server, saving the results to a `.json` file:

```
dirsearch {{[-u|--url]}} {{url}} {{[-e|--extensions]}} {{php}} --json-report {{path/to/report.json}}
```

diskonaut

Terminal disk space navigator, written in Rust.

More information: https://github.com/imsnif/diskonaut.

- Start `diskonaut` in the current directory:

```
diskonaut
```

- Start `diskonaut` in a specific directory:

```
diskonaut {{path/to/directory}}
```

- Show file sizes rather than their block usage on the disk:

```
diskonaut --apparent-size {{path/to/directory}}
```

- Disable deletion confirmation:

```
diskonaut --disable-delete-confirmation
```

disown

Allow sub-processes to live beyond the shell that they are attached to.

See also: **`jobs`** for finding job numbers.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-disown.

- Disown the current job:

```
disown
```

- Disown a specific job (run `jobs` to find the job number):

```
disown %{{job_number}}
```

- Disown all jobs (Bash only):

```
disown -a
```

- Keep job (do not disown it), but mark it so that no future SIGHUP is received on shell exit (Bash only):

```
disown -h %{{job_number}}
```

distcc

Distributed C/C++/ObjC compilation client that works with **`distccd`**.

More information: https://manned.org/distcc.

- Compile a source file using a compiler like `gcc`:

```
distcc {{gcc}} -c {{path/to/source.c}} -o {{path/to/output.o}}
```

- Set remote hosts to distribute compilation:

```
export DISTCC_HOSTS="localhost {{ip1 ip2 ...}}"
```

- Compile a project with `make` using `distcc`:

```
make {{[-j|--jobs]}} {{parallel_jobs}} CC="distcc {{gcc}}"
```

- Show the list of current `distcc` hosts:

```
distcc --show-hosts
```

- Display help:

```
distcc --help
```

- Display version:

```
distcc --version
```

distccd

Server daemon for the distcc distributed compiler.

More information: https://distcc.github.io.

- Start a daemon with the default settings:

```
distccd --daemon
```

- Start a daemon, accepting connections from IPv4 private network ranges:

```
distccd --daemon --allow-private
```

- Start a daemon, accepting connections from a specific network address or address range:

```
distccd --daemon {{[-a|--allow]}} {{ip_address|network_prefix}}
```

- Start a daemon with a lowered priority that can run a maximum of 4 tasks at a time:

```
distccd --daemon {{[-j|--jobs]}} {{4}} {{[-N|--nice]}} {{5}}
```

- Start a daemon and register it via mDNS/DNS-SD (Zeroconf):

```
distccd --daemon --zeroconf
```

dive

Explore a Docker image, layer contents, and discover ways to shrink it.

More information: https://github.com/wagoodman/dive.

- Analyze a Docker image:

```
dive {{your_image_tag}}
```

- Build an image and start analyzing it:

```
dive build -t {{some_tag}}
```

django-admin

Django's utility for administrative tasks.

More information: https://docs.djangoproject.com/en/stable/ref/django-admin/.

- Create a new Django project:

  ```
  django-admin startproject {{project_name}}
  ```

- Create a new app for the current project:

  ```
  django-admin startapp {{app_name}}
  ```

- Display help for a specific command:

  ```
  django-admin help {{command}}
  ```

- Display version:

  ```
  django-admin --version
  ```

djxl

Decompress JPEG XL images.

Accepted output extensions are PNG, APNG, JPEG, EXR, PGM, PPM, PNM, PFM, PAM, EXIF, XMP and JUMBF.

More information: https://github.com/libjxl/libjxl.

- Decompress a JPEG XL image to another format:

```
djxl {{path/to/image.jxl}} {{path/to/output.ext}}
```

- Display an extremely detailed help page:

```
djxl {{[-h -v -v -v -v|--help --verbose --verbose --verbose --verbose]}}
```

dlv

Debugger for the Go programming language.

More information: https://github.com/go-delve/delve/blob/master/Documentation/usage/dlv.md.

- Compile and begin debugging the main package in the current directory (by default, with no arguments):

```
dlv debug
```

- Compile and begin debugging a specific package:

```
dlv debug {{package}} {{arguments}}
```

- Compile a test binary and begin debugging the compiled program:

```
dlv test
```

- Connect to a headless debug server:

```
dlv connect {{ip_address}}
```

- Attach to a running process and begin debugging:

```
dlv attach {{pid}}
```

- Compile and begin tracing a program:

```
dlv trace {{package}} --regexp '{{regex}}'
```

dmd

Official D compiler.

More information: https://dlang.org/dmd.html.

- Build a D source file:

```
dmd {{path/to/source.d}}
```

- Generate code for all template instantiations:

```
dmd -allinst
```

- Control bounds checking:

```
dmd -boundscheck={{on|safeonly|off}}
```

- List information on all available checks:

```
dmd -check={{h|help|?}}
```

- Turn on colored console output:

```
dmd -color
```

dmypy

Type check Python code, running **mypy** as a daemon for better speed.

See also: **mypy** for more options to use with check and run.

More information: https://mypy.readthedocs.io/en/stable/mypy_daemon.html.

- Type check a file, and start the daemon if it is not running:

```
dmypy check -- {{path/to/file.py}}
```

- Start the daemon:

```
dmypy start
```

- Type check a file (requires the daemon to be running):

```
dmypy run -- {{path/to/file.py}}
```

- Stop the daemon:

```
dmypy stop
```

dnsmasq

Lightweight DNS, DHCP, TFTP, and PXE server.

More information: https://manned.org/dnsmasq.

- Start dnsmasq with default configuration:

```
dnsmasq
```

- Run dnsmasq in the foreground (for debugging):

```
dnsmasq --no-daemon
```

- Specify a custom configuration file:

```
dnsmasq --conf-file={{path/to/config.conf}}
```

- Enable verbose logging:

```
dnsmasq --log-queries --log-facility=-
```

- Set a DHCP range and lease time:

```
dnsmasq --dhcp-range={{192.168.0.50,192.168.0.150,12h}}
```

- Print dnsmasq version:

```
dnsmasq --version
```

dnswalk

DNS debugger.

"Walk" across zones and validate database consistency and best practices.

More information: https://manned.org/dnswalk.

- Debug a DNS pathway for a Fully Qualified Domain Name (FQDN):

```
dnswalk {{domain}}.
```

- Process sub-domains [r]ecursively:

```
dnswalk -r {{domain}}.
```

- Only perform a `dnswalk` if the zone has been [m]odified since the last run:

```
dnswalk -m {{domain}}.
```

- Print [d]ebugging and status information to `stderr` instead of `stdout`:

```
dnswalk -d {{domain}}.
```

- Suppress the check for [i]nvalid characters in the domain name:

```
dnswalk -i {{domain}}.
```

- Enable duplicate A record warnings:

```
dnswalk -a {{domain}}.
```

- Enable "[F]ascist checking" to compare the A record PTR name with the forward name and report mismatches:

```
dnswalk -F {{domain}}.
```

- Enable "[l]ame delegation" to test whether the listed host is returning authoritative answers:

```
dnswalk -l {{domain}}.
```

dnsx

A fast and multi-purpose DNS toolkit to run multiple DNS queries.

Note: Input to **`dnsx`** needs to be passed through **`stdin`** (pipe **`|`**) in some cases.

See also: **`dig`**, **`dog`**, **`dnstracer`**.

More information: https://docs.projectdiscovery.io/opensource/dnsx/usage.

- Query the A record of a (sub)domain and show [re]sponse received:

```
echo {{example.com}} | dnsx -a {{[-re|-resp]}}
```

- Query all the DNS records (A, AAAA, CNAME, NS, TXT, SRV, PTR, MX, SOA, AXFR, CAA):

```
dnsx <<< {{example.com}} -recon {{[-re|-resp]}}
```

- Query a specific type of DNS record:

```
echo {{example.com}} | dnsx {{[-re|-resp]}} -{{a|aaaa|cname|
ns|txt|srv|ptr|mx|soa|any|axfr|caa}}
```

- Output response only (do not show the queried domain or subdomain):

```
echo {{example.com}} | dnsx {{[-ro|-resp-only]}}
```

- Display raw response of a query, specifying resolvers to use and retry attempts for failures:

```
echo {{example.com}} | dnsx -{{debug|raw}} {{[-r|-resolver]}}
{{1.1.1.1,8.8.8.8,...}} -retry {{number}}
```

- Brute force DNS records using a placeholder:

```
dnsx {{[-d|-domain]}} {{FUZZ.example.com}} {{[-w|-wordlist]}}
{{path/to/wordlist.txt}} {{[-re|-resp]}}
```

- Brute force DNS records from a list of domains and wordlists, appending output to a file with no color codes:

```
dnsx {{[-d|-domain]}} {{path/to/domain.txt}} {{[-w|-
wordlist]}} {{path/to/wordlist.txt}} {{[-re|-resp]}} {{[-o|-
output]}} {{path/to/output.txt}} {{[-nc|-no-color]}}
```

- Extract `CNAME` records for the given list of subdomains, with rate limiting DNS queries per second:

```
subfinder -silent {{[-d|-domain]}} {{example.com}} | dnsx -
cname {{[-re|-resp]}} {{[-rl|-rate-limit]}} {{number}}
```

do

This shell keyword is used with **`for`**, **`while`**, **`select`**, and **`until`** to mark the start of a loop.

- View documentation for the `for` keyword:

  ```
  tldr for
  ```

- View documentation for the `while` keyword:

  ```
  tldr while
  ```

- View documentation for the `select` keyword:

  ```
  tldr select
  ```

- View documentation for the `until` keyword:

  ```
  tldr until
  ```

doas

Execute a command as another user.

See also: **`sudo`**, **`pkexec`**, **`run0`**.

More information: https://man.openbsd.org/doas.

- Run a command as root:

```
doas {{command}}
```

- Run a command as another user:

```
doas -u {{user}} {{command}}
```

- Launch the default shell as root:

```
doas -s
```

- Parse a configuration file and check if the execution of a command as another user is allowed:

```
doas -C {{path/to/config_file}} {{command}}
```

- Make `doas` request a password even after it was supplied earlier:

```
doas -L
```

dockdiver

A tool for interacting with Docker registries, including listing and dumping repositories.

More information: https://github.com/MachiavelliII/dockdiver.

- List all repositories in a Docker registry:

```
dockdiver -url {{https://example.com}} -list
```

- Dump a specific repository to the default output directory (docker_dump):

```
dockdiver -url {{https://example.com}} -dump
{{repository_name}}
```

- Dump all repositories with basic authentication:

```
dockdiver -url {{https://example.com}} -dump-all -username
{{username}} -password {{password}}
```

- Dump a repository with a rate limit and a custom port (the default port is `5000`):

```
dockdiver -url {{https://example.com}} -dump
{{repository_name}} -port {{port}} -rate
{{requests_per_second}} -dir {{path/to/output_directory}}
```

- Dump all repositories with bearer token for authorization:

```
dockdiver -url {{https://example.com}} -dump-all -bearer
{{bearer_token}}
```

- Add custom headers as JSON (e.g., '{"X-Custom": "Value"}'):

```
dockdiver -url {{https://example.com}} -list -headers '{{{"X-
Custom": "Value"}}}'
```

docker build

Build an image from a Dockerfile.

More information: https://docs.docker.com/reference/cli/docker/buildx/build/.

- Build a Docker image using the Dockerfile in the current directory:

```
docker build .
```

- Build a Docker image from a Dockerfile at a specified URL:

```
docker build {{github.com/creack/docker-firefox}}
```

- Build a Docker image and tag it:

```
docker build {{[-t|--tag]}} {{name:tag}} .
```

- Build a Docker image with no build context:

```
docker < {{Dockerfile}} build {{[-t|--tag]}} {{name:tag}} -
```

- Do not use the cache when building the image:

```
docker build --no-cache {{[-t|--tag]}} {{name:tag}} .
```

- Build a Docker image using a specific Dockerfile:

```
docker build {{[-f|--file]}} {{Dockerfile}} .
```

- Build with custom build-time variables:

```
docker build --build-arg {{HTTP_PROXY=http://
10.20.30.2:1234}} --build-arg {{FTP_PROXY=http://
40.50.60.5:4567}} .
```

docker commit

Create a new image from a container's changes.

More information: https://docs.docker.com/reference/cli/docker/container/commit/.

- Create an image from a specific container:

```
docker commit {{container}} {{image}}:{{tag}}
```

- Apply a `CMD` Dockerfile instruction to the created image:

```
docker commit {{[-c|--change]}} "CMD
{{command}}" {{container}} {{image}}:{{tag}}
```

- Apply an `ENV` Dockerfile instruction to the created image:

```
docker commit {{[-c|--change]}} "ENV {{name}}
={{value}}" {{container}} {{image}}:{{tag}}
```

- Create an image with a specific author in the metadata:

```
docker commit {{[-a|--author]}} "{{author}}" {{container}}
{{image}}:{{tag}}
```

- Create an image with a specific comment in the metadata:

```
docker commit {{[-m|--message]}} "{{comment}}" {{container}}
{{image}}:{{tag}}
```

- Create an image without pausing the container during commit:

```
docker commit {{[-p|--pause]}} {{false}} {{container}}
{{image}}:{{tag}}
```

- Display help:

```
docker commit --help
```

docker compose down

Stop and remove containers, networks, images, and volumes created by **`docker compose up`**.

More information: https://docs.docker.com/reference/cli/docker/compose/down/.

- Stop and remove all containers and networks:

```
docker compose down
```

- Stop and remove containers, networks, and all images used by services:

```
docker compose down --rmi all
```

- Stop and remove containers, networks, and only images without a custom tag:

```
docker compose down --rmi local
```

- Stop and remove containers, networks, and all volumes:

```
docker compose down {{[-v|--volumes]}}
```

- Stop and remove everything including orphaned containers:

```
docker compose down --remove-orphans
```

- Stop and remove containers using an alternate compose file:

```
docker compose {{[-f|--file]}} {{path/to/config}} down
```

- Stop and remove containers with a custom timeout in seconds:

```
docker compose down {{[-t|--timeout]}} {{timeout}}
```

- Remove containers for services not defined in the Compose file:

```
docker compose down --remove-orphans {{[-v|--volumes]}}
```

docker compose logs

View output from containers in a Docker Compose application.

More information: https://docs.docker.com/reference/cli/docker/compose/logs/.

- View logs for all services:

```
docker compose logs
```

- View logs for a specific service:

```
docker compose logs {{service_name}}
```

- View logs and follow new output (like `tail --follow`):

```
docker compose logs {{[-f|--follow]}}
```

- View logs with timestamps:

```
docker compose logs {{[-t|--timestamps]}}
```

- View only the last `n` lines of logs for each container:

```
docker compose logs {{[-n|--tail]}} {{n}}
```

- View logs from a specific time onwards:

```
docker compose logs --since {{timestamp}}
```

- View logs until a specific time:

```
docker compose logs --until {{timestamp}}
```

- View logs for multiple specific services:

```
docker compose logs {{service1 service2 ...}}
```

docker compose up

Start and run Docker services defined in a Compose file.

More information: https://docs.docker.com/compose/reference/up/.

- Start all services defined in the docker-compose file:

```
docker compose up
```

- Start services in the background (detached mode):

```
docker compose up {{[-d|--detach]}}
```

- Start services and rebuild images before starting:

```
docker compose up --build
```

- Start specific services only:

```
docker compose up {{service1 service2 ...}}
```

- Start services with custom compose file:

```
docker compose {{[-f|--file]}} {{path/to/config}} up
```

- Start services and remove orphaned containers:

```
docker compose up --remove-orphans
```

- Start services with scaled instances:

```
docker compose up --scale {{service}}={{count}}
```

- Start services and show logs with timestamps:

```
docker compose up --timestamps
```

docker compose

Run and manage multi container Docker applications.

More information: https://docs.docker.com/reference/cli/docker/compose/.

- List all running containers:

```
docker compose ps
```

- Create and start all containers in the background using a `docker-compose.yml` file from the current directory:

```
docker compose up {{[-d|--detach]}}
```

- Start all containers, rebuild if necessary:

```
docker compose up --build
```

- Start all containers by specifying a project name and using an alternate compose file:

```
docker compose {{[-p|--project-name]}} {{project_name}} {{[-f|--file]}} {{path/to/file}} up
```

- Stop all running containers:

```
docker compose stop
```

- Stop and remove all containers, networks, images, and volumes:

```
docker compose down --rmi all {{[-v|--volumes]}}
```

- Follow logs for all containers:

```
docker compose logs {{[-f|--follow]}}
```

- Follow logs for a specific container:

```
docker compose logs {{[-f|--follow]}} {{container_name}}
```

docker container diff

This command is an alias of **`docker diff`**.

- View documentation for the original command:

  ```
  tldr docker diff
  ```

docker container remove

This command is an alias of **`docker rm`**.

- View documentation for the original command:

```
tldr docker rm
```

docker container rename

This command is an alias of **`docker rename`**.

- View documentation for the original command:

```
tldr docker rename
```

docker container rm

This command is an alias of **`docker rm`**.

- View documentation for the original command:

```
tldr docker rm
```

docker container top

This command is an alias of **`docker top`**.

- View documentation for the original command:

```
tldr docker top
```

docker container

Manage Docker containers.

More information: https://docs.docker.com/reference/cli/docker/container/.

- List currently running Docker containers:

```
docker container ls
```

- Start one or more stopped containers:

```
docker container start {{container1_name}}
{{container2_name}}
```

- Kill one or more running containers:

```
docker container kill {{container_name}}
```

- Stop one or more running containers:

```
docker container stop {{container_name}}
```

- Pause all processes within one or more containers:

```
docker container pause {{container_name}}
```

- Display detailed information on one or more containers:

```
docker container inspect {{container_name}}
```

- Export a container's filesystem as a tar archive:

```
docker container export {{container_name}}
```

- Create a new image from a container's changes:

```
docker container commit {{container_name}}
```

docker context

Switch between contexts to manage multiple Docker environments.

More information: https://docs.docker.com/reference/cli/docker/context/.

- Create a context using a specific Docker endpoint:

```
docker context create {{my_context}} --docker "host={{tcp://remote-host:2375}}"
```

- Create a context based on the `DOCKER_HOST` environment variable:

```
docker context create {{my_context}}
```

- Switch to a context:

```
docker context use {{my_context}}
```

- List all contexts:

```
docker context ls
```

docker cp

Copy files or directories between host and container filesystems.

More information: https://docs.docker.com/reference/cli/docker/container/cp/.

- Copy a file or directory from the host to a container:

```
docker cp {{path/to/file_or_directory_on_host}}
{{container_name}}:{{path/to/file_or_directory_in_container}}
```

- Copy a file or directory from a container to the host:

```
docker cp {{container_name}}:{{path/to/
file_or_directory_in_container}} {{path/to/
file_or_directory_on_host}}
```

- Copy a file or directory from the host to a container, following symlinks (copies the symlinked files directly, not the symlinks themselves):

```
docker cp {{[-L|--follow-link]}} {{path/to/symlink_on_host}}
{{container_name}}:{{path/to/file_or_directory_in_container}}
```

docker diff

Inspect changes to files or directories on a container's filesystem.

More information: https://docs.docker.com/reference/cli/docker/container/diff/.

- Inspect the changes to a container since it was created:

```
docker diff {{container}}
```

- Display help:

```
docker diff --help
```

docker exec

Execute a command on an already running Docker container.

More information: https://docs.docker.com/reference/cli/docker/container/exec/.

- Enter an interactive shell session on an already-running container:

```
docker exec {{[-it|--interactive --tty]}} {{container_name}} {{/bin/bash}}
```

- Run a command in the background (detached) on a running container:

```
docker exec {{[-d|--detach]}} {{container_name}} {{command}}
```

- Select the working directory for a given command to execute into:

```
docker exec {{[-it|--interactive --tty]}} {{[-w|--workdir]}} {{path/to/directory}} {{container_name}} {{command}}
```

- Run a command in background on existing container but keep `stdin` open:

```
docker exec {{[-i|--interactive]}} {{[-d|--detach]}} {{container_name}} {{command}}
```

- Set an environment variable in a running Bash session:

```
docker exec {{[-it|--interactive --tty]}} {{[-e|--env]}} {{variable_name}}={{value}} {{container_name}} {{/bin/bash}}
```

- Run a command as a specific user:

```
docker exec {{[-u|--user]}} {{user}} {{container_name}} {{command}}
```

docker image

Manage Docker images.

See also: **`docker build`**, **`docker import`**, **`docker pull`**.

More information: https://docs.docker.com/reference/cli/docker/image/.

- List local Docker images:

```
docker image ls
```

- Delete unused local Docker images:

```
docker image prune
```

- Delete all unused images (not just those without a tag):

```
docker image prune {{[-a|--all]}}
```

- Show the history of a local Docker image:

```
docker image history {{image}}
```

- View documentation for `docker image rm`:

```
tldr docker rmi
```

docker images

Manage Docker images.

More information: https://docs.docker.com/reference/cli/docker/image/ls/.

- List all Docker images:

```
docker images
```

- List all Docker images including intermediates:

```
docker images {{[-a|--all]}}
```

- List the output in quiet mode (only numeric IDs):

```
docker images {{[-q|--quiet]}}
```

- List all Docker images not used by any container:

```
docker images {{[-f|--filter]}} dangling=true
```

- List images that contain a substring in their name:

```
docker images "{{*name*}}"
```

- Sort images by size:

```
docker images --format "\{\{.ID\}\}\t\{\{.Size\}\}\t\{\
{.Repository\}\}:\{\{.Tag\}\}" | sort {{[-k|--key]}} 2 {{[-
h|--human-numeric-sort]}}
```

docker inspect

Return low-level information on Docker objects.

More information: https://docs.docker.com/reference/cli/docker/inspect/.

- Display help:

```
docker inspect
```

- Display information about a container, image, or volume using a name or ID:

```
docker inspect {{container|image|ID}}
```

- Display a container's IP address:

```
docker inspect {{[-f|--format]}} '\{\
{range.NetworkSettings.Networks\}\}\{\{.IPAddress\}\}\{\
{end\}\}' {{container}}
```

- Display the path to the container's log file:

```
docker inspect {{[-f|--format]}} '\{\{.LogPath\}
\}' {{container}}
```

- Display the image name of the container:

```
docker inspect {{[-f|--format]}} '\{\{.Config.Image\}
\}' {{container}}
```

- Display the configuration information as JSON:

```
docker inspect {{[-f|--format]}} '\{\{json .Config\}
\}' {{container}}
```

- Display all port bindings:

```
docker inspect {{[-f|--format]}} '\{\{range $p,
$conf := .NetworkSettings.Ports\}\} \{\{$p\}\} -> \{\{(index
$conf 0).HostPort\}\} \{\{end\}\}' {{container}}
```

docker load

Load Docker images from files or **`stdin`**.

More information: https://docs.docker.com/reference/cli/docker/image/load/.

- Load a Docker image from `stdin`:

```
docker < {{path/to/image_file.tar}} load
```

- Load a Docker image from a specific file:

```
docker load {{[-i|--input]}} {{path/to/image_file.tar}}
```

- Load a Docker image from a specific file in quiet mode:

```
docker load {{[-q|--quiet]}} {{[-i|--input]}} {{path/to/
image_file.tar}}
```

docker login

Log into a Docker registry.

More information: https://docs.docker.com/reference/cli/docker/login/.

- Interactively log into a registry:

```
docker login
```

- Log into a registry with a specific username (user will be prompted for a password):

```
docker login {{[-u|--username]}} {{username}}
```

- Log into a registry with username and password:

```
docker login {{[-u|--username]}} {{username}} {{[-p|--password]}} {{password}} {{server}}
```

- Log into a registry with password from `stdin`:

```
echo "{{password}}" | docker login {{[-u|--username]}} {{username}} --password-stdin
```

docker logs

Print container logs.

More information: https://docs.docker.com/reference/cli/docker/container/logs/.

- Print logs from a container:

```
docker logs {{container_name}}
```

- Print logs and follow them:

```
docker logs {{[-f|--follow]}} {{container_name}}
```

- Print last 5 lines:

```
docker logs {{container_name}} {{[-n|--tail]}} {{5}}
```

- Print logs and append them with timestamps:

```
docker logs {{[-t|--timestamps]}} {{container_name}}
```

- Print logs from a certain point in time of container execution (i.e. 23m, 10s, 2013-01-02T13:23:37):

```
docker logs {{container_name}} --until {{time}}
```

docker-machine

Create and manage machines running Docker.

More information: https://github.com/docker/machine.

- List currently running Docker machines:

```
docker-machine ls
```

- Create a new Docker machine with specific name:

```
docker-machine create {{name}}
```

- Get the status of a machine:

```
docker-machine status {{name}}
```

- Start a machine:

```
docker-machine start {{name}}
```

- Stop a machine:

```
docker-machine stop {{name}}
```

- Inspect information about a machine:

```
docker-machine inspect {{name}}
```

docker network

Create and manage Docker networks.

More information: https://docs.docker.com/reference/cli/docker/network/.

- List all available and configured networks on Docker daemon:

  ```
  docker network ls
  ```

- Create a user-defined network:

  ```
  docker network create {{[-d|--driver]}} {{driver_name}}
  {{network_name}}
  ```

- Display detailed information about one or more networks:

  ```
  docker network inspect {{network_name1 network_name2 ...}}
  ```

- Connect a container to a network using a name or ID:

  ```
  docker network connect {{network_name}} {{container_name|ID}}
  ```

- Disconnect a container from a network:

  ```
  docker network disconnect {{network_name}} {{container_name|
  ID}}
  ```

- Remove all unused (not referenced by any container) networks:

  ```
  docker network prune
  ```

- Remove one or more unused networks:

  ```
  docker network rm {{network_name1 network_name2 ...}}
  ```

docker node

Manage Docker Swarm nodes.

More information: https://docs.docker.com/reference/cli/docker/node/.

- List nodes in the swarm:

```
docker node ls
```

- List tasks running on one or more nodes, defaults to the current node:

```
docker node ps {{node1 node2 node3 ...}}
```

- Display detailed information on one or more nodes:

```
docker node inspect {{node1 node2 node3 ...}}
```

- Promote one or more nodes to manager in the swarm:

```
docker node promote {{node1 node2 node3 ...}}
```

- Demote one or more nodes from manager in the swarm:

```
docker node demote {{node1 node2 node3 ...}}
```

- Remove one or more nodes from the swarm:

```
docker node rm {{node1 node2 node3 ...}}
```

- Update metadata about a node, such as its availability, labels, or roles:

```
docker node update --{{availability|role|label-add|...}}
{{active|worker|...}} {{node1}}
```

docker ps

List Docker containers.

More information: https://docs.docker.com/reference/cli/docker/container/ls/.

- List currently running Docker containers:

```
docker ps
```

- List all Docker containers (running and stopped):

```
docker ps {{[-a|--all]}}
```

- Show the latest created container (includes all states):

```
docker ps {{[-l|--latest]}}
```

- Filter containers that contain a substring in their name:

```
docker ps {{[-f|--filter]}} "name={{name}}"
```

- Filter containers that share a given image as an ancestor:

```
docker ps {{[-f|--filter]}} "ancestor={{image}}:{{tag}}"
```

- Filter containers by exit status code:

```
docker ps {{[-a|--all]}} {{[-f|--filter]}} "exited={{code}}"
```

- Filter containers by status (created, running, removing, paused, exited and dead):

```
docker ps {{[-f|--filter]}} "status={{status}}"
```

- Filter containers that mount a specific volume or have a volume mounted in a specific path:

```
docker ps {{[-f|--filter]}} "volume={{path/to/directory}}" --
format "table {{.ID}}\t{{.Image}}\t{{.Names}}\t{{.Mounts}}"
```

docker pull

Download Docker images from a registry.

More information: https://docs.docker.com/reference/cli/docker/image/pull/.

- Download a specific Docker image:

```
docker pull {{image}}:{{tag}}
```

- Download a specific Docker image in quiet mode:

```
docker pull {{[-q|--quiet]}} {{image}}:{{tag}}
```

- Download all tags of a specific Docker image:

```
docker pull {{[-a|--all-tags]}} {{image}}
```

- Download a Docker images for a specific platform, e.g. linux/amd64:

```
docker pull --platform {{linux/amd64}} {{image}}:{{tag}}
```

- Display help:

```
docker pull {{[-h|--help]}}
```

docker rename

Rename a container.

More information: https://docs.docker.com/reference/cli/docker/container/rename/.

- Rename a container:

```
docker rename {{container}} {{new_name}}
```

- Display help:

```
docker rename --help
```

docker rm

Remove containers.

More information: https://docs.docker.com/reference/cli/docker/container/rm/.

- Remove containers:

```
docker rm {{container1 container2 ...}}
```

- Force remove a container:

```
docker rm {{[-f|--force]}} {{container1 container2 ...}}
```

- Remove a container and its volumes:

```
docker rm {{[-v|--volumes]}} {{container}}
```

- Display help:

```
docker rm {{[-h|--help]}}
```

docker rmi

Remove Docker images.

More information: https://docs.docker.com/reference/cli/docker/image/rm/.

- Display help:

```
docker rmi
```

- Remove one or more images given their names:

```
docker rmi {{image1 image2 ...}}
```

- Force remove an image:

```
docker rmi {{[-f|--force]}} {{image}}
```

- Remove an image without deleting untagged parents:

```
docker rmi --no-prune {{image}}
```

docker run

Run a command in a new Docker container.

More information: https://docs.docker.com/reference/cli/docker/container/run/.

- Run command in a new container from a tagged image:

```
docker run {{image:tag}} {{command}}
```

- Run command in a new container in background and display its ID:

```
docker run {{[-d|--detach]}} {{image}} {{command}}
```

- Run command in a one-off container in interactive mode and pseudo-TTY:

```
docker run --rm {{[-it|--interactive --tty]}} {{image}}
{{command}}
```

- Run command in a new container with passed environment variables:

```
docker run {{[-e|--env]}} '{{variable}}={{value}}' {{[-e|--
env]}} {{variable}} {{image}} {{command}}
```

- Run command in a new container with bind mounted volumes:

```
docker run {{[-v|--volume]}} /{{path/to/host_path}}:/{{path/
to/container_path}} {{image}} {{command}}
```

- Run command in a new container with published ports:

```
docker run {{[-p|--publish]}} {{host_port}}:
{{container_port}} {{image}} {{command}}
```

- Run command in a new container overwriting the entrypoint of the image:

```
docker run --entrypoint {{command}} {{image}}
```

- Run command in a new container connecting it to a network:

```
docker run --network {{network}} {{image}}
```

docker save

Export Docker images to archive.

More information: https://docs.docker.com/reference/cli/docker/image/save/.

- Save an image by redirecting `stdout` to a tar archive:

```
docker save {{image}}:{{tag}} > {{path/to/file.tar}}
```

- Save an image to a tar archive:

```
docker save {{[-o|--output]}} {{path/to/file.tar}} {{image}}:
{{tag}}
```

- Save all tags of the image:

```
docker save {{[-o|--output]}} {{path/to/file.tar}}
{{image_name}}
```

- Cherry-pick particular tags of an image to save:

```
docker save {{[-o|--output]}} {{path/to/file.tar}}
{{image_name:tag1 image_name:tag2 ...}}
```

docker search

Search for Docker images on Docker Hub.

More information: https://docs.docker.com/reference/cli/docker/search/.

- Search for Docker images by name or keyword:

```
docker search {{keyword}}
```

- Search for images and only show official ones:

```
docker search {{[-f|--filter]}} is-official=true {{keyword}}
```

- Search for images and only show automated builds:

```
docker search {{[-f|--filter]}} is-automated=true {{keyword}}
```

- Search for images with a minimum number of stars:

```
docker search {{[-f|--filter]}} stars={{number}} {{keyword}}
```

- Limit the number of results:

```
docker search --limit {{number}} {{keyword}}
```

- Customize the output format:

```
docker search {{[-f|--format]}} "{{.Name}}:
{{.Description}}" {{keyword}}
```

docker secret

Manage Docker swarm secrets.

More information: https://docs.docker.com/reference/cli/docker/secret/.

- Create a new secret from `stdin`:

```
{{command}} | docker secret create {{secret_name}} -
```

- Create a new secret from a file:

```
docker secret create {{secret_name}} {{path/to/file}}
```

- List all secrets:

```
docker secret ls
```

- Display detailed information on one or multiple secrets in a human friendly format:

```
docker secret inspect --pretty {{secret_name1
secret_name2 ...}}
```

- Remove one or more secrets:

```
docker secret rm {{secret_name1 secret_name2 ...}}
```

docker service

Manage the services on a Docker daemon.

More information: https://docs.docker.com/reference/cli/docker/service/.

- List the services on a Docker daemon:

```
docker service ls
```

- Create a new service:

```
docker service create --name {{service_name}} {{image}}:
{{tag}}
```

- Display detailed information about one or more services:

```
docker service inspect {{service_name_or_ID1
service_name_or_ID2}}
```

- List the tasks of one or more services:

```
docker service ps {{service_name_or_ID1
service_name_or_ID2 ...}}
```

- Scale to a specific number of replicas for a space-separated list of services:

```
docker service scale {{service_name}}={{count_of_replicas}}
```

- Remove one or more services:

```
docker service rm {{service_name_or_ID1 service_name_or_ID2}}
```

docker-slim

Analyze and optimize Docker images.

More information: https://github.com/slimtoolkit/slim.

- Start DockerSlim on interactive mode:

```
docker-slim
```

- Analyze Docker layers from a specific image:

```
docker-slim xray --target {{image:tag}}
```

- Lint a Dockerfile:

```
docker-slim lint --target {{path/to/Dockerfile}}
```

- Analyze and generate an optimized Docker image:

```
docker-slim build {{image:tag}}
```

- Display help for a subcommand:

```
docker-slim {{subcommand}} --help
```

docker start

Start stopped containers.

More information: https://docs.docker.com/reference/cli/docker/container/start/.

- Display help:

```
docker start --help
```

- Start a Docker container:

```
docker start {{container}}
```

- Start a container, attaching `stdout` and `stderr` and forwarding signals:

```
docker start {{[-a|--attach]}} {{container}}
```

- Start one or more containers:

```
docker start {{container1 container2 ...}}
```

docker stats

Display a live stream of resource usage statistics for containers.

More information: https://docs.docker.com/reference/cli/docker/container/stats/.

- Display a live stream for the statistics of all running containers:

```
docker stats
```

- Display a live stream of statistics for one or more containers:

```
docker stats {{container1 container2 ...}}
```

- Change the columns format to display container's CPU usage percentage:

```
docker stats --format "{{.Name}}:\t{{.CPUPerc}}"
```

- Display statistics for all containers (both running and stopped):

```
docker stats {{[-a|--all]}}
```

- Disable streaming stats and only pull the current stats:

```
docker stats --no-stream
```

docker swarm

A container orchestration tool.

More information: https://docs.docker.com/engine/swarm/.

- Initialize a swarm cluster:

```
docker swarm init
```

- Display the token to join a manager or a worker:

```
docker swarm join-token {{worker|manager}}
```

- Join a new node to the cluster:

```
docker swarm join --token {{token}} {{manager_node_url:2377}}
```

- Remove a worker from the swarm (run inside the worker node):

```
docker swarm leave
```

- Display the current CA certificate in PEM format:

```
docker swarm ca
```

- Rotate the current CA certificate and display the new certificate:

```
docker swarm ca --rotate
```

- Change the valid period for node certificates:

```
docker swarm update --cert-expiry {{hours}}h{{minutes}}
m{{seconds}}s
```

docker system

Manage Docker data and display system-wide information.

More information: https://docs.docker.com/reference/cli/docker/system/.

- Display help:

```
docker system
```

- Show Docker disk usage:

```
docker system df
```

- Show detailed information on disk usage:

```
docker system df {{[-v|--verbose]}}
```

- Remove unused data (append `--volumes` to remove unused volumes as well):

```
docker system prune
```

- Remove unused data created more than a specified amount of time in the past:

```
docker system prune --filter "until={{hours}}h{{minutes}}m"
```

- Display real-time events from the Docker daemon:

```
docker system events
```

- Display real-time events from containers streamed as valid JSON Lines:

```
docker system events {{[-f|--filter]}} 'type=container' --
format '{{json .}}'
```

- Display system-wide information:

```
docker system info
```

docker tag

Assign tags to existing Docker images.

More information: https://docs.docker.com/reference/cli/docker/image/tag/.

- Assign a name and tag to a specific image ID:

```
docker tag {{id}} {{name}}:{{tag}}
```

- Assign a tag to a specific image:

```
docker tag {{image}}:{{current_tag}} {{image}}:{{new_tag}}
```

- Display help:

```
docker tag
```

docker top

Display the running processes of a container.

More information: https://docs.docker.com/reference/cli/docker/container/top/.

- Display the running processes of a container:

```
docker top {{container}}
```

- Display help:

```
docker top --help
```

docker update

Update configuration of Docker containers.

This command is not supported for Windows containers.

More information: https://docs.docker.com/reference/cli/docker/container/update/.

- Update restart policy to apply when a specific container exits:

```
docker update --restart {{always|no|on-failure|unless-stopped}} {{container_name}}
```

- Update the policy to restart up to three times a specific container when it exits with non-zero exit status:

```
docker update --restart on-failure:3 {{container_name}}
```

- Update the number of CPUs available to a specific container:

```
docker update --cpus {{count}} {{container_name}}
```

- Update the memory limit in [M]egabytes for a specific container:

```
docker update {{[-m|--memory]}} {{limit}}M {{container_name}}
```

- Update the maximum number of process IDs allowed inside a specific container (use `-1` for unlimited):

```
docker update --pids-limit {{count}} {{container_name}}
```

- Update the amount of memory in [M]egabytes a specific container can swap to disk (use `-1` for unlimited):

```
docker update --memory-swap {{limit}}M {{container_name}}
```

docker volume

Manage Docker volumes.

More information: https://docs.docker.com/reference/cli/docker/volume/.

- Create a volume:

```
docker volume create {{volume_name}}
```

- Create a volume with a specific label:

```
docker volume create --label {{label}} {{volume_name}}
```

- Create a `tmpfs` volume a size of 100 MiB and an uid of 1000:

```
docker volume create {{[-o|--opt]}} {{type}}={{tmpfs}} {{[-
o|--opt]}} {{device}}={{tmpfs}} {{[-o|--opt]}} {{o}}
={{size=100m,uid=1000}} {{volume_name}}
```

- List all volumes:

```
docker volume ls
```

- Remove a volume:

```
docker volume rm {{volume_name}}
```

- Display information about a volume:

```
docker volume inspect {{volume_name}}
```

- Remove all unused local volumes:

```
docker volume prune
```

- Display help for a subcommand:

```
docker volume {{subcommand}} --help
```

docker

Manage Docker containers and images.

Some subcommands such as **`run`** have their own usage documentation.

More information: https://docs.docker.com/reference/cli/docker/.

- List all Docker containers (running and stopped):

```
docker ps {{[-a|--all]}}
```

- Start a container from an image, with a custom name:

```
docker run --name {{container_name}} {{image}}
```

- Start or stop an existing container:

```
docker {{start|stop}} {{container_name}}
```

- Pull an image from a Docker registry:

```
docker pull {{image}}
```

- Display the list of already downloaded images:

```
docker images
```

- Open an interactive tty with Bourne shell (`sh`) inside a running container:

```
docker exec {{[-it|--interactive --tty]}} {{container_name}} {{sh}}
```

- Remove stopped containers:

```
docker rm {{container1 container2 ...}}
```

- Fetch and follow the logs of a container:

```
docker logs {{[-f|--follow]}} {{container_name}}
```

docsify

Initialize and serve markdown documentation.

More information: https://cli.docsifyjs.org.

- Initialize a new documentation in the current directory:

```
docsify init
```

- Initialize a new documentation in the specified directory:

```
docsify init {{path/to/directory}}
```

- Serve local documentation on `localhost:3000` with live reload:

```
docsify serve {{path/to/directory}}
```

- Serve local documentation on `localhost` at the specified port:

```
docsify serve --port {{80}} {{path/to/directory}}
```

- Generate a sidebar markdown file in the specified directory:

```
docsify generate {{path/to/directory}}
```

doctl account

Retrieve information about Digital Ocean accounts.

More information: https://docs.digitalocean.com/reference/doctl/reference/account/.

- Display account info:

```
doctl account {{[g|get]}}
```

- Show the hourly API limit, progress towards it, and when the rate limit resets:

```
doctl account {{[rl|ratelimit]}}
```

- Display help:

```
doctl account {{[-h|--help]}}
```

doctl apps

Manage DigitalOcean apps.

More information: https://docs.digitalocean.com/reference/doctl/reference/apps.

- Create an app:

```
doctl {{[a|apps]}} {{[c|create]}}
```

- Create a deployment for a specific app:

```
doctl {{[a|apps]}} {{[cd|create-deployment]}} {{app_id}}
```

- Delete an app interactively:

```
doctl {{[a|apps]}} {{[d|delete]}} {{app_id}}
```

- Get an app:

```
doctl {{[a|apps]}} {{[g|get]}}
```

- List all apps:

```
doctl {{[a|apps]}} {{[ls|list]}}
```

- List all deployments from a specific app:

```
doctl {{[a|apps]}} {{[lsd|list-deployments]}} {{app_id}}
```

- Get logs from a specific app:

```
doctl {{[a|apps]}} {{[l|logs]}} {{app_id}}
```

- Update a specific app with a given app spec:

```
doctl {{[a|apps]}} {{[u|update]}} {{app_id}} --spec {{path/
to/spec.yml}}
```

doctl auth

Authenticate **`doctl`** with API tokens.

More information: https://docs.digitalocean.com/reference/doctl/reference/auth/.

- Open a prompt to enter an API token and label its context:

  ```
  doctl auth init --context {{token_label}}
  ```

- List authentication contexts (API tokens):

  ```
  doctl auth {{[ls|list]}}
  ```

- Switch contexts (API tokens):

  ```
  doctl auth switch --context {{token_label}}
  ```

- Remove a stored authentication context (API token):

  ```
  doctl auth remove --context {{token_label}}
  ```

- Show available commands:

  ```
  doctl auth {{[-h|--help]}}
  ```

doctl balance

Show the balance of a Digital Ocean account.

More information: https://docs.digitalocean.com/reference/doctl/reference/balance/.

- Get balance of the account associated with the current context:

```
doctl balance {{[g|get]}}
```

- Get the balance of an account associated with an access token:

```
doctl balance {{[g|get]}} {{[-t|--access-token]}} {{access_token}}
```

- Get the balance of an account associated with a specified context:

```
doctl balance {{[g|get]}} --context
```

doctl compute droplet

List, create, and delete virtual machines which are called droplets.

More information: https://docs.digitalocean.com/reference/doctl/reference/compute/droplet/.

- Create a droplet:

```
doctl compute {{[d|droplet]}} {{[c|create]}} --region
{{region}} --image {{os_image}} --size {{vps_type}}
{{droplet_name}}
```

- Delete a droplet:

```
doctl compute {{[d|droplet]}} {{[d|delete]}} {{droplet_id|
droplet_name}}
```

- List droplets:

```
doctl compute {{[d|droplet]}} {{[ls|list]}}
```

doctl databases db

Manage databases that are served by a database cluster.

More information: https://docs.digitalocean.com/reference/doctl/reference/databases/db.

- Run a `doctl databases db` command with an access token:

```
doctl {{[d|databases]}} db {{command}} {{[-t|--access-token]}} {{access_token}}
```

- Retrieve the name of the given database hosted in the given database cluster:

```
doctl {{[d|databases]}} db {{[g|get]}} {{database_id}} {{database_name}}
```

- List existing databases hosted within a given database cluster:

```
doctl {{[d|databases]}} db {{[ls|list]}} {{database_id}}
```

- Create a database with the given name in the given database cluster:

```
doctl {{[d|databases]}} db {{[c|create]}} {{database_id}} {{database_name}}
```

- Delete the database with the given name in the given database cluster:

```
doctl {{[d|databases]}} db {{[rm|delete]}} {{database_id}} {{database_name}}
```

doctl databases firewalls

Manage firewalls for database clusters.

More information: https://docs.digitalocean.com/reference/doctl/reference/databases/firewalls.

- Run a `doctl databases firewalls` command with an access token:

```
doctl {{[d|databases]}} {{[fw|firewalls]}} {{command}} {{[-t|--access-token]}} {{access_token}}
```

- Retrieve a list of firewall rules for a given database:

```
doctl {{[d|databases]}} {{[fw|firewalls]}} {{[ls|list]}}
```

- Add a database firewall rule to a given database:

```
doctl {{[d|databases]}} {{[fw|firewalls]}} {{[a|append]}} {{database_id}} --rule {{droplet|k8s|ip_addr|tag|app}}:{{value}}
```

- Remove a firewall rule for a given database:

```
doctl {{[d|databases]}} {{[fw|firewalls]}} {{[rm|remove]}} {{database_id}} {{rule_uuid}}
```

doctl databases maintenance-window

Schedule, and check the schedule of, maintenance windows for your databases.

More information: https://docs.digitalocean.com/reference/doctl/reference/databases/maintenance-window.

- Run a `doctl databases maintenance-window` command with an access token:

  ```
  doctl {{[d|databases]}} {{[mw|maintenance-window]}} {{command}} {{[-t|--access-token]}} {{access_token}}
  ```

- Retrieve details about a database cluster's maintenance windows:

  ```
  doctl {{[d|databases]}} {{[mw|maintenance-window]}} {{[g|get]}} {{database_id}}
  ```

- Update the maintenance window for a database cluster:

  ```
  doctl {{[d|databases]}} {{[mw|maintenance-window]}} {{[u|update]}} {{database_id}} --day {{day_of_the_week}} --hour {{hour_in_24_hours_format}}
  ```

doctl databases options

Enable the navigation of available options under each database engine.

More information: https://docs.digitalocean.com/reference/doctl/reference/databases/options.

- Run a `doctl databases options` command with an access token:

```
doctl {{[d|databases]}} {{[o|options]}} {{command}} {{[-t|--access-token]}} {{access_token}}
```

- Retrieve a list of the available database engines:

```
doctl {{[d|databases]}} {{[o|options]}} {{[eng|engines]}}
```

- Retrieve a list of the available regions for a given database engine:

```
doctl {{[d|databases]}} {{[o|options]}} {{[r|regions]}} --engine {{pg|mysql|redis|mongodb}}
```

- Retrieve a list of the available slugs for a given database engine:

```
doctl {{[d|databases]}} {{[o|options]}} {{[s|slugs]}} --engine {{pg|mysql|redis|mongodb}}
```

- Retrieve a list of the available versions for a given database engine:

```
doctl {{[d|databases]}} {{[o|options]}} {{[v|versions]}} --engine {{pg|mysql|redis|mongodb}}
```

doctl databases pool

Manage connection pools for your database cluster.

More information: https://docs.digitalocean.com/reference/doctl/reference/databases/pool/.

- Run a `doctl databases pool` command with an access token:

```
doctl {{[d|databases]}} {{[p|pool]}} {{command}} {{[-t|--access-token]}} {{access_token}}
```

- Retrieve information about a database connection pool:

```
doctl {{[d|databases]}} {{[p|pool]}} {{[g|get]}} {{database_id}} {{pool_name}}
```

- List connection pools for a database cluster:

```
doctl {{[d|databases]}} {{[p|pool]}} {{[ls|list]}} {{database_id}}
```

- Create a connection pool for a database:

```
doctl {{[d|databases]}} {{[p|pool]}} {{[c|create]}} {{database_id}} {{pool_name}} --db {{new_pool_name}} --size {{pool_size}}
```

- Delete a connection pool for a database:

```
doctl {{[d|databases]}} {{[p|pool]}} {{[c|create]}} {{database_id}} {{pool_name}}
```

doctl databases replica

Manage read-only replicas associated with a database cluster.

More information: https://docs.digitalocean.com/reference/doctl/reference/databases/replica/.

- Run a `doctl databases replica` command with an access token:

```
doctl {{[d|databases]}} {{[p|pool]}} {{command}} {{[-t|--access-token]}} {{access_token}}
```

- Retrieve information about a read-only database replica:

```
doctl {{[d|databases]}} {{[r|replica]}} {{[g|get]}} {{database_id}} {{replica_name}}
```

- Retrieve list of read-only database replicas:

```
doctl {{[d|databases]}} {{[r|replica]}} {{[ls|list]}} {{database_id}}
```

- Create a read-only database replica:

```
doctl {{[d|databases]}} {{[r|replica]}} {{[c|create]}} {{database_id}} {{replica_name}}
```

- Delete a read-only database replica:

```
doctl {{[d|databases]}} {{[r|replica]}} {{[rm|delete]}} {{database_id}} {{replica_name}}
```

doctl databases sql-mode

View and configure a MySQL database cluster's global SQL modes.

More information: https://docs.digitalocean.com/reference/doctl/reference/databases/sql-mode/.

- Run a `doctl databases sql-mode` command with an access token:

```
doctl {{[d|databases]}} {{[sm|sql-mode]}} {{command}} {{[-t|--access-token]}} {{access_token}}
```

- Get a MySQL database cluster's SQL modes:

```
doctl {{[d|databases]}} {{[sm|sql-mode]}} {{[g|get]}} {{database_id}}
```

- Overwrite a MySQL database cluster's SQL modes to the specified modes:

```
doctl {{[d|databases]}} {{[sm|sql-mode]}} {{[s|set]}} {{database_id}} {{sql_mode_1 sql_mode_2 ...}}
```

doctl databases user

View details for, and create, database users.

More information: https://docs.digitalocean.com/reference/doctl/reference/databases/user.

- Run a `doctl databases user` command with an access token:

```
doctl {{[d|databases]}} {{[u|user]}} {{command}} {{[-t|--access-token]}} {{access_token}}
```

- Retrieve details about a database user:

```
doctl {{[d|databases]}} {{[u|user]}} {{[g|get]}} {{database_id}} {{user_name}}
```

- Retrieve a list of database users for a given database:

```
doctl {{[d|databases]}} {{[u|user]}} {{[ls|list]}} {{database_id}}
```

- Reset the auth password for a given user:

```
doctl {{[d|databases]}} {{[u|user]}} {{[rs|reset]}} {{database id}} {{user_name}}
```

- Reset the MySQL auth plugn for a given user:

```
doctl {{[d|databases]}} {{[u|user]}} {{[rs|reset]}} {{database_id}} {{user_name}} {{caching_sha2_password|mysql_native_password}}
```

- Create a user in the given database with a given username:

```
doctl {{[d|databases]}} {{[u|user]}} {{[c|create]}} {{database_id}} {{user_name}}
```

- Delete a user from the given database with the given username:

```
doctl {{[d|databases]}} {{[u|user]}} {{[rm|delete]}} {{database_id}} {{user_name}}
```

doctl databases

Manage your MySQL, Redis, PostgreSQL, and MongoDB database services.

More information: https://docs.digitalocean.com/reference/doctl/reference/databases.

- Run a `doctl databases` command with an access token:

  ```
  doctl {{[d|databases]}} {{command}} {{[-t|--access-token]}} {{access_token}}
  ```

- Get details for a database cluster:

  ```
  doctl {{[d|databases]}} {{[g|get]}}
  ```

- List your database clusters:

  ```
  doctl {{[d|databases]}} {{[ls|list]}}
  ```

- Create a database cluster:

  ```
  doctl {{[d|databases]}} {{[c|create]}} {{database_name}}
  ```

- Delete a cluster:

  ```
  doctl {{[d|databases]}} {{[rm|delete]}} {{database_id}}
  ```

doctl kubernetes cluster

Manage Kubernetes clusters and view configuration options relating to clusters.

More information: https://docs.digitalocean.com/reference/doctl/reference/kubernetes/cluster/.

- Create a Kubernetes cluster:

```
doctl {{[k|kubernetes]}} {{[c|cluster]}} {{[c|create]}} --count {{3}} --region {{nyc1}} --size {{s-1vcpu-2gb}} --version {{latest}} {{cluster_name}}
```

- List all Kubernetes clusters:

```
doctl {{[k|kubernetes]}} {{[c|cluster]}} {{[ls|list]}}
```

- Fetch and save the kubeconfig:

```
doctl {{[k|kubernetes]}} {{[c|cluster]}} {{[cfg|kubeconfig]}} {{[s|save]}} {{cluster_name}}
```

- Check for available upgrades:

```
doctl {{[k|kubernetes]}} {{[c|cluster]}} {{[gu|get-upgrades]}} {{cluster_name}}
```

- Upgrade a cluster to a new Kubernetes version:

```
doctl {{[k|kubernetes]}} {{[c|cluster]}} upgrade {{cluster_name}}
```

- Delete a cluster:

```
doctl {{[k|kubernetes]}} {{[c|cluster]}} {{[d|delete]}} {{cluster_name}}
```

doctl kubernetes options

Get values available for use with **`doctl`**'s Kubernetes commands.

More information: https://docs.digitalocean.com/reference/doctl/reference/kubernetes/options/.

- List regions that support Kubernetes clusters:

```
doctl {{[k|kubernetes]}} {{[o|options]}} {{[r|regions]}}
```

- List machine sizes that can be used in a Kubernetes cluster:

```
doctl {{[k|kubernetes]}} {{[o|options]}} {{[s|sizes]}}
```

- List Kubernetes versions that can be used with DigitalOcean clusters:

```
doctl {{[k|kubernetes]}} {{[o|options]}} {{[v|versions]}}
```

doctl serverless

Manage serverless functions.

More information: https://docs.digitalocean.com/reference/doctl/reference/serverless/.

- Connect local serverless support to a functions namespace:

```
doctl {{[sls|serverless]}} connect
```

- Deploy a functions project to your functions namespace:

```
doctl {{[sls|serverless]}} deploy
```

- Obtain metadata of a functions project:

```
doctl {{[sls|serverless]}} get-metadata
```

- Provide information about serverless support:

```
doctl {{[sls|serverless]}} status
```

doctum

Generate documentation for a PHP API.

More information: https://github.com/code-lts/doctum.

- Parse a project:

```
doctum parse
```

- Render a project:

```
doctum render
```

- Parse then render a project:

```
doctum update
```

- Parse and render only a specific version of a project:

```
doctum update --only-version={{version}}
```

- Parse and render a project using a specific configuration:

```
doctum update {{path/to/config.php}}
```

dog

DNS lookup utility.

It has colorful output, supports DNS-over-TLS and DNS-over-HTTPS protocols, and can emit JSON.

More information: https://github.com/ogham/dog#examples.

- Lookup the IP(s) associated with a hostname (A records):

```
dog {{example.com}}
```

- Query the MX records type associated with a given domain name:

```
dog {{example.com}} MX
```

- Specify a specific DNS server to query (e.g. Cloudflare):

```
dog {{example.com}} MX @{{1.1.1.1}}
```

- Query over TCP rather than UDP:

```
dog {{example.com}} MX @{{1.1.1.1}} {{[-T|--tcp]}}
```

- Query the MX records type associated with a given domain name over TCP using explicit arguments:

```
dog {{[-q|--query]}} {{example.com}} {{[-t|--type]}} MX {{[-n|--nameserver]}} {{1.1.1.1}} {{[-T|--tcp]}}
```

- Lookup the IP(s) associated with a hostname (A records) using DNS over HTTPS (DoH):

```
dog {{example.com}} {{[-H|--https]}} @{{https://cloudflare-dns.com/dns-query}}
```

doggo

DNS client for Humans.

Written in Golang.

More information: https://github.com/mr-karan/doggo/blob/main/docs/src/content/docs/guide/reference.md.

- Perform a simple DNS lookup:

```
doggo {{example.com}}
```

- Query MX records using a specific nameserver:

```
doggo MX {{codeberg.org}} @{{1.1.1.2}}
```

- Use DNS over HTTPS:

```
doggo {{example.com}} @{{https://dns.quad9.net/dns-query}}
```

- Output in the JSON format:

```
doggo {{example.com}} {{[-J|--json]}} | jq
'{{.responses[0].answers[].address}}'
```

- Perform a reverse DNS lookup:

```
doggo {{[-x|--reverse]}} {{8.8.4.4}} --short
```

dokku

Docker powered mini-Heroku (PaaS).

Easily deploy multiple apps to your server in different languages using a single **`git-push`** command.

More information: https://dokku.com/docs/deployment/application-deployment/.

- List running apps:

  ```
  dokku apps
  ```

- Create an app:

  ```
  dokku apps:create {{app_name}}
  ```

- Remove an app:

  ```
  dokku apps:destroy {{app_name}}
  ```

- Install plugin:

  ```
  dokku plugin:install {{full_repo_url}}
  ```

- Link database to an app:

  ```
  dokku {{db}}:link {{db_name}} {{app_name}}
  ```

dolt add

Add the contents of a table to the list of Dolt staged tables.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-add.

- Add a table to the list of staged tables (stage a table):

```
dolt add {{table}}
```

- Stage all tables:

```
dolt add {{[-A|--all]}}
```

dolt blame

Display commit information for each row of a Dolt table.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-blame.

- Display the latest commit for each row of a table:

```
dolt blame {{table}}
```

- Display the latest commits for each row of a table when the specified commit was made:

```
dolt blame {{commit}} {{table}}
```

- Display help:

```
dolt blame --help
```

dolt branch

Manage Dolt branches.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-branch.

- List local branches (current branch is highlighted by *):

```
dolt branch
```

- List all local and remote branches:

```
dolt branch {{[-A|--all]}}
```

- Create a new branch based on the current branch:

```
dolt branch {{branch_name}}
```

- Create a new branch with the specified commit as the latest:

```
dolt branch {{branch_name}} {{commit}}
```

- Rename a branch:

```
dolt branch {{[-m|--move]}} {{branch_name1}} {{branch_name2}}
```

- Duplicate a branch:

```
dolt branch {{[-c|--copy]}} {{branch_name1}} {{branch_name2}}
```

- Delete a branch:

```
dolt branch {{[-d|--delete]}} {{branch_name}}
```

- Display the name of the current branch:

```
dolt branch --show-current
```

dolt checkout

Checkout the work tree or tables to a branch or commit.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-checkout.

- Switch to a branch:

```
dolt checkout {{branch_name}}
```

- Revert unstaged changes to a table:

```
dolt checkout {{table}}
```

- Create new branch and switch to it:

```
dolt checkout -b {{branch_name}}
```

- Create new branch based on a specified commit and switch to it:

```
dolt checkout -b {{branch_name}} {{commit}}
```

dolt clone

Clone a repository into a new directory.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-clone.

- Clone an existing repository into a specific directory (defaults to the repository name):

```
dolt clone {{repository_url}} {{path/to/directory}}
```

- Clone an existing repository and add a specific remote (defaults to origin):

```
dolt clone --remote {{remote_name}} {{repository_url}}
```

- Clone an existing repository only fetching a specific branch (defaults to all branches):

```
dolt clone {{[-b|--branch]}} {{branch_name}}
{{repository_url}}
```

- Clone a repository, using an AWS region (uses the profile's default region if none is provided):

```
dolt clone --aws-region {{region_name}} {{repository_url}}
```

- Clone a repository, using an AWS credentials file:

```
dolt clone --aws-creds-file {{credentials_file}}
{{repository_url}}
```

- Clone a repository, using an AWS credentials profile (uses the default profile if none is provided):

```
dolt clone --aws-creds-profile {{profile_name}}
{{repository_url}}
```

- Clone a repository, using an AWS credentials type:

```
dolt clone --aws-creds-type {{credentials_type}}
{{repository_url}}
```

dolt commit

Commit staged changes to tables.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-commit.

- Commit all staged changes, opening the editor specified by `$EDITOR` to enter the commit message:

```
dolt commit
```

- Commit all staged changes with the specified message:

```
dolt commit {{[-m|--message]}} "{{commit_message}}"
```

- Stage all unstaged changes to tables before committing:

```
dolt commit {{[-a|--all]}}
```

- Use the specified ISO 8601 commit date (defaults to current date and time):

```
dolt commit --date "{{2021-12-31T00:00:00}}"
```

- Use the specified author for the commit:

```
dolt commit --author "{{author_name}} <{{author_email}}>"
```

- Allow creating an empty commit, with no changes:

```
dolt commit --allow-empty
```

- Ignore foreign key warnings:

```
dolt commit {{[-f|--force]}}
```

dolt config

Read and write local (per repository) and global (per user) Dolt configuration variables.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-config.

- List all local and global configuration options and their values:

```
dolt config --list
```

- Display the value of a local or global configuration variable:

```
dolt config --get {{name}}
```

- Modify the value of a local configuration variable, creating it if it doesn't exist:

```
dolt config --add {{name}} {{value}}
```

- Modify the value of a global configuration variable, creating it if it doesn't exist:

```
dolt config --global --add {{name}} {{value}}
```

- Delete a local configuration variable:

```
dolt config --unset {{name}}
```

- Delete a global configuration variable:

```
dolt config --global --unset {{name}}
```

dolt fetch

Download objects and refs from another repository.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-fetch.

- Fetch the latest changes from the default remote upstream repository (origin):

```
dolt fetch
```

- Fetch latest changes from a specific remote upstream repository:

```
dolt fetch {{remote_name}}
```

- Update branches with the current state of the remote, overwriting any conflicting history:

```
dolt fetch -f
```

dolt gc

Search the repository for data that are no longer referenced and no longer needed.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-gc.

- Clean up unreferenced data from the repository:

```
dolt gc
```

- Initiate a faster but less thorough garbage collection process:

```
dolt gc {{[-s|--shallow]}}
```

dolt init

Create an empty Dolt data repository.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-init.

- Initialize a new Dolt data repository in the current directory:

```
dolt init
```

- Initialize a new Dolt data repository creating a commit with the specified metadata:

```
dolt init --name "{{name}}" --email "{{email}}" --date
"{{2021-12-31T00:00:00}}" {{[-b|--initial-branch]}}
"{{branch_name}}"
```

dolt merge

Join two or more development histories together.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-merge.

- Incorporate changes from the named commits into the current branch:

```
dolt merge {{branch_name}}
```

- Incorporate changes from the named commits into the current branch without updating the commit history:

```
dolt merge --squash {{branch_name}}
```

- Merge a branch and create a merge commit even when the merge resolves as a fast-forward:

```
dolt merge --no-ff {{branch_name}}
```

- Merge a branch and create a merge commit with a specific commit message:

```
dolt merge --no-ff {{[-m|--message]}}
"{{message}}" {{branch_name}}
```

- Abort the current conflict resolution process:

```
dolt merge --abort
```

dolt sql

Run an SQL query. Multiple SQL statements must be separated by semicolons.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-sql.

- Run a single query:

```
dolt sql {{[-q|--query]}} "{{INSERT INTO t values (1, 3);}}"
```

- List all saved queries:

```
dolt sql {{[-l|--list-saved]}}
```

dolt status

Display the status of the database session.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-status.

- Display the status:

```
dolt status
```

dolt version

Display the current dolt CLI version.

More information: https://docs.dolthub.com/cli-reference/cli#dolt-version.

- Display version:

```
dolt version
```

dolt

An SQL database that you can fork, clone, branch, merge, push and pull just like a Git repository.

Some subcommands such as **`commit`** have their own usage documentation.

More information: https://docs.dolthub.com/cli-reference/cli.

- Execute a `dolt` subcommand:

  ```
  dolt {{subcommand}}
  ```

- List available subcommands:

  ```
  dolt help
  ```

done

This shell keyword is used with **`for`**, **`while`**, **`select`**, and **`until`** to mark the end of a loop.

- View documentation for the `for` keyword:

  ```
  tldr for
  ```

- View documentation for the `while` keyword:

  ```
  tldr while
  ```

- View documentation for the `select` keyword:

  ```
  tldr select
  ```

- View documentation for the `until` keyword:

  ```
  tldr until
  ```

DOOM

Classic open-source boomer shooter featuring modding and multiplayer.

`zandronum-server` can be replaced with almost any sourceport.

More information: https://doomwiki.org/wiki/Source_port_parameters.

- Host co-op multiplayer:

```
{{zandronum-server}} -iwad {{wad}} +map {{map}} -host
{{players}}
```

- Host deathmatch multiplayer:

```
{{zandronum-server}} -iwad {{wad}} +map {{map}} -host
{{players}} -deathmatch
```

doppler projects

Manage Doppler Projects.

More information: https://docs.doppler.com/docs/cli.

- Get all projects:

```
doppler projects
```

- Get info for a project:

```
doppler projects get {{name|project_id}}
```

- Create a project:

```
doppler projects create {{name}}
```

- Update a project's name and description:

```
doppler projects update {{name|project_id}} --name
"{{new_name}}" --description "{{new_description}}"
```

- Delete a project:

```
doppler projects delete {{name|project_id}}
```

doppler run

Run a command with Doppler secrets injected into the environment.

More information: https://docs.doppler.com/docs/cli#run-a-command-with-secrets-populated-in-environment.

- Run a command:

```
doppler run --command {{command}}
```

- Run multiple commands:

```
doppler run --command {{command1 && command2}}
```

- Run a script:

```
doppler run {{path/to/command.sh}}
```

- Run command with specified project and config:

```
doppler run -p {{project_name}} -c {{config_name}} --
{{command}}
```

- Automatically restart process when secrets change:

```
doppler run --watch {{command}}
```

doppler secrets

Manage your Doppler project's secrets.

More information: https://docs.doppler.com/docs/accessing-secrets.

- Get all secrets:

```
doppler secrets
```

- Get value(s) of one or more secrets:

```
doppler secrets get {{secrets}}
```

- Upload a secrets file:

```
doppler secrets upload {{path/to/file.env}}
```

- Delete value(s) of one or more secrets:

```
doppler secrets delete {{secrets}}
```

- Download secrets as `.env`:

```
doppler secrets download --format=env --no-file > {{path/
to/.env}}
```

doppler

Manage environment variables across different environments using Doppler.

Some subcommands such as **`run`** and **`secrets`** have their own usage documentation.

More information: https://docs.doppler.com/docs/cli.

- Setup Doppler CLI in the current directory:

```
doppler setup
```

- Setup Doppler project and configuration in current directory:

```
doppler setup
```

- Run a command with secrets injected into the environment:

```
doppler run --command {{command}}
```

- View your project list:

```
doppler projects
```

- View your secrets for current project:

```
doppler secrets
```

- Open Doppler dashboard in browser:

```
doppler open
```

dosbox

MS-DOS emulator to run legacy DOS applications and games.

More information: https://www.dosbox.com/wiki/Usage.

- Start DOSBox with default settings:

```
dosbox
```

- Run a DOS executable located at a specific path:

```
dosbox {{path/to/executable.exe}}
```

- Mount a folder as C: and run an executable:

```
dosbox {{path/to/executable.exe}} -c "MOUNT C {{path/to/folder}}"
```

- Start DOSBox in fullscreen mode:

```
dosbox -fullscreen
```

- Exit DOSBox automatically after running a program:

```
dosbox {{path/to/executable.exe}} -exit
```

dot

Render an image of a **linear directed** network graph from a **graphviz** file.

Layouts: **dot**, **neato**, **twopi**, **circo**, **fdp**, **sfdp**, **osage** & **patchwork**.

More information: https://graphviz.org/doc/info/command.html.

- Render a PNG image with a filename based on the input filename and output format (uppercase -O):

```
dot -T {{png}} -O {{path/to/input.gv}}
```

- Render a SVG image with the specified output filename (lowercase -o):

```
dot -T {{svg}} -o {{path/to/image.svg}} {{path/to/input.gv}}
```

- Render the output in PS, PDF, SVG, Fig, PNG, GIF, JPEG, JSON, or DOT format:

```
dot -T {{format}} -O {{path/to/input.gv}}
```

- Render a GIF image using `stdin` and `stdout`:

```
echo "{{digraph {this -> that} }}" | dot -T {{gif}} > {{path/to/image.gif}}
```

- Display help:

```
dot -?
```

dotenvx

A better **`dotenv`**, from the creator of **`dotenv`**.

More information: https://dotenvx.com/docs.

- Run a command with environment variables from a `.env` file:

```
dotenvx run -- {{command}}
```

- Run a command with environment variables from a specific `.env` file:

```
dotenvx run -f {{path/to/file.env}} -- {{command}}
```

- Set an environment variable with encryption:

```
dotenvx set {{key}} {{value}}
```

- Set an environment variable without encryption:

```
dotenvx set {{key}} {{value}} --plain
```

- Return environment variables defined in a `.env` file:

```
dotenvx get
```

- Return the value of an environment variable defined in a `.env` file:

```
dotenvx get {{key}}
```

- Return all environment variables from `.env` files and OS:

```
dotenvx get --all
```

dotnet add package

Add or update a .NET package reference in a project file.

More information: https://learn.microsoft.com/dotnet/core/tools/dotnet-add-package.

- Add a package to the project in the current directory:

```
dotnet add package {{package}}
```

- Add a package to a specific project:

```
dotnet add {{path/to/file.csproj}} package {{package}}
```

- Add a specific version of a package to the project:

```
dotnet add package {{package}} {{[-v|--version]}} {{1.0.0}}
```

- Add a package using a specific NuGet source:

```
dotnet add package {{package}} {{[-s|--source]}} {{https://api.nuget.org/v3/index.json}}
```

- Add a package only when targeting a specific framework:

```
dotnet add package {{package}} {{[-f|--framework]}} {{net7.0}}
```

- Add and specify the directory where to restore packages (`~/.nuget/packages` by default):

```
dotnet add package {{package}} --package-directory {{path/to/directory}}
```

dotnet add reference

Add .NET project-to-project references.

More information: https://learn.microsoft.com/dotnet/core/tools/dotnet-add-reference.

- Add a reference to the project in the current directory:

```
dotnet add reference {{path/to/reference.csproj}}
```

- Add multiple references to the project in the current directory:

```
dotnet add reference {{path/to/reference1.csproj path/to/reference2.csproj ...}}
```

- Add a reference to the specific project:

```
dotnet add {{path/to/project.csproj}} reference {{path/to/reference.csproj}}
```

- Add multiple references to the specific project:

```
dotnet add {{path/to/project.csproj}} reference {{path/to/reference1.csproj path/to/reference2.csproj ...}}
```

dotnet build

Builds a .NET application and its dependencies.

More information: https://learn.microsoft.com/dotnet/core/tools/dotnet-build.

- Compile the project or solution in the current directory:

```
dotnet build
```

- Compile a .NET project or solution in debug mode:

```
dotnet build {{path/to/project_or_solution}}
```

- Compile in release mode:

```
dotnet build {{[-c|--configuration]}} {{Release}}
```

- Compile without restoring dependencies:

```
dotnet build --no-restore
```

- Compile with a specific verbosity level:

```
dotnet build {{[-v|--verbosity]}} {{quiet|minimal|normal|
detailed|diagnostic}}
```

- Compile for a specific runtime:

```
dotnet build {{[-r|--runtime]}} {{runtime_identifier}}
```

- Specify the output directory:

```
dotnet build {{[-o|--output]}} {{path/to/directory}}
```

dotnet ef

Perform design-time development tasks for Entity Framework Core.

More information: https://learn.microsoft.com/ef/core/cli/dotnet.

- Update the database to a specified migration:

```
dotnet ef database update {{migration}}
```

- Drop the database:

```
dotnet ef database drop
```

- List available `DbContext` types:

```
dotnet ef dbcontext list
```

- Generate code for a `DbContext` and entity types for a database:

```
dotnet ef dbcontext scaffold {{connection_string}}
{{provider}}
```

- Add a new migration:

```
dotnet ef migrations add {{name}}
```

- Remove the last migration, rolling back the code changes that were done for the latest migration:

```
dotnet ef migrations remove
```

- List available migrations:

```
dotnet ef migrations list
```

- Generate an SQL script from migrations range:

```
dotnet ef migrations script {{from_migration}}
{{to_migration}}
```

dotnet publish

Publish a .NET application and its dependencies to a directory for deployment to a hosting system.

More information: https://learn.microsoft.com/dotnet/core/tools/dotnet-publish.

- Compile a .NET project in release mode:

```
dotnet publish {{[-c|--configuration]}} Release {{path/to/project_file}}
```

- Publish the .NET Core runtime with your application for the specified runtime:

```
dotnet publish {{[-sc|--self-contained]}} true {{[-r|--runtime]}} {{runtime_identifier}} {{path/to/project_file}}
```

- Package the application into a platform-specific single-file executable:

```
dotnet publish {{[-r|--runtime]}} {{runtime_identifier}} -p:PublishSingleFile=true {{path/to/project_file}}
```

- Trim unused libraries to reduce the deployment size of an application:

```
dotnet publish {{[-sc|--self-contained]}} true {{[-r|--runtime]}} {{runtime_identifier}} -p:PublishTrimmed=true {{path/to/project_file}}
```

- Compile a .NET project without restoring dependencies:

```
dotnet publish --no-restore {{path/to/project_file}}
```

- Specify the output directory:

```
dotnet publish {{[-o|--output]}} {{path/to/directory}} {{path/to/project_file}}
```

dotnet restore

Restores the dependencies and tools of a .NET project.

More information: https://learn.microsoft.com/dotnet/core/tools/dotnet-restore.

- Restore dependencies for a .NET project or solution in the current directory:

```
dotnet restore
```

- Restore dependencies for a .NET project or solution in a specific location:

```
dotnet restore {{path/to/project_or_solution}}
```

- Restore dependencies without caching the HTTP requests:

```
dotnet restore --no-cache
```

- Force all dependencies to be resolved even if the last restore was successful:

```
dotnet restore --force
```

- Restore dependencies using package source failures as warnings:

```
dotnet restore --ignore-failed-sources
```

- Restore dependencies with a specific verbosity level:

```
dotnet restore {{[-v|--verbosity]}} {{quiet|minimal|normal|
detailed|diagnostic}}
```

dotnet run

Run a .NET application without explicit compile or launch commands.

More information: https://learn.microsoft.com/dotnet/core/tools/dotnet-run.

- Run the project in the current directory:

```
dotnet run
```

- Run a specific project:

```
dotnet run --project {{path/to/file.csproj}}
```

- Run the project with specific arguments:

```
dotnet run -- {{arg1=foo arg2=bar ...}}
```

- Run the project using a target framework moniker:

```
dotnet run {{[-f|--framework]}} {{net7.0}}
```

- Specify architecture and OS, available since .NET 6 (Don't use `--runtime` with these options):

```
dotnet run {{[-a|--arch]}} {{x86|x64|arm|arm64}} --os {{win|win7|osx|linux|ios|android}}
```

dotnet test

Execute tests for a .NET application.

Note: View https://learn.microsoft.com/en-us/dotnet/core/testing/selective-unit-tests for supported filter expressions.

More information: https://learn.microsoft.com/dotnet/core/tools/dotnet-test.

- Execute tests for a .NET project/solution in the current directory:

  ```
  dotnet test
  ```

- Execute tests for a .NET project/solution in a specific location:

  ```
  dotnet test {{path/to/project_or_solution}}
  ```

- Execute tests matching the given filter expression:

  ```
  dotnet test --filter {{Name~TestMethod1}}
  ```

dotnet tool

Manage .NET tools and search published tools in NuGet.

More information: https://learn.microsoft.com/dotnet/core/tools/global-tools.

- Install a global tool (don't use `--global` for local tools):

```
dotnet tool install {{[-g|--global]}} {{dotnetsay}}
```

- Install tools defined in the local tool manifest:

```
dotnet tool restore
```

- Update a specific global tool (don't use `--global` for local tools):

```
dotnet tool update {{[-g|--global]}} {{tool_name}}
```

- Uninstall a global tool (don't use `--global` for local tools):

```
dotnet tool uninstall {{[-g|--global]}} {{tool_name}}
```

- List installed global tools (don't use `--global` for local tools):

```
dotnet tool list {{[-g|--global]}}
```

- Search tools in NuGet:

```
dotnet tool search {{search_term}}
```

- Display help:

```
dotnet tool {{[-h|--help]}}
```

dotnet

Cross platform .NET tools for .NET Core.

Some subcommands such as **`build`** have their own usage documentation.

More information: https://learn.microsoft.com/dotnet/core/tools.

- Initialize a new .NET project:

```
dotnet new {{template_short_name}}
```

- Restore NuGet packages:

```
dotnet restore
```

- Build and execute the .NET project in the current directory:

```
dotnet run
```

- Run a packaged dotnet application (only needs the runtime, the rest of the commands require the .NET Core SDK installed):

```
dotnet {{path/to/application.dll}}
```

dotty

A customizable interactive graph editor for the X Window System.

Note: This tool has been deprecated, but this entry is retained to aid anyone still using it.

More information: https://graphviz.org/pdf/dotty.1.pdf.

- Open a graph file (`.gv` or `.dot`) in the Dotty editor:

```
dotty {{path/to/graph.gv}}
```

- Start Dotty with the [V]ersion information displayed:

```
dotty -V
```

- Set the [l]ayout [m]ode to synchronous or asynchronous:

```
dotty -lm {{sync|async}} {{path/to/graph.gv}}
```

- Set the m[e]ssage verbosity [l]evel (`0` for minimal, `1` for detailed):

```
dotty -el {{0|1}} {{path/to/graph.gv}}
```

doxygen

A documentation system for various programming languages.

More information: https://manned.org/doxygen.

- Generate documentation, using `Doxyfile` as configuration file if it exists:

```
doxygen
```

- Generate a default template configuration file `Doxyfile`:

```
doxygen -g
```

- Generate a template configuration file:

```
doxygen -g {{path/to/config_file}}
```

- Generate documentation using an existing configuration file:

```
doxygen {{path/to/config_file}}
```

drill

Perform various DNS queries.

More information: https://manned.org/drill.

- Lookup the IP(s) associated with a hostname (A records):

```
drill {{example.com}}
```

- Lookup the mail server(s) associated with a given domain name (MX record):

```
drill mx {{example.com}}
```

- Get all types of records for a given domain name:

```
drill any {{example.com}}
```

- Specify an alternate DNS server to query:

```
drill {{example.com}} @{{8.8.8.8}}
```

- Perform a reverse DNS lookup on an IP address (PTR record):

```
drill -x {{8.8.8.8}}
```

- Perform DNSSEC trace from root servers down to a domain name:

```
drill -TD {{example.com}}
```

- Show DNSKEY record(s) for a domain name:

```
drill -s dnskey {{example.com}}
```

dropdb

Remove a PostgreSQL database.

A simple wrapper around the SQL command **`DROP DATABASE`**.

More information: https://www.postgresql.org/docs/current/app-dropdb.html.

- Destroy a database:

```
dropdb {{database_name}}
```

- Request a verification prompt before any destructive actions:

```
dropdb {{[-i|--interactive]}} {{database_name}}
```

- Connect with a specific username and destroy a database:

```
dropdb {{[-U|--username]}} {{username}} {{database_name}}
```

- Force a password prompt before connecting to the database:

```
dropdb {{[-W|--password]}} {{database_name}}
```

- Suppress a password prompt before connecting to the database:

```
dropdb {{[-w|--no-password]}} {{database_name}}
```

- Specify the server host name:

```
dropdb {{[-h|--host]}} {{host}} {{database_name}}
```

- Specify the server port:

```
dropdb {{[-p|--port]}} {{port}} {{database_name}}
```

- Attempt to terminate existing connections before destroying the database:

```
dropdb {{[-f|--force]}} {{database_name}}
```

dropuser

Remove an existing PostgreSQL user (role).

More information: https://www.postgresql.org/docs/current/app-dropuser.html.

- Prompt for confirmation and the username before user removal:

```
dropuser {{[-i|--interactive]}}
```

- Remove user instantly:

```
dropuser {{username}}
```

- No error if the user to be removed doesn't exist:

```
dropuser --if-exists {{username}}
```

- Remove a user on the server with address 127.0.0.1 on port 4321:

```
dropuser {{[-h|--host]}} 127.0.0.1 {{[-p|--port]}} 4321
{{username}}
```

- Remove a user on the server with address 127.0.0.1 on port 4321 as user "admin" :

```
dropuser {{[-U|--username]}} admin {{[-h|--host]}} 127.0.0.1
{{[-p|--port]}} 4321 {{username}}
```

drupal-check

Check Drupal PHP code for deprecations.

More information: https://github.com/mglaman/drupal-check#usage.

- Check the code in a specific directory for deprecations:

```
drupal-check {{path/to/directory}}
```

- Check the code excluding a comma-separated list of directories:

```
drupal-check {{[-e|--exclude-dir]}} {{path/to/
excluded_directory}},{{path/to/excluded_files/*.php}} {{path/
to/directory}}
```

- Don't show a progress bar:

```
drupal-check --no-progress {{path/to/directory}}
```

- Perform static analysis to detect bad coding practices:

```
drupal-check {{[-a|--analysis]}} {{path/to/directory}}
```

drupal

Generate boilerplate code, interact with and debug Drupal projects.

Some subcommands such as **check** have their own usage documentation.

More information: https://drupalconsole.com/.

- Install a module:

```
drupal module:install {{module_name}}
```

- Uninstall a module:

```
drupal module:uninstall {{module_name}}
```

- Clear all caches:

```
drupal cache:rebuild
```

- View current Drupal installation status:

```
drupal site:status
```

drush

Shell and scripting interface for Drupal.

More information: https://www.drush.org/latest/usage/.

- Enable module "foo":

```
drush en {{foo}}
```

- Uninstall module "foo":

```
drush pmu {{foo}}
```

- Clear all caches:

```
drush cr
```

- Clear CSS and JavaScript caches:

```
drush cc css-js
```

dtc

The Device Tree Compiler, a tool for recompiling device trees between formats.

More information: https://github.com/dgibson/dtc.

- Compile a device tree source `.dts` file into a binary device tree blob `.dtb` file:

```
dtc -I dts -O dtb -o {{path/to/output_file.dtb}} {{path/to/input_file.dts}}
```

- Compile a device tree source `.dts` file into a binary device tree blob overlay `.dtbo` file:

```
dtc -@ -I dts -O dtb -o {{path/to/output_file.dtbo}} {{path/to/input_file.dts}}
```

- Decompile a device tree blob `.dtb` file into a readable device tree source `.dts` file:

```
dtc -I dtb -O dts -o {{path/to/output_file.dts}} {{path/to/input_file.dtb}}
```

- Decompile the current device tree from the system into a readable device tree source `.dts` file:

```
dtc -I fs -O dts /proc/device-tree
```

dtrx

"Do The Right eXtraction" - extract any archive into new directory by guessing the tool from extension.

More information: https://github.com/dtrx-py/dtrx.

- Extract archive, guessing the extraction tool from extension:

```
dtrx {{path/to/file}}
```

- Extract archive, overwrite any existing target output:

```
dtrx --overwrite {{path/to/file}}
```

- Extract archive, put everything into current directory:

```
dtrx --flat {{path/to/file}}
```

du

Disk usage: estimate and summarize file and directory space usage.

More information: https://www.gnu.org/software/coreutils/manual/html_node/du-invocation.html.

- List the sizes of a directory and any subdirectories, in the given unit (B/KiB/MiB):

```
du -{{b|k|m}} {{path/to/directory}}
```

- List the sizes of a directory and any subdirectories, in human-readable form (i.e. auto-selecting the appropriate unit for each size):

```
du {{[-h|--human-readable]}} {{path/to/directory}}
```

- Show the size of a single directory, in human-readable units:

```
du {{[-sh|--summarize --human-readable]}} {{path/to/directory}}
```

- List the human-readable sizes of a directory and of all the files and directories within it:

```
du {{[-ah|--all --human-readable]}} {{path/to/directory}}
```

- List the human-readable sizes of a directory and any subdirectories, up to `n` levels deep:

```
du {{[-h|--human-readable]}} {{[-d|--max-depth]}} {{n}} {{path/to/directory}}
```

- List the human-readable size of all `.jpg` files in current directory, and show a cumulative total at the end:

```
du {{[-ch|--total --human-readable]}} *.jpg
```

- List all files and directories (including hidden ones) above a certain threshold size (useful for investigating what is actually taking up the space):

```
du {{[-ah|--all --human-readable]}} {{[-t|--threshold]}} {{1G|1024M|1048576K}} .[^.]* *
```

dua

Dua (Disk Usage Analyzer): get the disk space usage of a directory.

More information: https://github.com/Byron/dua-cli.

- Analyze specific directory:

```
dua {{path/to/directory}}
```

- Display apparent size instead of disk usage:

```
dua --apparent-size
```

- Count hard-linked files each time they are seen:

```
dua --count-hard-links
```

- Aggregate the consumed space of one or more directories or files:

```
dua aggregate
```

- Launch the terminal user interface:

```
dua interactive
```

- Format printing byte counts:

```
dua --format {{metric|binary|bytes|GB|GiB|MB|MiB}}
```

- Use a specific number of threads (defaults to the process number of threads):

```
dua --threads {{count}}
```

dub

Package manager for D packages.

More information: https://dub.pm/commandline.

- Interactively create a new D project:

```
dub init {{project_name}}
```

- Non-interactively create a new D project:

```
dub init {{project_name}} {{[-n|--non-interactive]}}
```

- Build and run a D project:

```
dub
```

- Install dependencies specified in a D project's `dub.json` or `dub.sdl` file:

```
dub fetch
```

- Update the dependencies in a D project:

```
dub upgrade
```

- Display help:

```
dub {{[-h|--help]}}
```

duc

A collection of tools for indexing, inspecting, and visualizing disk usage.

Duc maintains a database of accumulated sizes of directories of the file system, allowing queries in this database, or creating fancy graphs to show where data is.

More information: https://duc.zevv.nl/.

- Index the `/usr` directory, writing to the default database location `~/.duc.db`:

```
duc index {{/usr}}
```

- List all files and directories under `/usr/local`, showing relative file sizes in a graph:

```
duc ls {{[-Fg|--classify --graph]}} {{/usr/local}}
```

- List all files and directories under `/usr/local` using treeview recursively:

```
duc ls {{[-Fg|--classify --graph]}} {{[-R|--recursive]}} {{/usr/local}}
```

- Start the graphical interface to explore the file system using sunburst graphs:

```
duc gui {{/usr}}
```

- Run the ncurses console interface to explore the file system:

```
duc ui {{/usr}}
```

- Dump database info:

```
duc info
```

duckdb

Client for DuckDB, an in-process analytical SQL engine.

More information: https://duckdb.org.

- Start an interactive shell with a transient in-memory database:

```
duckdb
```

- Start an interactive shell on a database file. If the file does not exist, a new database is created:

```
duckdb {{path/to/dbfile}}
```

- Query a CSV, JSON, or Parquet file using SQL:

```
duckdb -c "{{SELECT * FROM 'data_source.[csv|csv.gz|json|json.gz|parquet]'}}"
```

- Directly query a CSV, JSON, or Parquet file using the `file` view:

```
duckdb {{data_source.[csv|csv.gz|json|json.gz|parquet]}} -c "{{ SELECT * FROM file }}"
```

- Run an SQL script:

```
duckdb -f {{path/to/script.sql}}
```

- Run query on database file and keep the interactive shell open:

```
duckdb {{path/to/dbfile}} -cmd "{{SELECT DISTINCT * FROM tbl}}"
```

- Read CSV from `stdin` and write CSV to `stdout`:

```
cat {{path/to/source.csv}} | duckdb -c "{{COPY (FROM read_csv('/dev/stdin')) TO '/dev/stdout' WITH (FORMAT CSV, HEADER)}}"
```

- Start the DuckDB UI, a web interface with notebooks:

```
duckdb -ui
```

duf

Disk Usage/Free Utility.

More information: https://github.com/muesli/duf.

- List accessible devices:

```
duf
```

- List everything (such as pseudo, duplicate or inaccessible file systems):

```
duf --all
```

- Only show specified devices or mount points:

```
duf {{path/to/directory1 path/to/directory2 ...}}
```

- Sort the output by a specified criteria:

```
duf --sort {{size|used|avail|usage}}
```

- Show or hide specific filesystems:

```
duf --{{only-fs|hide-fs}} {{tmpfs|vfat|ext4|xfs}}
```

- Sort the output by key:

```
duf --sort {{mountpoint|size|used|avail|usage|inodes|
inodes_used|inodes_avail|inodes_usage|type|filesystem}}
```

- Change the theme (if `duf` fails to use the right theme):

```
duf --theme {{dark|light}}
```

dumpcap

A network traffic dump tool.

More information: https://www.wireshark.org/docs/man-pages/dumpcap.html.

- Display available interfaces:

```
dumpcap {{[-D|--list-interfaces]}}
```

- Capture packets on a specific interface:

```
dumpcap {{[-i|--interface]}} {{1}}
```

- Capture packets to a specific location:

```
dumpcap {{[-i|--interface]}} {{1}} -w {{path/to/
output_file.pcapng}}
```

- Write to a ring buffer with a specific max file limit of a specific size:

```
dumpcap {{[-i|--interface]}} {{1}} -w {{path/to/
output_file.pcapng}} {{[-b|--ring-buffer]}} filesize:
{{500000}} {{[-b|--ring-buffer]}} files:{{10}}
```

dune

A build system for OCaml programs.

More information: https://dune.build.

- Build all targets:

  ```
  dune build
  ```

- Clean up the workspace:

  ```
  dune clean
  ```

- Run all tests:

  ```
  dune runtest
  ```

- Start the utop REPL with compiled modules automatically loaded into it, to remove the need to load them by hand:

  ```
  dune utop
  ```

dunst

A lightweight and customizable notification daemon for X11 and Wayland.

If not started manually, D-Bus will automatically start **`dunst`** when a notification is sent.

More information: https://dunst-project.org/documentation/dunst.

- Start `dunst`:

```
dunst
```

- Display a notification on startup:

```
dunst -startup_notification
```

- Print incoming notifications to `stdout`:

```
dunst -print
```

- Use the specified configuration file (default: `$XDG_CONFIG_HOME/dunst/dunstrc`):

```
dunst {{[-conf|-config]}} {{path/to/file}}
```

dunstctl

Control the **`dunst`** notification daemon.

More information: https://dunst-project.org/documentation/dunstctl.

- Pause/Unpause/Toggle desktop notifications:

```
dunstctl set-paused {{true|false|toggle}}
```

- Close all notifications:

```
dunstctl close-all
```

- Delete all notifications from history:

```
dunstctl history-clear
```

- Display the latest notification from history:

```
dunstctl history-pop
```

- Reload the configuration file:

```
dunstctl reload
```

- Display help:

```
dunstctl {{[-h|--help]}}
```

dunstify

A notification tool that is an extension of **`notify-send`**, but has more features based around **`dunst`**.

Accepts all options of **`notify-send`**.

More information: https://dunst-project.org/documentation/dunstify.

- Show a notification with a given title and message:

```
dunstify "{{Title}}" "{{Message}}"
```

- Show a notification with the specified urgency:

```
dunstify "{{Title}}" "{{Message}}" {{[-u|--urgency]}} {{low|normal|critical}}
```

- Specify a message ID (overwrites any previous messages with the same ID):

```
dunstify "{{Title}}" "{{Message}}" {{[-r|--replace]}} {{123}}
```

- Display help:

```
dunstify {{[-?|--help]}}
```

duplicacy

A lock-free deduplication cloud backup tool.

More information: https://github.com/gilbertchen/duplicacy/wiki.

- Use current directory as the repository, initialize a SFTP storage and encrypt the storage with a password:

```
duplicacy init {{[-e|-encrypt]}} {{snapshot_id}} {{sftp://user@192.168.2.100/path/to/storage}}/
```

- Save a snapshot of the repository to the default storage:

```
duplicacy backup
```

- List snapshots of current repository:

```
duplicacy list
```

- Restore the repository to a previously saved snapshot:

```
duplicacy restore -r {{revision}}
```

- Check the integrity of snapshots:

```
duplicacy check
```

- Add another storage to be used for the existing repository:

```
duplicacy add {{storage_name}} {{snapshot_id}} {{storage_url}}
```

- Prune a specific revision of snapshot:

```
duplicacy prune -r {{revision}}
```

- Prune revisions, keeping one revision every `n` days for all revisions older than `m` days:

```
duplicacy prune -keep {{n:m}}
```

duplicity

Create incremental, compressed, encrypted and versioned backups.

Can also upload the backups to a variety of backend services.

It is worth mentioning that depending on the version, some options may not be available (e.g. **`--gio`** in 2.0.0).

More information: https://duplicity.gitlab.io/stable/duplicity.1.html#name.

- Backup a directory via FTPS to a remote machine, encrypting it with a password:

```
FTP_PASSWORD={{ftp_login_password}}
PASSPHRASE={{encryption_password}} duplicity {{path/to/
source_directory}} {{ftps://user@hostname/path/to/
target_directory}}/
```

- Backup a directory to Amazon S3, doing a full backup every month:

```
duplicity --full-if-older-than {{1M}} s3://{{bucket_name[/
prefix]}}
```

- Delete versions older than 1 year from a backup stored on a WebDAV share:

```
FTP_PASSWORD={{webdav_login_password}} duplicity remove-
older-than {{1Y}} --force {{webdav[s]://user@hostname[:port]/
some_directory}}
```

- List the available backups:

```
duplicity collection-status "file://{{absolute/path/to/
backup_directory}}"
```

- List the files in a backup stored on a remote machine, via SSH:

```
duplicity list-current-files {{[-t|--time]}} {{YYYY-MM-DD}}
scp://{{user@hostname}}/{{path/to/backup_directory}}
```

- Restore a subdirectory from a GnuPG-encrypted local backup to a given location:

```
PASSPHRASE={{gpg_key_password}} duplicity restore --encrypt-
key {{gpg_key_id}} --path-to-restore {{relative/path/
restoredirectory}} file://{{absolute/path/to/
backup_directory}} {{path/to/directory_to_restore_to}}
```

dust

Give an instant overview of which directories are using disk space.

See also: **du**, **ncdu**.

More information: https://github.com/bootandy/dust#usage.

- Display information for the current directory:

```
dust
```

- Display information about one or more directories:

```
dust {{path/to/directory1 path/to/directory2 ...}}
```

- Display 30 directories (defaults to 21):

```
dust {{[-n|--number-of-lines]}} 30
```

- Display information for the current directory, up to 3 levels deep:

```
dust {{[-d|--depth]}} 3
```

- Display the largest directories at the top in descending order:

```
dust {{[-r|--reverse]}}
```

- Ignore a file or directory:

```
dust {{[-X|--ignore-directory]}} {{path/to/
file_or_directory}}
```

- Do not display percent bars and percentages:

```
dust {{[-b|--no-percent-bars]}}
```

dvc add

Add changed files to the index.

More information: https://dvc.org/doc/command-reference/add.

- Add a single target file to the index:

```
dvc add {{path/to/file}}
```

- Add a target directory to the index:

```
dvc add {{path/to/directory}}
```

- Recursively add all the files in a given target directory:

```
dvc add --recursive {{path/to/directory}}
```

- Add a target file with a custom `.dvc` filename:

```
dvc add --file {{custom_name.dvc}} {{path/to/file}}
```

dvc checkout

Checkout data files and directories from cache.

More information: https://dvc.org/doc/command-reference/checkout.

- Checkout the latest version of all target files and directories:

```
dvc checkout
```

- Checkout the latest version of a specified target:

```
dvc checkout {{target}}
```

- Checkout a specific version of a target from a different Git commit/tag/branch:

```
git checkout {{commit_hash|tag|branch}} {{target}} && dvc checkout {{target}}
```

dvc commit

Record changes to DVC-tracked files in the project.

More information: https://dvc.org/doc/command-reference/commit.

- Commit changes to all DVC-tracked files and directories:

```
dvc commit
```

- Commit changes to a specified DVC-tracked target:

```
dvc commit {{target}}
```

- Recursively commit all DVC-tracked files in a directory:

```
dvc commit {{[-R|--recursive]}} {{path/to/directory}}
```

dvc config

Low level command to manage custom configuration options for dvc repositories.

These configurations can be on project, local, global, or system level.

More information: https://dvc.org/doc/command-reference/config.

- Get the name of the default remote:

```
dvc config core.remote
```

- Set the project's default remote:

```
dvc config core.remote {{remote_name}}
```

- Unset the project's default remote:

```
dvc config {{[-u|--unset]}} core.remote
```

- Get the configuration value for a specified key for the current project:

```
dvc config {{key}}
```

- Set the configuration value for a key on a project level:

```
dvc config {{key}} {{value}}
```

- Unset a project level configuration value for a given key:

```
dvc config {{[-u|--unset]}} {{key}}
```

- Set a local, global, or system level configuration value:

```
dvc config --{{local|global|system}} {{key}} {{value}}
```

dvc dag

Visualize the pipeline(s) defined in **`dvc.yaml`**.

More information: https://dvc.org/doc/command-reference/dag.

- Visualize the entire pipeline:

```
dvc dag
```

- Visualize the pipeline stages up to a specified target stage:

```
dvc dag {{target}}
```

- Export the pipeline in the dot format:

```
dvc dag --dot > {{path/to/pipeline.dot}}
```

dvc destroy

Remove all DVC files and directories from a DVC project.

More information: https://dvc.org/doc/command-reference/destroy.

- Destroy the current project:

```
dvc destroy
```

- Force destroy the current project:

```
dvc destroy {{[-f|--force]}}
```

dvc diff

Show changes in DVC tracked file and directories.

More information: https://dvc.org/doc/command-reference/diff.

- Compare DVC tracked files from different Git commits, tags, and branches w.r.t the current workspace:

```
dvc diff {{commit_hash/tag/branch}}
```

- Compare the changes in DVC tracked files from 1 Git commit to another:

```
dvc diff {{revision1}} {{revision2}}
```

- Compare DVC tracked files, along with their latest hash:

```
dvc diff --show-hash {{commit}}
```

- Compare DVC tracked files, displaying the output as JSON:

```
dvc diff --show-json --show-hash {{commit}}
```

- Compare DVC tracked files, displaying the output as Markdown:

```
dvc diff --show-md --show-hash {{commit}}
```

dvc fetch

Download DVC tracked files and directories from a remote repository.

More information: https://dvc.org/doc/command-reference/fetch.

- Fetch the latest changes from the default remote upstream repository (if set):

```
dvc fetch
```

- Fetch changes from a specific remote upstream repository:

```
dvc fetch {{[-r|--remote]}} {{remote_name}}
```

- Fetch the latest changes for a specific target/s:

```
dvc fetch {{target/s}}
```

- Fetch changes for all branch and tags:

```
dvc fetch {{[-a|--all-branches]}} {{[-T|--all-tags]}}
```

- Fetch changes for all commits:

```
dvc fetch {{[-A|--all-commits]}}
```

dvc freeze

Freeze stages in the DVC pipeline.

This prevents DVC from tracking changes in stage dependencies and re-execution until unfreeze.

See also: **`dvs unfreeze`**.

More information: https://dvc.org/doc/command-reference/freeze.

- Freeze one or more specified stages:

```
dvc freeze {{stage_name1 stage_name2 ...}}
```

dvc gc

Remove unused files and directories from the cache or remote storage.

More information: https://dvc.org/doc/command-reference/gc.

- Garbage collect from the cache, keeping only versions referenced by the current workspace:

```
dvc gc {{[-w|--workspace]}}
```

- Garbage collect from the cache, keeping only versions referenced by branch, tags, and commits:

```
dvc gc {{[-a|--all-branches]}} {{[-T|--all-tags]}} {{[-a|--all-commits]}}
```

- Garbage collect from the cache, including the default cloud remote storage (if set):

```
dvc gc {{[-a|--all-commits]}} {{[-c|--cloud]}}
```

- Garbage collect from the cache, including a specific cloud remote storage:

```
dvc gc {{[-a|--all-commits]}} {{[-c|--cloud]}} {{[-r|--remote]}} {{remote_name}}
```

dvc init

Initialize a new local DVC repository.

More information: https://dvc.org/doc/command-reference/init.

- Initialize a new local repository:

```
dvc init
```

- Initialize DVC without Git:

```
dvc init --no-scm
```

- Initialize DVC in a subdirectory:

```
cd {{path/to/subdirectory}} && dvc init --sudir
```

dvc unfreeze

Unfreeze stages in the DVC pipeline.

This allows DVC to start tracking changes in stage dependencies again after they were frozen.

See also: **`dvc freeze`**.

More information: https://dvc.org/doc/command-reference/unfreeze.

- Unfreeze one or more specified stages:

```
dvc unfreeze {{stage_name1 stage_name2 ...}}
```

dvc

Data Version Control: like **`git`** for data.

Some subcommands such as **`commit`** have their own usage documentation.

More information: https://dvc.org/doc/command-reference.

- Execute a DVC subcommand:

  ```
  dvc {{subcommand}}
  ```

- Display help:

  ```
  dvc {{[-h|--help]}}
  ```

- Display help about a specific subcommand:

  ```
  dvc {{subcommand}} {{[-h|--help]}}
  ```

- Display version:

  ```
  dvc --version
  ```

dwebp

dwebp decompresses WebP files into PNG, PAM, PPM or PGM images.

Animated WebP files are not supported.

More information: https://developers.google.com/speed/webp/docs/dwebp/.

- Convert a WebP file into a PNG file:

```
dwebp {{path/to/input.webp}} -o {{path/to/output.png}}
```

- Convert a WebP file into a specific filetype:

```
dwebp {{path/to/input.webp}} -bmp|-tiff|-pam|-ppm|-pgm|-yuv -
o {{path/to/output}}
```

- Convert a WebP file, using multi-threading if possible:

```
dwebp {{path/to/input.webp}} -o {{path/to/output.png}} -mt
```

- Convert a WebP file, but also crop and scale at the same time:

```
dwebp {{input.webp}} -o {{output.png}} -crop {{x_pos}}
{{y_pos}} {{width}} {{height}} -scale {{width}} {{height}}
```

- Convert a WebP file and flip the output:

```
dwebp {{path/to/input.webp}} -o {{path/to/output.png}} -flip
```

- Convert a WebP file and don't use in-loop filtering to speed up the decoding process:

```
dwebp {{path/to/input.webp}} -o {{path/to/output.png}} -
nofilter
```

ebook-convert

Can be used to convert e-books between common formats, e.g. PDF, EPUB and MOBI.

Part of the Calibre e-book library tool.

More information: https://manual.calibre-ebook.com/generated/en/ebook-convert.html.

- Convert an e-book into another format:

```
ebook-convert {{path/to/input_file}} {{output_file}}
```

- Convert Markdown or HTML to e-book with TOC, title and author:

```
ebook-convert {{path/to/input_file}} {{output_file}} --
level1-toc="//h:h1" --level2-toc="//h:h2" --level3-toc="//
h:h3" --title={{title}} --authors={{author}}
```

echo

Print given arguments.

See also: **`printf`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/echo-invocation.html.

- Print a text message. Note: Quotes are optional:

```
echo "{{Hello World}}"
```

- Print a message with environment variables:

```
echo "{{My path is $PATH}}"
```

- Print a message without the trailing newline:

```
echo -n "{{Hello World}}"
```

- Append a message to the file:

```
echo "{{Hello World}}" >> {{file.txt}}
```

- Enable interpretation of backslash escapes (special characters):

```
echo -e "{{Column 1\tColumn 2}}"
```

- Print the exit status of the last executed command (Note: In Windows Command Prompt and PowerShell the equivalent commands are `echo %errorlevel%` and `$lastexitcode` respectively):

```
echo $?
```

- Pass text to another program through `stdin`:

```
echo "{{Hello World}}" | {{program}}
```

ecpg

Embedded SQL preprocessor for C programs.

More information: https://www.postgresql.org/docs/current/app-ecpg.html.

- Preprocess a specific file:

```
ecpg {{input.pgc}}
```

- Preprocess from `stdin` and output to `stdout`:

```
ecpg -o - -
```

- Preprocess from `stdin` and write to a file:

```
cat input.pgc | ecpg -o output.c -
```

- Preprocess and specify an output file:

```
ecpg -o {{output.c}} {{input.pgc}}
```

- Preprocess a header file (`.pgh` extension):

```
ecpg {{input.pgh}}
```

- Preprocess in a specific compatibility mode:

```
ecpg -C {{INFORMIX|INFORMIX_SE|ORACLE}} {{input.pgc}}
```

- Preprocess with autocommit enabled:

```
ecpg -t {{input.pgc}}
```

ect

Efficient Compression Tool.

File optimizer written in C++. It supports PNG, JPEG, gzip and Zip files.

More information: https://github.com/fhanau/Efficient-Compression-Tool.

- Compress a file:

```
ect {{path/to/file.png}}
```

- Compress a file with specified compression level and multithreading (1=Fastest (Worst), 9=Slowest (Best), default is 3):

```
ect -{{9}} --mt-deflate {{path/to/file.zip}}
```

- Compress all files in a directory recursively:

```
ect -recurse {{path/to/directory}}
```

- Compress a file, keeping the original modification time:

```
ect -keep {{path/to/file.png}}
```

- Compress a file, stripping metadata:

```
ect -strip {{path/to/file.png}}
```

ed

The original Unix text editor.

See also: **awk**, **sed**.

More information: https://www.gnu.org/software/ed/manual/ed_manual.html.

- Start an interactive editor session with an empty document:

```
ed
```

- Start an interactive editor session with an empty document and a specific prompt:

```
ed {{[-p|--prompt]}} '{{> }}'
```

- Start an interactive editor session with user-friendly errors:

```
ed {{[-v|--verbose]}}
```

- Start an interactive editor session with an empty document and without diagnostics, byte counts and '!' prompt:

```
ed {{[-q|--quiet]}} {{[-s|--script]}}
```

- Start an interactive editor session without exit status change when command fails:

```
ed {{[-l|--loose-exit-status]}}
```

- Edit a specific file (this shows the byte count of the loaded file):

```
ed {{path/to/file}}
```

- Replace a string with a specific replacement for all lines:

```
,s/{{regex}}/{{replacement}}/g<Enter>
```

- Exit `ed`:

```
q<Enter>
```

edgepaint

Colorize edges of a graph layout to clarify crossing edges.

Graphviz filters: **`acyclic`**, **`bcomps`**, **`comps`**, **`edgepaint`**, **`gvcolor`**, **`gvpack`**, **`mingle`**, **`nop`**, **`sccmap`**, **`tred`**, & **`unflatten`**.

More information: https://graphviz.org/pdf/edgepaint.1.pdf.

- Colorize edges of one or more graph layouts (that already have layout information) to clarify crossing edges:

```
edgepaint {{path/to/layout1.gv path/to/layout2.gv ...}} > {{path/to/output.gv}}
```

- Colorize edges using a color scheme. (See https://graphviz.org/doc/info/colors.html#brewer):

```
edgepaint -color-scheme={{accent7}} {{path/to/layout.gv}} > {{path/to/output.gv}}
```

- Lay out a graph and colorize its edges, then convert to a PNG image:

```
dot {{path/to/input.gv}} | edgepaint | dot -T {{png}} > {{path/to/output.png}}
```

- Display help:

```
edgepaint -?
```

eget

Easily install prebuilt binaries from GitHub.

More information: https://github.com/zyedidia/eget.

- Download a prebuilt binary for the current system from a repository on GitHub:

```
eget {{zyedidia/micro}}
```

- Download from a URL:

```
eget {{https://go.dev/dl/go1.17.5.linux-amd64.tar.gz}}
```

- Specify the location to place the downloaded files:

```
eget {{zyedidia/micro}} --to={{path/to/directory}}
```

- Specify a Git tag instead of using the latest version:

```
eget {{zyedidia/micro}} --tag={{v2.0.10}}
```

- Install the latest pre-release instead of the latest stable version:

```
eget {{zyedidia/micro}} --pre-release
```

- Only download the asset, skipping extraction:

```
eget {{zyedidia/micro}} --download-only
```

- Only download if there is a newer release then the currently downloaded version:

```
eget {{zyedidia/micro}} --upgrade-only
```

egrep

Find patterns in files using extended **regex** (supports **?**, **+**, **{}**, **()**, and **|**).

More information: https://manned.org/egrep.

- Search for a pattern within a file:

```
egrep "{{search_pattern}}" {{path/to/file}}
```

- Search for a pattern within multiple files:

```
egrep "{{search_pattern}}" {{path/to/file1 path/to/
file2 ...}}
```

- Search `stdin` for a pattern:

```
cat {{path/to/file}} | egrep {{search_pattern}}
```

- Print file name and line number for each match:

```
egrep {{[-H|--with-filename]}} {{[-n|--line-number]}}
"{{search_pattern}}" {{path/to/file}}
```

- Search for a pattern in all files recursively in a directory, ignoring binary files:

```
egrep {{[-r|--recursive]}} --binary-files={{without-match}}
"{{search_pattern}}" {{path/to/directory}}
```

- Search for lines that do not match a pattern:

```
egrep {{[-v|--invert-match]}} "{{search_pattern}}" {{path/to/
file}}
```

eim

Install and manage ESP-IDF.

More information: https://docs.espressif.com/projects/idf-im-ui/en/latest/cli_commands.html.

- Install the default (latest) ESP-IDF version in the default location (`C:\esp` on Windows and `~/.espressif` on POSIX systems):

```
eim install
```

- Install a specific ESP-IDF version:

```
eim install {{[-i|--idf-versions]}} {{v5.3.2}}
```

- Run the interactive, guided installation wizard:

```
eim wizard
```

- Install a specific version to a custom path, forcing interactive mode (to prompt for choices):

```
eim install {{[-i|--idf-versions]}} {{v5.3.2}} {{[-p|--path]}} {{/opt/esp-idf}} {{[-n|--non-interactive]}} false
```

- List all currently installed ESP-IDF versions:

```
eim list
```

- Remove a specific installed ESP-IDF version:

```
eim remove {{v5.3.2}}
```

- Install in headless mode using all options defined in a TOML configuration file:

```
eim install {{[-c|--config]}} {{path/to/config.toml}}
```

- Install offline using a pre-downloaded archive file:

```
eim install --use-local-archive {{path/to/archive.zst}}
```

eksctl

The official CLI for Amazon EKS.

More information: https://eksctl.io.

- Create a basic cluster:

```
eksctl create cluster
```

- List the details about a cluster or all of the clusters:

```
eksctl get cluster --name={{name}} --region={{region}}
```

- Create a cluster passing all configuration information in a file:

```
eksctl create cluster --config-file={{path/to/file}}
```

- Create a cluster using a configuration file and skip creating nodegroups until later:

```
eksctl create cluster --config-file=<path> --without-nodegroup
```

- Delete a cluster:

```
eksctl delete cluster --name={{name}} --region={{region}}
```

- Create cluster and write cluster credentials to a file other than the default:

```
eksctl create cluster --name={{name}} --nodes={{4}} --kubeconfig={{path/to/config.yaml}}
```

- Create a cluster and prevent storing cluster credentials locally:

```
eksctl create cluster --name={{name}} --nodes={{4}} --write-kubeconfig=false
```

- Create a cluster and let `eksctl` manage cluster credentials under the `~/.kube/eksctl/clusters` directory:

```
eksctl create cluster --name={{name}} --nodes={{4}} --auto-kubeconfig
```

elasticsearch-certutil

Generate and manage SSL certificates for Elasticsearch security.

More information: https://www.elastic.co/guide/en/elasticsearch/reference/current/certutil.html.

- Generate a new Certificate Authority (CA) with default options:

```
elasticsearch-certutil ca
```

- Generate a new certificate using the built-in CA:

```
elasticsearch-certutil cert
```

- Generate certificates non-interactively and output PEM files:

```
elasticsearch-certutil cert {{[-s|--silent]}} --pem
```

- Generate HTTP certificates with the built-in CA:

```
elasticsearch-certutil http
```

- Generate transport certificates non-interactively:

```
elasticsearch-certutil transport {{[-s|--silent]}}
```

- Generate a certificate signing request (CSR):

```
elasticsearch-certutil csr
```

- Generate encrypted keystore passwords:

```
elasticsearch-certutil password
```

- Generate a keystore password with a specified value:

```
elasticsearch-certutil password --pass {{password}}
```

elasticsearch-create-enrollment-token

Create enrollment tokens for Elasticsearch nodes and Kibana instances.

More information: https://www.elastic.co/guide/en/elasticsearch/reference/current/create-enrollment-token.html.

- Create an enrollment token for adding a new Elasticsearch node:

```
elasticsearch-create-enrollment-token {{[-s|--scope]}} node
```

- Create an enrollment token for adding a new Kibana instance:

```
elasticsearch-create-enrollment-token {{[-s|--scope]}} kibana
```

- Create an enrollment token and display verbose output:

```
elasticsearch-create-enrollment-token {{[-s|--scope]}} node
--verbose
```

- Create an enrollment token for a Kibana instance with a custom Elasticsearch URL:

```
elasticsearch-create-enrollment-token {{[-s|--scope]}} kibana
--url "{{IP}}"
```

- Display help:

```
elasticsearch-create-enrollment-token {{[-h|--help]}}
```

elasticsearch-croneval

Validates and evaluates a **`cron`** expression. This command helps verify that **`cron`** expressions are valid for use with Elasticsearch and produce the expected results.

More information: https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-croneval.html.

- Evaluate a `cron` expression and display the next 10 trigger times (default behavior):

  ```
  elasticsearch-croneval "{{cron_expression}}"
  ```

- Evaluate a `cron` expression and display a specific number of future trigger times:

  ```
  elasticsearch-croneval "{{cron_expression}}" {{[-c|--count]}} {{integer}}
  ```

- Display detailed information (including stack trace) for an invalid `cron` expression:

  ```
  elasticsearch-croneval "{{invalid_cron_expression}}" {{[-d|--detail]}}
  ```

- Display minimal output (silent mode):

  ```
  elasticsearch-croneval "{{cron_expression}}" {{[-s|--silent]}}
  ```

- Display verbose output:

  ```
  elasticsearch-croneval "{{cron_expression}}" {{[-v|--verbose]}}
  ```

elasticsearch-keystore

Manage secure settings (e.g., passwords, tokens, and credentials) used by Elasticsearch.

More information: https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-keystore.html.

- Create a new keystore (not password-protected):

```
elasticsearch-keystore create
```

- Create a new password-protected keystore:

```
elasticsearch-keystore create -p
```

- Add a setting interactively:

```
elasticsearch-keystore add {{setting_name}}
```

- Add a setting from `stdin`:

```
echo "{{setting_value}}" | elasticsearch-keystore add --stdin
{{setting_name}}
```

- Remove a setting from the keystore:

```
elasticsearch-keystore remove {{setting_name}}
```

- Change the keystore password:

```
elasticsearch-keystore passwd
```

- List all settings stored in the keystore:

```
elasticsearch-keystore list
```

- Upgrade the keystore format (after an Elasticsearch upgrade):

```
elasticsearch-keystore upgrade
```

elasticsearch-node

Manage low-level Elasticsearch node operations such as shutdown, repurpose, or viewing info.

More information: https://www.elastic.co/docs/reference/elasticsearch/command-line-tools/node-tool.

- Display information about the current node:

  ```
  elasticsearch-node info
  ```

- Prepare the node for a full cluster restart (e.g., after upgrading):

  ```
  elasticsearch-node unsafe-bootstrap
  ```

- Repurpose a node for a different role (e.g., from master to data node):

  ```
  elasticsearch-node repurpose
  ```

- List the roles assigned to the node:

  ```
  elasticsearch-node roles
  ```

- Show the installed JVM version, Elasticsearch home path, and other diagnostic information:

  ```
  elasticsearch-node diagnostics
  ```

- Display help:

  ```
  elasticsearch-node {{[-h|--help]}}
  ```

elasticsearch-reset-password

Reset the passwords of users in the native realm and built-in users.

More information: https://www.elastic.co/docs/reference/elasticsearch/command-line-tools/reset-password.

- Reset the password of the user to an auto-generated value and print it in the console:

```
elasticsearch-reset-password {{[-u|--username]}} {{user}}
```

- Prompt interactively to reset the password for a native user:

```
elasticsearch-reset-password {{[-u|--username]}} {{user}} {{[-i|--interactive]}}
```

- Interactively reset the password for a user at a specified Elasticsearch node URL:

```
elasticsearch-reset-password --url {{host}}:{{port}} {{[-u|--username]}} {{user}} {{[-i|--interactive]}}
```

elasticsearch-saml-metadata

Generate SAML Service Provider metadata for configuring a SAML Identity Provider.

More information: https://www.elastic.co/guide/en/elasticsearch/reference/current/saml-metadata.html.

- Generate SAML metadata for a specific realm and print it to `stdout`:

  ```
  elasticsearch-saml-metadata --realm {{realm_name}}
  ```

- Generate SAML metadata and write it to a specific file:

  ```
  elasticsearch-saml-metadata --realm {{realm_name}} --out {{path/to/file.xml}}
  ```

- Display help:

  ```
  elasticsearch-saml-metadata {{[-h|--help]}}
  ```

elasticsearch-syskeygen

Create a system key file in the Elasticsearch configuration directory.

More information: https://www.elastic.co/docs/reference/elasticsearch/command-line-tools/syskeygen.

- Generate the `system_key` file in the default `$ES_HOME/config` directory:

  ```
  elasticsearch-syskeygen
  ```

elasticsearch-users

Manage native realm users in Elasticsearch, including creating, updating, and deleting users.

More information: https://www.elastic.co/guide/en/elasticsearch/reference/current/users-command.html.

- Add a new user interactively (prompts for password):

```
elasticsearch-users useradd {{username}}
```

- Add a new user and specify roles:

```
elasticsearch-users useradd {{username}} -r {{role1,role2}}
```

- Change the password for an existing user:

```
elasticsearch-users passwd {{username}}
```

- Delete a user:

```
elasticsearch-users userdel {{username}}
```

- List all users in the native realm:

```
elasticsearch-users list
```

electron-packager

Build Electron app executables for Windows, Linux and macOS.

Requires a valid package.json in the application directory.

More information: https://github.com/electron/electron-packager.

- Package an application for the current architecture and platform:

```
electron-packager "{{path/to/app}}" "{{app_name}}"
```

- Package an application for all architectures and platforms:

```
electron-packager "{{path/to/app}}" "{{app_name}}" --all
```

- Package an application for 64-bit Linux:

```
electron-packager "{{path/to/app}}" "{{app_name}}" --
platform="{{linux}}" --arch="{{x64}}"
```

- Package an application for ARM macOS:

```
electron-packager "{{path/to/app}}" "{{app_name}}" --
platform="{{darwin}}" --arch="{{arm64}}"
```

electrum

Ergonomic Bitcoin wallet and private key management.

More information: https://electrum.org.

- Create a new wallet:

```
electrum {{[-w|--wallet]}} {{path/to/new_wallet.dat}} create
```

- Restore an existing wallet from seed offline:

```
electrum {{[-w|--wallet]}} {{path/to/recovery_wallet.dat}}
restore {{[-o|--offline]}}
```

- Create a signed transaction offline:

```
electrum mktx {{recipient}} {{amount}} {{[-f|--fee]}}
0.0000001 {{[-F|--from-addr]}} {{from}} {{[-o|--offline]}}
```

- Display all wallet receiving addresses:

```
electrum listaddresses -a
```

- Sign a message:

```
electrum signmessage {{address}} {{message}}
```

- Verify a message:

```
electrum verifymessage {{address}} {{signature}} {{message}}
```

- Connect only to a specific electrum-server instance:

```
electrum {{[-p|--proxy]}} socks5:{{127.0.0.1}}:9050 {{[-s|--
server]}} {{56ckl5obj37gypcu.onion}}:50001:t {{[-1|--
oneserver]}}
```

elif

This shell keyword is used in combination with **`if`** and **`else`** to define a code branch.

- View documentation for `if` command:

  ```
  tldr if
  ```

elinks

A text based browser similar to **`lynx`**.

More information: https://github.com/rkd77/elinks.

- Start ELinks:

```
elinks
```

- Quit elinks:

```
<Ctrl c>
```

- Dump output of webpage to console, colorizing the text with ANSI control codes:

```
elinks -dump -dump-color-mode {{1}} {{url}}
```

elixir

Elixir programming language interpreter.

More information: https://elixir-lang.org.

- Run an Elixir file:

```
elixir {{path/to/file}}
```

- Evaluate Elixir code by passing it as an argument:

```
elixir {{[-e|--eval]}} "{{code}}"
```

elm

Compile and run Elm source files.

More information: https://elm-lang.org.

- Initialize an Elm project, generates an elm.json file:

```
elm init
```

- Start interactive Elm shell:

```
elm repl
```

- Compile an Elm file, output the result to an `index.html` file:

```
elm make {{source}}
```

- Compile an Elm file, output the result to a JavaScript file:

```
elm make {{source}} --output={{destination}}.js
```

- Start local web server that compiles Elm files on page load:

```
elm reactor
```

- Install Elm package from https://package.elm-lang.org:

```
elm install {{author}}/{{package}}
```

else

This shell keyword is used in combination with **`if`** and **`elif`** to define a code branch if no other branch was valid.

- View documentation for the `if` keyword:

  ```
  tldr if
  ```

elvish

Expressive programming language and a versatile interactive shell.

See also: **nu**.

More information: https://elv.sh/ref/command.html.

- Start an interactive shell session:

```
elvish
```

- Execute specific [c]ommands:

```
elvish -c "{{echo 'elvish is executed'}}"
```

- Execute a specific script:

```
elvish {{path/to/script.elv}}
```

emacs

The extensible, customizable, self-documenting, real-time display editor.

See also: **`emacsclient`**.

More information: https://www.gnu.org/software/emacs.

- Start Emacs and open a file:

```
emacs {{path/to/file}}
```

- Open a file at a specified line number:

```
emacs +{{line_number}} {{path/to/file}}
```

- Run an Emacs Lisp file as a script:

```
emacs --script {{path/to/file.el}}
```

- Start Emacs in console mode (without an X window):

```
emacs {{[-nw|--no-window-system]}}
```

- Start an Emacs server in the background (accessible via `emacsclient`):

```
emacs --daemon
```

- Stop a running Emacs server and all its instances, asking for confirmation on unsaved files:

```
emacsclient --eval '(save-buffers-kill-emacs)'
```

- Save a file in Emacs:

```
<Ctrl x><Ctrl s>
```

- Quit Emacs:

```
<Ctrl x><Ctrl c>
```

emacsclient

Open files in an existing Emacs server.

See also: **`emacs`**.

More information: https://www.gnu.org/software/emacs/manual/html_node/emacs/emacsclient-Options.html.

- Open a file in an existing Emacs server (using GUI if available):

```
emacsclient {{path/to/file}}
```

- Open a file in console mode (without an X window):

```
emacsclient {{[-nw|--no-window-system]}} {{path/to/file}}
```

- Open a file in a new Emacs window:

```
emacsclient {{[-c|--create-frame]}} {{path/to/file}}
```

- Evaluate a command, printing the output to `stdout`, and then quit:

```
emacsclient {{[-e|--eval]}} '({{command}})'
```

- Specify an alternative editor in case no Emacs server is running:

```
emacsclient {{[-a|--alternate-editor]}} {{editor}} {{path/to/file}}
```

- Stop a running Emacs server and all its instances, asking for confirmation on unsaved files:

```
emacsclient {{[-e|--eval]}} '(save-buffers-kill-emacs)'
```

ember

Ember CLI: create and manage Ember.js applications.

More information: https://cli.emberjs.com/release/advanced-use/cli-commands-reference/.

- Create a new Ember application:

```
ember new {{my_new_app}}
```

- Create a new Ember addon:

```
ember addon {{my_new_addon}}
```

- Build the project:

```
ember build
```

- Build the project in production mode:

```
ember build -prod
```

- Run the development server:

```
ember serve
```

- Run the test suite:

```
ember test
```

- Run a blueprint to generate something like a route or component:

```
ember generate {{type}} {{name}}
```

- Install an ember-cli addon:

```
ember install {{name_of_addon}}
```

emulator

Manage Android emulators.

More information: https://developer.android.com/studio/run/emulator-commandline.

- Start an Android emulator device:

```
emulator -avd {{name}}
```

- Display the webcams on your development computer that are available for emulation:

```
emulator -avd {{name}} -webcam-list
```

- Start an emulator overriding the facing back camera setting (use `-camera-front` for front camera):

```
emulator -avd {{name}} -camera-back {{none|emulated|webcamN}}
```

- Start an emulator, with a maximum network speed:

```
emulator -avd {{name}} -netspeed {{gsm|hscsd|gprs|edge|hsdpa|lte|evdo|full}}
```

- Start an emulator with network latency:

```
emulator -avd {{name}} -netdelay {{gsm|hscsd|gprs|edge|hsdpa|lte|evdo|none}}
```

- Start an emulator, making all TCP connections through a specified HTTP/HTTPS proxy (port number is required):

```
emulator -avd {{name}} -http-proxy {{http://example.com:80}}
```

- Start an emulator with a given SD card partition image file:

```
emulator -avd {{name}} -sdcard {{path/to/sdcard.img}}
```

- Display help:

```
emulator -help
```

enable

Enable and disable shell builtins.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-enable.

- Print the list of builtins:

```
enable
```

- Disable a builtin (works in Bash only):

```
enable -n {{command}}
```

enca

Detect and convert the encoding of text files.

More information: https://github.com/nijel/enca.

- Detect file(s) encoding according to the system's locale:

```
enca {{path/to/file1 path/to/file2 ...}}
```

- Detect file(s) encoding specifying a language in the POSIX/C locale format (e.g. zh_CN, en_US):

```
enca {{[-L|--language]}} {{language}} {{path/to/file1 path/to/file2 ...}}
```

- Convert file(s) to a specific encoding:

```
enca {{[-L|--language]}} {{language}} {{[-x|--convert-to]}} {{to_encoding}} {{path/to/file1 path/to/file2 ...}}
```

- Create a copy of an existing file using a different encoding:

```
enca < {{original_file}} {{[-L|--language]}} {{language}} {{[-x|--convert-to]}} {{to_encoding}} > {{new_file}}
```

encfs

Mount or create encrypted virtual filesystems.

See also: **`fusermount`** which can unmount filesystems mounted by this command.

More information: https://github.com/vgough/encfs.

- Initialize or mount an encrypted filesystem:

```
encfs /{{path/to/cipher_directory}} /{{path/to/mount_point}}
```

- Initialize an encrypted filesystem with standard settings:

```
encfs --standard /{{path/to/cipher_directory}} /{{path/to/
mount_point}}
```

- Run encfs in the foreground instead of spawning a daemon:

```
encfs -f /{{path/to/cipher_directory}} /{{path/to/
mount_point}}
```

- Mount an encrypted snapshot of a plain directory:

```
encfs --reverse {{path/to/plain_directory}} {{path/to/
cipher_directory}}
```

enscript

Convert text files to PostScript, HTML, RTF, ANSI, and overstrikes.

More information: https://manned.org/enscript.

- Generate a PostScript file from a text file:

```
enscript {{path/to/input_file}} {{[-o|--output]}} {{path/to/output_file}}
```

- Generate a file in a different language than PostScript:

```
enscript {{path/to/input_file}} {{[-w|--language]}} {{html|rtf|...}} {{[-o|--output]}} {{path/to/output_file}}
```

- Generate a PostScript file with a landscape layout, splitting the page into columns (maximum 9):

```
enscript {{path/to/input_file}} --columns {{num}} {{[-r|--landscape]}} {{[-o|--output]}} {{path/to/output_file}}
```

- Display available syntax highlighting languages and file formats:

```
enscript --help-highlight
```

- Generate a PostScript file with syntax highlighting and color for a specified language:

```
enscript {{path/to/input_file}} --color 1 {{[-E|--highlight]}} {{language}} {{[-o|--output]}} {{path/to/output_file}}
```

entr

Run arbitrary commands when files change.

More information: https://eradman.com/entrproject/.

- Rebuild with `make` if any file in any subdirectory changes:

```
{{ag --files-with-matches}} | entr {{make}}
```

- Rebuild and test with `make` if any `.c` source files in the current directory change:

```
{{ls *.c}} | entr '{{make && make test}}'
```

- Send a `SIGTERM` to any previously spawned ruby subprocesses before executing `ruby main.rb`:

```
{{ls *.rb}} | entr -r {{ruby main.rb}}
```

- Run a command with the changed file (`/_`) as an argument:

```
{{ls *.sql}} | entr {{psql -f}} /_
```

- [c]lear the screen and run a query after the SQL script is updated:

```
{{echo my.sql}} | entr -cp {{psql -f}} /_
```

- Rebuild the project if source files change, limiting output to the first few lines:

```
{{find src/}} | entr -s '{{make | sed 10q}}'
```

- Launch and auto-[r]eload a Node.js server:

```
{{ls *.js}} | entr -r {{node app.js}}
```

env

Show the environment or run a program in a modified environment.

More information: https://www.gnu.org/software/coreutils/manual/html_node/env-invocation.html.

- Show the environment:

```
env
```

- Run a program. Often used in scripts after the shebang (#!) for looking up the path to the program:

```
env {{program}}
```

- Clear the environment and run a program:

```
env {{[-i|--ignore-environment]}} {{program}}
```

- Remove variable from the environment and run a program:

```
env {{[-u|--unset]}} {{variable}} {{program}}
```

- Set a variable and run a program:

```
env {{variable}}={{value}} {{program}}
```

- Set one or more variables and run a program:

```
env {{variable1=value variable2=value variable3=value ...}} {{program}}
```

- Run a program under a different name:

```
env {{[-a|--argv0]}} {{custom_name}} {{program}}
```

envoy

A PHP-based task manager for Laravel remote servers.

More information: https://laravel.com/docs/envoy.

- Initialize a configuration file:

```
envoy init {{host_name}}
```

- Run a task:

```
envoy run {{task_name}}
```

- Run a task from a specific project:

```
envoy run --path {{path/to/directory}} {{task_name}}
```

- Run a task and continue on failure:

```
envoy run --continue {{task_name}}
```

- Dump a task as a Bash script for inspection:

```
envoy run --pretend {{task_name}}
```

- Connect to the specified server via SSH:

```
envoy ssh {{server_name}}
```

envsubst

Substitutes environment variables with their value in shell format strings.

Variables to be replaced should be in either **`${var}`** or **`$var`** format.

More information: https://www.gnu.org/software/gettext/manual/gettext.html#envsubst-Invocation.

- Replace environment variables in `stdin` and output to `stdout`:

```
echo '{{$HOME}}' | envsubst
```

- Replace environment variables in an input file and output to `stdout`:

```
envsubst < {{path/to/input_file}}
```

- Replace environment variables in an input file and output to a file:

```
envsubst < {{path/to/input_file}} > {{path/to/output_file}}
```

- Replace environment variables in an input file from a space-separated list:

```
envsubst < {{path/to/input_file}} '{{$USER $SHELL $HOME}}'
```

eol

Show end-of-life dates (EoLs) for a number of products.

More information: https://github.com/hugovk/norwegianblue.

- List all available products:

```
eol
```

- Get EoLs of one or more products:

```
eol {{product1 product2 ...}}
```

- Open the product webpage:

```
eol {{product}} --web
```

- Get EoLs of a one or more products in a specific format:

```
eol {{product1 product2 ...}} --format {{html|json|md|
markdown|pretty|rst|csv|tsv|yaml}}
```

- Get EoLs of one or more products as a single markdown file:

```
eol {{product1 product2 ...}} --format {{markdown}} >
{{eol_report.md}}
```

- Display help:

```
eol --help
```

eqn

Equation preprocessor for the groff (GNU Troff) document formatting system.

See also: **`troff`**, **`groff`**.

More information: https://manned.org/eqn.

- Process input with equations, saving the output for future typesetting with groff to PostScript:

```
eqn {{path/to/input.eqn}} > {{path/to/output.roff}}
```

- Typeset an input file with equations to PDF using the [me] macro package:

```
eqn -T {{pdf}} {{path/to/input.eqn}} | groff -{{me}} -T
{{pdf}} > {{path/to/output.pdf}}
```

erl

Run and manage programs in the Erlang programming language.

More information: https://www.erlang.org.

- Compile and run sequential Erlang program as a common script and then exit:

```
erlc {{path/to/file1 path/to/file2 ...}} && erl -noshell
'{{mymodule:myfunction(arguments)}}, init:stop().'
```

- Connect to a running Erlang node:

```
erl -remsh {{nodename}}@{{hostname}} -sname
{{custom_shortname}} -hidden -setcookie
{{cookie_of_remote_node}}
```

- Tell the Erlang shell to load modules from a directory:

```
erl -pa {{path/to/directory_with_beam_files}}
```

ern

Electrode Native platform client.

More information: https://native.electrode.io/reference/index-6.

- Create a new `ern` application (`MiniApp`):

```
ern create-miniapp {{application_name}}
```

- Run one or more `MiniApps` in the iOS/Android Runner application:

```
ern run-{{ios|android}}
```

- Create an Electrode Native container:

```
ern create-container --miniapps /{{path/to/
miniapp_directory}} --platform {{ios|android}}
```

- Publish an Electrode Native container to a local Maven repository:

```
ern publish-container --publisher {{maven}} --platform
{{android}} --extra
'{{{"groupId":"com.walmart.ern","artifactId":"quickstart"}}}'
```

- Transform an iOS container into a pre-compiled binary framework:

```
ern transform-container --platform {{ios}} --transformer
{{xcframework}}
```

- List all installed versions of Electrode Native:

```
ern platform versions
```

- Set a logging level:

```
ern platform config set logLevel {{trace|debug}}
```

errno

Look up errno names and descriptions.

More information: https://joeyh.name/code/moreutils/.

- Lookup errno description by name or code:

```
errno {{name|code}}
```

- List all errno names, codes, and descriptions:

```
errno {{[-l|--list]}}
```

- Search for code whose description contains all of the given text:

```
errno {{[-s|--search]}} {{text}}
```

- Search for code whose description contains all of the given text (all locales):

```
errno {{[-S|--search-all-locales]}} {{text}}
```

esac

This shell keyword is used to end a **`case`** statement.

- View documentation for the `case` keyword:

  ```
  tldr case
  ```

esbuild

JavaScript bundler and minifier built for speed.

More information: https://esbuild.github.io/api/#general-options.

- Bundle a JavaScript application and print to `stdout`:

```
esbuild --bundle {{path/to/file.js}}
```

- Bundle a JSX application from `stdin`:

```
esbuild < {{path/to/file.jsx}} --bundle --outfile={{path/to/out.js}}
```

- Bundle and minify a JSX application with source maps in `production` mode:

```
esbuild --bundle --define:{{process.env.NODE_ENV=\"production\"}} --minify --sourcemap {{path/to/file.js}}
```

- Bundle a JSX application for a comma-separated list of browsers:

```
esbuild --bundle --minify --sourcemap --target={{chrome58,firefox57,safari11,edge16}} {{path/to/file.jsx}}
```

- Bundle a JavaScript application for a specific node version:

```
esbuild --bundle --platform={{node}} --target={{node12}} {{path/to/file.js}}
```

- Bundle a JavaScript application enabling JSX syntax in `.js` files:

```
esbuild --bundle app.js --loader:{{.js=jsx}} {{path/to/file.js}}
```

- Bundle and serve a JavaScript application on an HTTP server:

```
esbuild --bundle --serve={{port}} --outfile={{index.js}} {{path/to/file.js}}
```

- Bundle a list of files to an output directory:

```
esbuild --bundle --outdir={{path/to/output_directory}} {{path/to/file1 path/to/file2 ...}}
```

escp2topbm

Convert a PBM image to a ESC/P2 printer file.

See also: **pbmtoescp2**.

More information: https://netpbm.sourceforge.net/doc/escp2topbm.html.

- Convert a ESC/P2 printer file to a PBM image:

```
escp2topbm {{path/to/image.escp2}} > {{path/to/output.pbm}}
```

eslint

A pluggable linting utility for JavaScript and JSX.

More information: https://eslint.org.

- Create the ESLint configuration file:

```
eslint --init
```

- Lint one or more files:

```
eslint {{path/to/file1.js path/to/file2.js ...}}
```

- Fix lint issues:

```
eslint --fix
```

- Lint using the specified configuration file:

```
eslint {{[-c|--config]}} {{path/to/config_file}} {{path/to/
file1.js path/to/file2.js ...}}
```

espanso

Cross-platform Text Expander written in Rust.

More information: https://espanso.org/docs/command%20lIne/cli_list/.

- Check status:

```
espanso status
```

- Edit the configuration:

```
espanso edit config
```

- Install a package from the hub store (https://hub.espanso.org/):

```
espanso install {{package}}
```

- Restart (required after installing a package, useful in case of failure):

```
espanso restart
```

espeak

Use text-to-speech to speak through the default sound device.

More information: https://espeak.sourceforge.net.

- Speak a phrase aloud:

```
espeak "I like to ride my bike."
```

- Speak a file aloud:

```
espeak -f {{path/to/file}}
```

- Save output to a WAV audio file, rather than speaking it directly:

```
espeak -w {{filename.wav}} "It's GNU plus Linux"
```

- Use a different voice:

```
espeak -v {{voice}}
```

esptool.py

Bootloader utility for Espressif chips (e.g. ESP8266).

More information: https://docs.espressif.com/projects/esptool/en/latest/esp32/.

- Flash a firmware file to an ESP chip with a given port and baud rate:

```
sudo esptool.py --port {{port}} --baud {{baud_rate}}
write_flash 0x0 {{path/to/firmware.bin}}
```

- Clear the flash of an ESP chip:

```
sudo esptool.py --port {{port}} --baud {{baud_rate}}
erase_flash
```

etcd

A distributed, reliable key-value store for the most critical data of a distributed system.

More information: https://etcd.io.

- Start a single-node etcd cluster:

```
etcd
```

- Start a single-node etcd cluster, listening for client requests on a custom URL:

```
etcd --advertise-client-urls {{http://127.0.0.1:1234}} --
listen-client-urls {{http://127.0.0.1:1234}}
```

- Start a single-node etcd cluster with a custom name:

```
etcd --name {{my_etcd_cluster}}
```

- Start a single-node etcd cluster with extensive metrics available at http://localhost:2379/debug/pprof/:

```
etcd --enable-pprof --metrics extensive
```

etcdctl

Interact with **etcd**, a highly-available key-value pair store.

More information: https://etcd.io/docs/latest/dev-guide/interacting_v3/.

- Display the value associated with a specified key:

```
etcdctl get {{my/key}}
```

- Store a key-value pair:

```
etcdctl put {{my/key}} {{my_value}}
```

- Delete a key-value pair:

```
etcdctl del {{my/key}}
```

- Store a key-value pair, reading the value from a file:

```
etcdctl < {{path/to/file.txt}} put {{my/file}}
```

- Save a snapshot of the etcd keystore:

```
etcdctl snapshot save {{path/to/snapshot.db}}
```

- Restore a snapshot of an etcd keystore (restart the etcd server afterwards):

```
etcdctl snapshot restore {{path/to/snapshot.db}}
```

- Add a user:

```
etcdctl user add {{my_user}}
```

- Watch a key for changes:

```
etcdctl watch {{my/key}}
```

eva

Simple calculator REPL, similar to **bc**, with syntax highlighting and persistent history.

More information: https://github.com/NerdyPepper/eva.

- Run the calculator in interactive mode:

```
eva
```

- Calculate the result of an expression:

```
eva "{{(1 + 2) * 2 ^ 2}}"
```

- Calculate an expression forcing the number of decimal places to 5:

```
eva --fix {{5}} "{{5 / 3}}"
```

- Calculate an expression with sine and cosine:

```
eva "{{sin(1) + cos(1)}}"
```

eval

Execute arguments as a single command in the current shell and return its result.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-eval.

- Call `echo` with the "foo" argument:

```
eval "{{echo foo}}"
```

- Set a variable in the current shell:

```
eval "{{foo=bar}}"
```

evil-winrm

Windows Remote Management (WinRM) shell for pentesting.

Once connected, we get a PowerShell prompt on the target host.

More information: https://github.com/Hackplayers/evil-winrm.

- Connect to a host:

```
evil-winrm {{[-i|--ip]}} {{ip}} {{[-u|--user]}} {{user}} {{[-p|--password]}} {{password}}
```

- Connect to a host using pass-the-hash authentication instead of a password:

```
evil-winrm {{[-i|--ip]}} {{ip}} {{[-u|--user]}} {{user}} {{[-H|--hash]}} {{nt_hash}}
```

- Connect to a host, specifying directories for PowerShell scripts and executables:

```
evil-winrm {{[-i|--ip]}} {{ip}} {{[-u|--user]}} {{user}} {{[-p|--password]}} {{password}} {{[-s|--scripts]}} {{path/to/scripts}} {{[-e|--executables]}} {{path/to/executables}}
```

- Connect to a host, using SSL:

```
evil-winrm {{[-i|--ip]}} {{ip}} {{[-u|--user]}} {{user}} {{[-p|--password]}} {{password}} {{[-S|--ssl]}} {{[-c|--pub-key]}} {{path/to/pubkey}} {{[-k|--priv-key]}} {{path/to/privkey}}
```

- Upload a file to the host:

```
PS > upload {{path/to/local_file}} {{path/to/remote_file}}
```

- List all loaded PowerShell functions:

```
PS > menu
```

- Load a PowerShell script from the `--scripts` directory:

```
PS > {{script.ps1}}
```

- Invoke a binary on the host from the `--executables` directory:

```
PS > Invoke-Binary {{binary.exe}}
```

ex

Text editor.

See also: **`vim`**.

More information: https://www.vim.org.

- Open a file:

  `ex {{path/to/file}}`

- Save and Quit:

  `wq<Enter>`

- Undo the last operation:

  `undo<Enter>`

- Search for a pattern in the file:

  `/{{search_pattern}}<Enter>`

- Perform a `regex` substitution in the whole file:

  `%s/{{regex}}/{{replacement}}/g<Enter>`

- Insert text:

  `i<Enter>{{text}}<Ctrl c>`

- Switch to Vim:

  `visual<Enter>`

exa

A modern replacement for **`ls`** (List directory contents).

More information: https://github.com/ogham/exa#command-line-options.

- List files one per line:

```
exa {{[-1|--oneline]}}
```

- List all files, including hidden files:

```
exa {{[-a|--all]}}
```

- Long format list (permissions, ownership, size and modification date) of all files:

```
exa {{[-l|--long]}} {{[-a|--all]}}
```

- List files with the largest at the top:

```
exa {{[-r|--reverse]}} {{[-s|--sort]}} {{size}}
```

- Display a tree of files, three levels deep:

```
exa {{[-l|--long]}} {{[-T|--tree]}} {{[-L|--level]}} {{3}}
```

- List files sorted by modification date (oldest first):

```
exa {{[-l|--long]}} {{[-s|--sort]}} {{modified}}
```

- List files with their headers, icons, and Git statuses:

```
exa {{[-l|--long]}} {{[-h|--header]}} --icons --git
```

- Don't list files mentioned in `.gitignore`:

```
exa --git-ignore
```

exec

Execute a command without creating a child process.

More information: https://manned.org/exec.1posix.

- Execute a specific command using the current environment variables:

```
exec {{command -with -flags}}
```

exenv

Easily install Elixir versions and manage application environments.

More information: https://github.com/mururu/exenv.

- Display a list of installed versions:

  ```
  exenv versions
  ```

- Use a specific version of Elixir across the whole system:

  ```
  exenv global {{version}}
  ```

- Use a specific version of Elixir for the current application/project directory:

  ```
  exenv local {{version}}
  ```

- Show the currently selected Elixir version:

  ```
  exenv {{version}}
  ```

- Install a version of Elixir (requires `elixir-build` plugin https://github.com/mururu/elixir-build):

  ```
  exenv install {{version}}
  ```

exercism

Download coding exercises in any language and submit your solutions.

More information: https://exercism.org/docs/using/solving-exercises/working-locally.

- Configure exercism following instructions for the API token:

  ```
  exercism configure
  ```

- Download `hello-world` for any programming language into exercism workspace:

  ```
  exercism download --track {{programming_language}} --exercise hello-world
  ```

- Open an exercise directory on the website:

  ```
  exercism open {{path/to/directory}}
  ```

- Test your solution. It will also tell you if you are missing tools for the programming language:

  ```
  exercism test {{path/to/directory}}
  ```

- Submit your solution. You can share from the website or ask a community mentor for help:

  ```
  exercism submit {{path/to/directory}}
  ```

exfatlabel

Get or set an exFAT filesystem label.

More information: https://manned.org/exfatlabel.

- Display the current filesystem label:

```
exfatlabel {{/dev/sda}}
```

- Set the filesystem label:

```
exfatlabel {{/dev/sda}} {{new_label}}
```

exiftool

Read and write meta information in files.

More information: https://exiftool.org.

- Print the EXIF metadata for a given file:

```
exiftool {{path/to/file}}
```

- Remove all EXIF metadata from the given files:

```
exiftool -All= {{path/to/file1 path/to/file2 ...}}
```

- Remove GPS EXIF metadata from given image files:

```
exiftool "-gps*=" {{path/to/image1 path/to/image2 ...}}
```

- Remove all EXIF metadata from the given image files, then re-add metadata for color and orientation:

```
exiftool -All= -tagsfromfile @ -colorspacetags -orientation
{{path/to/image1 path/to/image2 ...}}
```

- Move the date at which all photos in a directory were taken 1 hour forward:

```
exiftool "-AllDates+=0:0:0 1:0:0" {{path/to/directory}}
```

- Move the date at which all JPEG photos in the current directory were taken 1 day and 2 hours backward:

```
exiftool "-AllDates-=0:0:1 2:0:0" {{[-ext|-extension]}} jpg
```

- Only change the `DateTimeOriginal` field subtracting 1.5 hours, without keeping backups:

```
exiftool -DateTimeOriginal-=1.5 -overwrite_original
```

- Recursively rename all JPEG photos in a directory based on the `DateTimeOriginal` field:

```
exiftool '-filename<DateTimeOriginal' {{[-d|-dateFormat]}}
%Y-%m-%d_%H-%M-%S%%lc.%%e {{path/to/directory}} {{[-r|-
recurse]}} {{[-ext|-extension]}} jpg
```

exit

Exit the shell.

More information: https://manned.org/exit.1posix.

- Exit with the exit status of the most recently executed command:

```
exit
```

- Exit with a specific exit status:

```
exit {{exit_code}}
```

exiv2

Image metadata manipulation tool.

More information: https://www.exiv2.org/manpage.html.

- Print a summary of the image Exif metadata:

```
exiv2 {{path/to/file}}
```

- Print all metadata (Exif, IPTC, XMP) with interpreted values:

```
exiv2 {{[-P|-Print]}} kt {{path/to/file}}
```

- Print all metadata with raw values:

```
exiv2 {{[-P|-Print]}} kv {{path/to/file}}
```

- Delete all metadata from an image:

```
exiv2 {{[-d|--delete]}} a {{path/to/file}}
```

- Delete all metadata, preserving the file timestamp:

```
exiv2 {{[-d|--delete]}} a {{[-k|--keep]}} {{path/to/file}}
```

- Rename the file, prepending the date and time from metadata (not from the file timestamp):

```
exiv2 {{[-r|--rename]}}
'{{%Y%m%d_%H%M%S_:basename:}}' {{path/to/file}}
```

exo compute block-storage

Manage the Exoscale Block Storage service.

More information: https://community.exoscale.com/product/storage/block-storage/.

- Create a 20GB Block Storage Volume:

```
exo compute block-storage create {{volume_name}} --size 20
{{[-z|--zone]}} {{zone}}
```

- List Block Storage Volumes:

```
exo compute block-storage list
```

- Attach a Block Storage Volume to a Compute instance:

```
exo compute block-storage attach {{volume_name|id}}
{{instance_name|id}} {{[-z|--zone]}} {{zone}}
```

- Forcefully detach a Block Storage Volume (does not require confirmation):

```
exo compute block-storage detach {{volume_name|id}} {{[-z|--
zone]}} {{zone}} {{[-f|--force]}}
```

- Create a snapshot of a Block Storage Volume:

```
exo compute block-storage snapshot create {{volume_name|id}}
--name {{snapshot_name}} {{[-z|--zone]}} {{zone}}
```

- Create a Block Storage Volume from a snapshot:

```
exo compute block-storage create {{volume_name}} --snapshot
{{snapshot_name|id}} {{[-z|--zone]}} {{zone}}
```

- Update an existing Block Storage Volume with a new name and a new volume size of 30GB:

```
exo compute block-storage update {{volume_name|id}} --size 30
--name {{new_name}}
```

exo compute instance

Manage Exoscale Compute instances.

More information: https://community.exoscale.com/product/compute/instances/.

- Create a Debian based Compute instance with a disk size of 10GB:

```
exo compute instance create --disk-size 10 {{instance_name}}
{{[-z|--zone]}} {{zone}} --template '{{Linux Debian 12
(Bookworm) 64-bit}}'
```

- Log into a Compute instance via SSH:

```
exo compute instance ssh {{instance_name|id}}
```

- List all of the Compute instances:

```
exo compute instance list
```

- Add an instance to a Security Group:

```
exo compute instance security-group add {{instance_name|id}}
{{security_group_name|id}}
```

- Scale the size of a Compute instance:

```
exo compute instance scale {{instance_name|id}}
{{instance_type}}
```

- Create a snapshot of a Compute instance:

```
exo compute instance snapshot create {{instance_name|id}}
```

- Revert a Compute instance to a snapshot (the data written after the snapshot has been created will be lost):

```
exo compute instance snapshot revert {{snapshot_id}}
{{instance_name|id}}
```

- Resize the disk size of a Compute instance to 20GB:

```
exo compute instance resize-disk {{instance_name|id}} 20
```

exo compute sks

Manage the Exoscale Scalable Kubernetes Service (SKS).

More information: https://community.exoscale.com/product/compute/containers/.

- List the supported SKS cluster versions:

```
exo compute sks versions
```

- Create a new SKS cluster:

```
exo compute sks create {{cluster_name}} {{[-z|--zone]}}
{{zone}}
```

- List all SKS clusters:

```
exo compute sks list
```

- Generate a Kubernetes kubeconfig file for an SKS cluster that expires in 1800 seconds:

```
exo compute sks kubeconfig {{cluster_name|id}} {{user}} --ttl
1800 {{[-z|--zone]}} {{zone}}
```

- Create and add a Nodepool containing 3 nodes to an SKS cluster:

```
exo compute sks nodepool add {{cluster_name|id}}
{{nodepool_name}} --size 3 {{[-z|--zone]}} {{zone}}
```

- Remove a Nodepool from an SKS cluster:

```
exo compute sks nodepool delete {{cluster_name|id}}
{{nodepool_name|id}}
```

- Evict a Node from a Nodepool in an SKS cluster:

```
exo compute sks nodepool evict {{cluster_name|id}}
{{nodepool_name|id}} {{node_name|id}}
```

- Enable the Exoscale CSI driver for an existing SKS cluster:

```
exo compute sks update {{cluster_name|id}} --enable-csi-addon
{{[-z|--zone]}} {{zone}}
```

exo compute

Manage Exoscale Compute resources.

Some subcommands such as **`instance`** have their own documentation.

More information: https://community.exoscale.com/product/.

- Quickly create an Exoscale Compute resource (e.g., instance, Security Group, SKS cluster,...):

```
exo compute {{resource_type}} create {{resource_name}}
```

- List Exoscale Compute instance types:

```
exo compute instance-type list
```

- Register a new SSH key that can be used to access Compute instances:

```
exo compute ssh-key register {{key_name}} {{public_key_file}}
```

- Create a Compute instance with an ssh-key deployed on it:

```
exo compute instance create {{instance_name}}
{{ssh_key_name}}
```

- Register a new Compute instance template based on a Snapshot of a Compute instance (useful when you want to quickly create a replica of a Compute instance):

```
exo compute instance template register {{template_name}} --
from-snapshot {{snapshot_id}}
```

- Add a new rule to an existing Security Group:

```
exo compute security-group rule add {{security_group_name|
id}} --description '{{Allow SSH access}}' --flow {{ingress}}
--port {{22}} --network {{0.0.0.0/0}}
```

- Manage the services of an existing Network Load Balancer:

```
exo compute load-balancer service add {{load_balancer_name|
id}} {{service_name}} --port {{service_port}}
```

exo dbaas

Manage Exoscale DBaaS.

More information: https://community.exoscale.com/product/dbaas/.

- List available Database Service types:

```
exo dbaas type list
```

- List available plans for a Database Service type:

```
exo dbaas type show {{database_service_type}} --plans
```

- Create a new Database Service (an IP filter must be specified in order to access the service):

```
exo dbaas create {{database_service_type}}
{{database_service_type_plan}} {{database_service_name}} --
{{database_service_type}}-ip-filter {{1.2.3.4/32}}
```

- Show the connection URI for a Database Service:

```
exo dbaas show {{database_service_name}} --uri
```

- Set a specified maintenance time and day of the week for a Database Service:

```
exo dbaas update {{database_service_name}} --maintenance-dow
{{day_of_the_week}} --maintenance-time {{HH:MM:SS}}
```

- Get help for a specific Database Service type:

```
exo dbaas {{subcommand}} --help-{{database_service_type}}
```

exo iam

Manage the Exoscale IAM service.

More information: https://community.exoscale.com/product/iam/.

- List all of the IAM roles:

```
exo iam role list
```

- Create a new API key:

```
exo iam api-key create {{api_key_name}} {{iam_role_name}}
```

- Create a new IAM role:

```
cat {{path/to/policy.json}} | exo iam role create
{{iam_role_name}} --editable --policy -
```

- Show the policy of an existing IAM role:

```
exo iam role show {{iam_role_name}} --policy {{[-O|--output-
format]}} {{json}} | jq .
```

- Update the default Organization policy (the default Organization policy will be applied to all of the API keys within the Organization):

```
cat {{path/to/policy.json}} | exo iam org-policy update -
```

exo storage

Manage the Exoscale Simple Object Storage (SOS) service.

More information: https://community.exoscale.com/product/storage/object-storage/.

- Create a new SOS bucket:

```
exo storage mb {{bucket_name}}
```

- Upload a file to a bucket:

```
exo storage put {{path/to/file}} {{bucket_name}}/{{prefix/}}
```

- List the objects within a bucket:

```
exo storage ls {{bucket_name}}
```

- Simulate the download of an object from a bucket:

```
exo storage get {{bucket_name}}/{{object_key}} {{local_path}}
--dry-run
```

- Manage the metadata of an object:

```
exo storage metadata add {{bucket_name}}/{{object_key}}
{{key=value}}
```

exo

Manage the Exoscale services.

Some subcommands such as **`exo compute`** have their own usage documentation.

More information: https://community.exoscale.com/tools/command-line-interface/.

- Configure the exo command-line:

```
exo config
```

- Generate the exo autocompletion script for a specified shell:

```
exo completion {{zsh}}
```

- List all of the available zones and output them as json:

```
exo zone {{[-O|--output-format]}} {{json}}
```

- Quietly create a Compute instance in a specific zone (disables the non-essential command output):

```
exo compute instance create {{instance_name}} --zone {{zone}} {{[-Q|--quiet]}}
```

- List just the name of all of the buckets in the Organization:

```
exo storage list {{[-O|--output-template]}} '\{\{ .Name \}\}
```

- Display help for a specific sub-command:

```
exo {{iam}} {{[-h|--help]}}
```

expand

Convert tabs to spaces.

More information: https://www.gnu.org/software/coreutils/manual/html_node/expand-invocation.html.

- Convert tabs in each file to spaces, writing to `stdout`:

```
expand {{path/to/file}}
```

- Convert tabs to spaces, reading from `stdin`:

```
expand
```

- Do not convert tabs after non blanks:

```
expand {{[-i|--initial]}} {{path/to/file}}
```

- Have tabs a certain number of characters apart, not 8:

```
expand {{[-t|--tabs]}} {{number}} {{path/to/file}}
```

- Use a comma separated list of explicit tab positions:

```
expand {{[-t|--tabs]}} {{1,4,6}}
```

export

Export shell variables to child processes.

More information: https://manned.org/export.1posix.

- Set an environment variable:

```
export {{VARIABLE}}={{value}}
```

- Append a pathname to the environment variable `PATH`:

```
export PATH=$PATH:{{path/to/append}}
```

expose

An open source tunnel application for sharing websites.

More information: https://beyondco.de/docs/expose.

- Register your authentication token:

  ```
  expose token {{token}}
  ```

- Share the current working directory:

  ```
  expose
  ```

- Share the current working directory with a specific subdomain:

  ```
  expose --subdomain={{subdomain}}
  ```

- Share a local URL:

  ```
  expose share {{url}}
  ```

- Run the Expose server:

  ```
  expose serve
  ```

- Run the Expose server with a specific hostname:

  ```
  expose serve {{hostname}}
  ```

expr

Evaluate expressions and manipulate strings.

More information: https://www.gnu.org/software/coreutils/manual/html_node/expr-invocation.html.

- Get the length of a specific string:

```
expr length "{{string}}"
```

- Get the substring of a string with a specific length:

```
expr substr "{{string}}" {{from}} {{length}}
```

- Match a specific substring against an anchored pattern:

```
expr match "{{string}}" '{{pattern}}'
```

- Get the first char position from a specific set in a string:

```
expr index "{{string}}" "{{chars}}"
```

- Calculate a specific mathematic expression:

```
expr {{expression1}} {{+|-|*|/|%}} {{expression2}}
```

- Get the first expression if its value is non-zero and not null otherwise get the second one:

```
expr {{expression1}} \| {{expression2}}
```

- Get the first expression if both expressions are non-zero and not null otherwise get zero:

```
expr {{expression1}} \& {{expression2}}
```

exrex

Generate all/random matching strings for a **regex**.

It can also simplify **regex**es.

More information: https://github.com/asciimoo/exrex.

- Generate all possible strings that match a `regex`:

```
exrex '{{regex}}'
```

- Generate a random string that matches a `regex`:

```
exrex {{[-r|--random]}} '{{regex}}'
```

- Generate at most 100 strings that match a `regex`:

```
exrex {{[-m|--max-number]}} {{100}} '{{regex}}'
```

- Generate all possible strings that match a `regex`, joined by a custom delimiter string:

```
exrex {{[-d|--delimiter]}} "{{, }}" '{{regex}}'
```

- Print count of all possible strings that match a `regex`:

```
exrex {{[-c|--count]}} '{{regex}}'
```

- Simplify a `regex`:

```
exrex {{[-s|--simplify]}} '{{ab|ac}}'
```

- Print eyes:

```
exrex '{{[oO0](_)[oO0]}}'
```

- Print a boat:

```
exrex '{{( {20}(\| *\\|-{22}|\|)|\.={50}| ( ){0,5}\\\.| {12}
~{39})}}'
```

eyuvtoppm

Convert a Berkeley YUV file to PPM.

More information: https://netpbm.sourceforge.net/doc/eyuvtoppm.html.

- Read a Berkeley YUV file from the specified input file, convert it to a PPM image and store it in the specified output file:

```
eyuvtoppm --width {{width}} --height {{height}} {{path/to/input_file.eyuv}} > {{path/to/output_file.ppm}}
```

eza

Modern, maintained replacement for **`ls`**, built on **`exa`**.

More information: https://github.com/eza-community/eza.

- List files one per line:

```
eza {{[-1|--oneline]}}
```

- List all files, including hidden files:

```
eza {{[-a|--all]}}
```

- Long format list (permissions, ownership, size and modification date) of all files:

```
eza {{[-al|--all --long]}}
```

- List files with the largest at the top:

```
eza {{[-r|--reverse]}} {{[-s|--sort]}} {{size}}
```

- Display a tree of files, three levels deep:

```
eza {{[-lT|--long --tree]}} {{[-L|--level]}} {{3}}
```

- List files sorted by modification date (oldest first):

```
eza {{[-l|--long]}} {{[-s|--sort]}} {{modified}}
```

- List files with their headers, icons, and Git statuses:

```
eza {{[-lh|--long --header]}} --icons --git
```

- Don't list files mentioned in `.gitignore`:

```
eza --git-ignore
```

f3fix

Edit the partition table of a fake flash drive.

See also: **`f3probe`**, **`f3write`**, **`f3read`**.

More information: https://oss.digirati.com.br/f3/.

- Fill a fake flash drive with a single partition that matches its real capacity:

```
sudo f3fix {{/dev/device_name}}
```

- Mark the partition as bootable:

```
sudo f3fix --boot {{/dev/device_name}}
```

- Specify the filesystem:

```
sudo f3fix --fs-type={{filesystem_type}} {{/dev/device_name}}
```

f3probe

Probe a block device (e.g. a flash drive or a microSD card) for counterfeit flash memory.

See also: **`f3read`**, **`f3write`**, **`f3fix`**.

More information: https://github.com/AltraMayor/f3.

- Probe a block device:

```
sudo f3probe {{path/to/block_device}}
```

- Use the minimum about of RAM possible:

```
sudo f3probe --min-memory {{path/to/block_device}}
```

- Time disk operations:

```
sudo f3probe --time-ops {{path/to/block_device}}
```

f3read

Validate .h2w files to test the real capacity of the drive.

See also: **`f3write`**, **`f3probe`**, **`f3fix`**.

More information: https://oss.digirati.com.br/f3/.

- Validate a device by checking the files in a given directory:

```
f3read {{path/to/mount_point}}
```

f3write

Fill a drive out with .h2w files to test its real capacity.

See also: **`f3read`**, **`f3probe`**, **`f3fix`**.

More information: https://oss.digirati.com.br/f3/.

- Write test files to a given directory, filling the drive:

```
f3write {{path/to/mount_point}}
```

- Limit the write speed:

```
f3write --max-write-rate={{kb_per_second}} {{path/to/
mount_point}}
```

fabric

An open-source framework for augmenting humans using AI.

Provides a modular framework for solving specific problems using a crowdsourced set of AI prompts.

More information: https://github.com/danielmiessler/fabric#usage.

- Run the setup to configure fabric:

```
fabric {{[-S|--setup]}}
```

- List all available patterns:

```
fabric {{[-l|--listpatterns]}}
```

- Run a pattern with input from a file:

```
fabric < {{path/to/input_file}} {{[-p|--pattern]}}
{{pattern_name}}
```

- Run a pattern on a YouTube video URL:

```
fabric {{[-y|--youtube]}} "{{https://www.youtube.com/watch?
v=video_id}}" {{[-p|--pattern]}} {{pattern_name}}
```

- Chain patterns together by piping output from one to another:

```
fabric {{[-p|--pattern]}} {{pattern1}} | fabric {{[-p|--
pattern]}} {{pattern2}}
```

- Run a custom user-defined pattern:

```
fabric {{[-p|--pattern]}} {{custom_pattern_name}}
```

- Run a pattern and save the output to a file:

```
fabric {{[-p|--pattern]}} {{pattern_name}} {{[-o|--output]}}
{{path/to/output_file}}
```

- Run a pattern with the specified variables:

```
fabric {{[-p|--pattern]}} {{pattern_name}} {{[-v|--
variable]}} "{{variable_name}}:{{value}}"
```

factor

Print the prime factorization of a number.

More information: https://www.gnu.org/software/coreutils/manual/html_node/factor-invocation.html.

- Display the prime-factorization of a number:

```
factor {{number}}
```

- Take the input from `stdin` if no argument is specified:

```
echo {{number}} | factor
```

fakedata

Generate fake data using a large variety of generators.

More information: https://github.com/lucapette/fakedata.

- List all valid generators:

  ```
  fakedata --generators
  ```

- Generate data using one or more generators:

  ```
  fakedata {{generator1}} {{generator2}}
  ```

- Generate data with a specific output format:

  ```
  fakedata {{[-f|--format]}} {{csv|tab|sql}} {{generator}}
  ```

- Generate a given number of data items (defaults to 10):

  ```
  fakedata {{[-l|--limit]}} {{n}} {{generator}}
  ```

- Generate data using a custom output template (the first letter of generator names must be capitalized):

  ```
  echo "{{\{\{Generator\}\}}}" | fakedata
  ```

faker

A Python library and a tool to generate fake data.

More information: https://faker.readthedocs.io/en/master/.

- Show all fake data providers along with examples:

```
faker
```

- Generate fake data of a specific type:

```
faker {{name|address|passport_full|credit_card_full|
phone_number|email|company|date_time|user_name|password|
job|...}}
```

- Generate a number of fake addresses from a specific country (use `localectl list-locales | cut --delimiter . --fields 1` to get list of locales):

```
faker {{[-r|--repeat]}} {{number}} {{[-l|--lang]}} {{de_DE|
de_CH|...}} address
```

- Generate a number of cities in a specific country and output them to a file (use `localectl list-locales | cut --delimiter . --fields 1` to get list of locales):

```
faker {{[-r|--repeat]}} {{number}} {{[-l|--lang]}} {{en_AU|
en_US|...}} city -o {{path/to/file.txt}}
```

- Generate a number of random HTTP user-agents showing verbose output:

```
faker {{[-r|--repeat]}} {{number}} {{[-v|--verbose]}}
user_agent
```

- Generate a number of domain names and separate each using a specific separator:

```
faker {{[-r|--repeat]}} {{number}} {{[-s|--sep]}} '{{,}}'
domain_name
```

false

Returns a non-zero exit code.

More information: https://www.gnu.org/software/coreutils/manual/html_node/false-invocation.html.

- Return a non-zero exit code:

```
false
```

fast

Test your download and upload speed using fast.com.

More information: https://github.com/sindresorhus/fast-cli.

- Measure the current download speed:

```
fast
```

- Measure the current upload speed in addition to download speed:

```
fast {{[-u|--upload]}}
```

- Display results on a single line to reduce spacing:

```
fast --single-line
```

- Display complete results - download/upload speeds, latency, and ISP info:

```
fast --verbose
```

fastapi

Run FastAPI apps which uses Uvicorn under the hood.

More information: https://manned.org/fastapi.

- Run a FastAPI app with automatic reload (for development):

```
fastapi run {{path/to/file.py}} --reload
```

- Serve your app in both development mode:

```
fastapi dev {{path/to/file.py}}
```

- Specify the host and port to run on:

```
fastapi run {{path/to/file.py}} --host {{host_address}} --port {{port}}
```

- Set the app variable name (if not `app`) or specify a custom app directory:

```
fastapi run {{path/to/file.py}} --app-dir {{path/to/app}} --app {{custom_app_name}}
```

- Display help:

```
fastapi --help
```

- Display help for a subcommand:

```
fastapi {{subcommand}} --help
```

fastboot

Communicate with connected Android devices when in bootloader mode (the one place ADB doesn't work).

More information: https://cs.android.com/android/platform/superproject/+/main:system/core/fastboot.

- Unlock the bootloader:

```
fastboot oem unlock
```

- Relock the bootloader:

```
fastboot oem lock
```

- Reboot the device from fastboot mode into fastboot mode again:

```
fastboot reboot bootloader
```

- Flash a given image:

```
fastboot flash {{path/to/file.img}}
```

- Flash a custom recovery image:

```
fastboot flash recovery {{path/to/file.img}}
```

- List connected devices:

```
fastboot devices
```

- Display all information of a device:

```
fastboot getvar all
```

fastd

VPN daemon.

Works on Layer 2 or Layer 3, supports different encryption methods, used by Freifunk.

See also: **`ivpn`**, **`mozillavpn`**, **`mullvad`**, **`warp-cli`**.

More information: https://fastd.readthedocs.io/en/stable/.

- Start `fastd` with a specific configuration file:

```
fastd {{[-c|--config]}} {{path/to/fastd.conf}}
```

- Start a Layer 3 VPN with an MTU of 1400, loading the rest of the configuration parameters from a file:

```
fastd {{[-m|--mode]}} {{tap}} {{[-M|--mtu]}} {{1400}} {{[-c|--config]}} {{path/to/fastd.conf}}
```

- Validate a configuration file:

```
fastd --verify-config {{[-c|--config]}} {{path/to/fastd.conf}}
```

- Generate a new keypair:

```
fastd --generate-key
```

- Show the public key to a private key in a configuration file:

```
fastd --show-key {{[-c|--config]}} {{path/to/fastd.conf}}
```

- Display version:

```
fastd {{[-v|--version]}}
```

fastfetch

Display information about your operating system, software and hardware.

More information: https://manned.org/fastfetch.

- Display system information:

```
fastfetch
```

- Display full system information with all the modules enabled:

```
fastfetch {{[-c|--config]}} all
```

- Load a custom configuration file:

```
fastfetch {{[-c|--config]}} {{path/to/config_file}}
```

- Fetch a specific structure:

```
fastfetch {{[-s|--structure]}} {{os:kernel:de:cpu:gpu}}
```

- Use a specific logo:

```
fastfetch {{[-l|--logo]}} {{logo}}
```

- Display system information without a logo:

```
fastfetch {{[-l|--logo]}} none
```

- Use a specific color for the keys and title:

```
fastfetch --color {{blue}}
```

fastlane

Build and release mobile applications.

More information: https://docs.fastlane.tools/actions/.

- Build and sign the iOS application in the current directory:

  ```
  fastlane run build_app
  ```

- Run `pod install` for the project in the current directory:

  ```
  fastlane run cocoapods
  ```

- Delete the derived data from Xcode:

  ```
  fastlane run clear_derived_data
  ```

- Remove the cache for pods:

  ```
  fastlane run clean_cocoapods_cache
  ```

fastmod

A fast partial replacement for the codemod tool, replace and replace all in the whole codebase.

Regexes are matched by Rust regex crate.

More information: https://github.com/facebookincubator/fastmod.

- Replace a `regex` in all files of the current directory, ignoring files on .ignore and .gitignore:

```
fastmod {{regex}} {{replacement}}
```

- Replace a `regex` in case-insensitive mode in specific files or directories:

```
fastmod --ignore-case {{regex}} {{replacement}} -- {{path/to/file path/to/directory ...}}
```

- Replace a `regex` in a specific directory in files filtered with a case-insensitive glob pattern:

```
fastmod {{regex}} {{replacement}} --dir {{path/to/directory}} --iglob '{{**/*.{js,json}}}'
```

- Replace for an exact string in `.js` or JSON files:

```
fastmod --fixed-strings {{exact_string}} {{replacement}} --extensions {{json,js}}
```

- Replace for an exact string without prompt for a confirmation (disables `regex`):

```
fastmod --accept-all --fixed-strings {{exact_string}} {{replacement}}
```

- Replace for an exact string without prompt for a confirmation, printing changed files:

```
fastmod --accept-all --print-changed-files --fixed-strings {{exact_string}} {{replacement}}
```

fc-cache

Scan font directories to build font cache files.

More information: https://manned.org/fc-cache.

- Generate font cache files:

```
fc-cache
```

- Generate font cache files verbosely:

```
fc-cache {{[-v|--verbose]}}
```

- Force a rebuild of all font cache files, without checking if cache is up-to-date:

```
fc-cache {{[-f|--force]}}
```

- Erase font cache files, then generate new font cache files:

```
fc-cache {{[-r|--really-force]}}
```

- Scan a specific directory:

```
fc-cache {{path/to/directory}}
```

- Scan system-wide directories, skipping the user's home directory:

```
fc-cache {{[-s|--system-only]}}
```

- Display version:

```
fc-cache {{[-V|--version]}}
```

fc-list

List available fonts installed on the system.

More information: https://manned.org/fc-list.

- Return a list of installed fonts:

```
fc-list
```

- Return a list of installed fonts with given name:

```
fc-list | grep '{{DejaVu Serif}}'
```

- Return the number of installed fonts:

```
fc-list | wc {{[-l|--lines]}}
```

- Return a list of installed fonts that support the language based on its locale code:

```
fc-list :lang={{jp}}
```

- Return a list of installed fonts that contain the glyph specified by its Unicode code-point:

```
fc-list :charset={{f303}}
```

fc-match

Match available fonts.

More information: https://manned.org/fc-match.

- Return a sorted list of best matching fonts:

```
fc-match {{[-s|--sort]}} '{{DejaVu Serif}}'
```

fc-pattern

Shows information about a font matching a pattern.

More information: https://manned.org/fc-pattern.

- Display default information about a font:

```
fc-pattern {{[-d|--default]}} '{{DejaVu Serif}}'
```

- Display configuration information about a font:

```
fc-pattern {{[-c|--config]}} '{{DejaVu Serif}}'
```

fc

Open the most recent command for editing and then run it.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-fc.

- Open the last command in the default system editor and run it after editing:

```
fc
```

- Specify an editor to open with:

```
fc -e '{{emacs}}'
```

- List recent commands from history:

```
fc -l
```

- List recent commands in reverse order:

```
fc -l -r
```

- Edit and run a command from history:

```
fc {{number}}
```

- Edit commands in a given interval and run them:

```
fc '{{416}}' '{{420}}'
```

- Display help:

```
fc --help
```

fclones

Efficient duplicate file finder and remover.

More information: https://github.com/pkolaczk/fclones.

- Search for duplicate files in the current directory:

```
fclones group .
```

- Search multiple directories for duplicate files and cache the results:

```
fclones group --cache {{path/to/directory1 path/to/
directory2 ...}}
```

- Search only the specified directory for duplicate files, skipping subdirectories and save the results into a file:

```
fclones group {{path/to/directory}} --depth 1 > {{path/to/
file.txt}}
```

- Move the duplicate files in a TXT file to a different directory:

```
fclones < {{path/to/file.txt}} move {{path/to/
target_directory}}
```

- Perform a dry run for soft links in a TXT file without actually linking:

```
fclones < {{path/to/file.txt}} link --soft --dry-run 2 > /
dev/null
```

- Delete the newest duplicates from the current directory without storing them in a file:

```
fclones group . | fclones remove --priority newest
```

- Preprocess JPEG files in the current directory by using an external command to strip their EXIF data before matching for duplicates:

```
fclones group . --name '*.jpg' -i --transform 'exiv2 -d a
$IN' --in-place
```

fd

An alternative to **`find`**.

Aims to be faster and easier to use than **`find`**.

More information: https://github.com/sharkdp/fd#how-to-use.

- Recursively find files matching a specific pattern in the current directory:

```
fd "{{string|regex}}"
```

- Find files that begin with a specific string:

```
fd "{{^string}}"
```

- Find files with a specific extension:

```
fd {{[-e|--extension]}} {{txt}}
```

- Find files in a specific directory:

```
fd "{{string|regex}}" {{path/to/directory}}
```

- Include ignored and hidden files in the search:

```
fd {{[-H|--hidden]}} {{[-I|--no-ignore]}} "{{string|regex}}"
```

- Execute a command on each search result returned:

```
fd "{{string|regex}}" {{[-x|--exec]}} {{command}}
```

fdp

Render an image of a **`force-directed`** network graph from a **`graphviz`** file.

Layouts: **`dot`**, **`neato`**, **`twopi`**, **`circo`**, **`fdp`**, **`sfdp`**, **`osage`** & **`patchwork`**.

More information: https://graphviz.org/doc/info/command.html.

- Render a PNG image with a filename based on the input filename and output format (uppercase -O):

```
fdp -T png -O {{path/to/input.gv}}
```

- Render a SVG image with the specified output filename (lowercase -o):

```
fdp -T svg -o {{path/to/image.svg}} {{path/to/input.gv}}
```

- Render the output in a specific format:

```
fdp -T {{ps|pdf|svg|fig|png|gif|jpg|json|dot}} -O {{path/to/input.gv}}
```

- Render a `gif` image using `stdin` and `stdout`:

```
echo "{{digraph {this -> that} }}" | fdp -T gif > {{path/to/image.gif}}
```

- Display help:

```
fdp -?
```

fdroid

F-Droid build tool.

F-Droid is an installable catalog of FOSS (Free and Open Source Software) applications for the Android platform.

More information: https://f-droid.org/en/docs/Building_Applications/.

- Build a specific app:

  ```
  fdroid build {{app_id}}
  ```

- Build a specific app in a build server VM:

  ```
  fdroid build {{app_id}} --server
  ```

- Publish the app to the local repository:

  ```
  fdroid publish {{app_id}}
  ```

- Install the app on every connected device:

  ```
  fdroid install {{app_id}}
  ```

- Check if the metadata is formatted correctly:

  ```
  fdroid lint --format {{app_id}}
  ```

- Fix the formatting automatically (if possible):

  ```
  fdroid rewritemeta {{app_id}}
  ```

fdroidcl

Manage F-Droid apps of devices connected via ADB.

More information: https://github.com/mvdan/fdroidcl.

- Fetch the F-Droid index:

```
fdroidcl update
```

- Display information about an app:

```
fdroidcl show {{app_id}}
```

- Download the APK file of an app:

```
fdroidcl download {{app_id}}
```

- Search for an app in the index:

```
fdroidcl search {{search_pattern}}
```

- Install an app on a connected device:

```
fdroidcl install {{app_id}}
```

- Add a repository:

```
fdroidcl repo add {{repo_name}} {{url}}
```

- Remove, enable or disable a repository:

```
fdroidcl repo {{remove|enable|disable}} {{repo_name}}
```

fdupes

Finds duplicate files in a set of directories.

More information: https://github.com/adrianlopezroche/fdupes.

- Search a single directory:

```
fdupes {{path/to/directory}}
```

- Search multiple directories:

```
fdupes {{directory1}} {{directory2}}
```

- Search a directory recursively:

```
fdupes {{[-r|--recurse]}} {{path/to/directory}}
```

- Search multiple directories, one recursively:

```
fdupes {{path/to/irectory1}} {{[-R|--recurse:]}} {{path/to/directory2}}
```

- Search recursively, considering hardlinks as duplicates:

```
fdupes {{[-rH|--recurse --hardlinks]}} {{path/to/directory}}
```

- Search recursively for duplicates and display interactive prompt to pick which ones to keep, deleting the others:

```
fdupes {{[-rd|--recurse --delete]}} {{path/to/directory}}
```

- Search recursively and delete duplicates without prompting:

```
fdupes {{[-rdN|--recurse --delete --noprompt]}} {{path/to/directory}}
```

feh

Lightweight image viewing utility.

More information: https://man.finalrewind.org/1/feh/.

- View images locally or using a URL:

```
feh {{path/to/images}}
```

- View images recursively:

```
feh {{[-r|--recursive]}} {{path/to/images}}
```

- View images and display the file name at the top-left of the images:

```
feh {{[-d|--draw-filename]}} {{path/to/images}}
```

- View images without window borders:

```
feh {{[-x|--borderless]}} {{path/to/images}}
```

- Set the behavior when reaching the beginning or end of the image list:

```
feh --on-last-slide {{hold|quit|resume}} {{path/to/images}}
```

- Use a specific slideshow cycle delay:

```
feh {{[-D|--slideshow-delay]}} {{seconds}} {{path/to/images}}
```

- Use a specific wallpaper mode (centered, filled, maximized, scaled or tiled):

```
feh --bg-{{center|fill|max|scale|tile}} {{path/to/image}}
```

- Create a montage of all images within a directory, outputting as a new image:

```
feh {{[-m|--montage]}} {{[-E|--thumb-height]}} {{150}} {{[-
y|--thumb-width]}} {{150}} --index-info "{{%nn%wx%h}}" {{[-
o|--output]}} {{path/to/montage_image.png}}
```

feroxbuster

Simple, fast, recursive content discovery tool written in Rust.

Used to brute-force hidden paths on web servers and more.

More information: https://epi052.github.io/feroxbuster-docs/docs/.

- Discover specific directories and files that match in the wordlist with extensions and 100 threads and a random user-agent:

```
feroxbuster --url "{{https://example.com}}" --wordlist
{{path/to/file}} --threads {{100}} --extensions "{{php,txt}}"
--random-agent
```

- Enumerate directories without recursion through a specific proxy:

```
feroxbuster --url "{{https://example.com}}" --wordlist
{{path/to/file}} --no-recursion --proxy "{{http://
127.0.0.1:8080}}"
```

- Find links in webpages:

```
feroxbuster --url "{{https://example.com}}" --extract-links
```

- Filter by a specific status code and a number of chars:

```
feroxbuster --url "{{https://example.com}}" --filter-status
{{301}} --filter-size {{4092}}
```

ffe

Extract fields from a flat database file and write to another format.

A configuration file is required to interpret the input and format the output.

More information: https://ff-extractor.sourceforge.net/ffe.html.

- Display all input data using the specified data configuration:

```
ffe {{[-c|--configuration]}} {{path/to/config.ffe}} {{path/to/input}}
```

- Convert an input file to an output file in a new format:

```
ffe --output={{path/to/output}} {{[-c|--configuration]}} {{path/to/config.ffe}} {{path/to/input}}
```

- Select input structure and print format from definitions in `~/.fferc` configuration file:

```
ffe {{[-s|--structure]}} {{structure}} {{[-p|--print]}} {{format}} {{path/to/input}}
```

- Write only the selected fields:

```
ffe {{[-f|--field-list]}} "{{FirstName,LastName,Age}}" {{[-c|--configuration]}} {{path/to/config.ffe}} {{path/to/input}}
```

- Write only the records that match an expression:

```
ffe {{[-e|--expression]}} "{{LastName=Smith}}" {{[-c|--configuration]}} {{path/to/config.ffe}} {{path/to/input}}
```

- Display help:

```
ffe {{[-?|--help]}}
```

ffmpeg

Video conversion tool.

See also: **`gst-launch-1.0`**.

More information: https://ffmpeg.org/ffmpeg.html#Options.

- Extract the sound from a video and save it as MP3:

```
ffmpeg -i {{path/to/video.mp4}} -vn {{path/to/sound.mp3}}
```

- Transcode a FLAC file to Red Book CD format (44100kHz, 16bit):

```
ffmpeg -i {{path/to/input_audio.flac}} -ar 44100 -sample_fmt
s16 {{path/to/output_audio.wav}}
```

- Save a video as GIF, scaling the height to 1000px and setting framerate to 15:

```
ffmpeg -i {{path/to/video.mp4}} {{[-vf|-filter:v]}}
'scale=-1:{{1000}}' -r {{15}} {{path/to/output.gif}}
```

- Combine numbered images (`frame_1.jpg`, `frame_2.jpg`, etc) into a video or GIF:

```
ffmpeg -i {{path/to/frame_%d.jpg}} -f image2 {{video.mpg|
video.gif}}
```

- Trim a video from a given start time mm:ss to an end time mm2:ss2 (omit the -to flag to trim till the end):

```
ffmpeg -i {{path/to/input_video.mp4}} -ss {{mm:ss}} -to
{{mm2:ss2}} {{[-c|-codec]}} copy {{path/to/output_video.mp4}}
```

- Convert AVI video to MP4. AAC Audio @ 128kbit, h264 Video @ CRF 23:

```
ffmpeg -i {{path/to/input_video}}.avi {{[-c|-codec]}}:a aac -
b:a 128k {{[-c|-codec]}}:v libx264 -crf 23 {{path/to/
output_video}}.mp4
```

- Remux MKV video to MP4 without re-encoding audio or video streams:

```
ffmpeg -i {{path/to/input_video}}.mkv {{[-c|-codec]}} copy
{{path/to/output_video}}.mp4
```

- Convert MP4 video to VP9 codec. For the best quality, use a CRF value (recommended range 15-35) and -b:v MUST be 0:

```
ffmpeg -i {{path/to/input_video}}.mp4 {{[-c|-codec]}}:v
libvpx-vp9 -crf {{30}} -b:v 0 {{[-c|-codec]}}:a libopus -vbr
on -threads {{number_of_threads}} {{path/to/
output_video}}.webm
```

ffplay

A simple and portable media player using the FFmpeg libraries and the SDL library.

More information: https://ffmpeg.org/ffplay-all.html.

- Play a media file:

```
ffplay {{path/to/file}}
```

- Play audio from a media file without a GUI:

```
ffplay -nodisp {{path/to/file}}
```

- Play media passed by `ffmpeg` through `stdin`:

```
ffmpeg -i {{path/to/file}} -c {{copy}} -f {{media_format}} -
| ffplay -
```

- Play a video and show motion vectors in real time:

```
ffplay -flags2 +export_mvs -vf codecview=mv=pf+bf+bb {{path/
to/file}}
```

- Show only video keyframes:

```
ffplay -vf select="{{eq(pict_type\,PICT_TYPE_I)}}" {{path/to/
file}}
```

ffprobe

Multimedia stream analyzer.

More information: https://ffmpeg.org/ffprobe.html.

- Display all available stream info for a media file:

```
ffprobe {{[-v|-loglevel]}} error -show_streams {{input.mp4}}
```

- Display media duration:

```
ffprobe {{[-v|-loglevel]}} error -show_entries
format=duration {{[-of|-output_format]}}
default=noprint_wrappers=1:nokey=1 {{input.mp4}}
```

- Display the frame rate of a video:

```
ffprobe {{[-v|-loglevel]}} error -select_streams v:0 -
show_entries stream=avg_frame_rate {{[-of|-output_format]}}
default=noprint_wrappers=1:nokey=1 {{input.mp4}}
```

- Display the width or height of a video:

```
ffprobe {{[-v|-loglevel]}} error -select_streams v:0 -
show_entries stream={{width|height}} {{[-of|-output_format]}}
default=noprint_wrappers=1:nokey=1 {{input.mp4}}
```

- Display the average bit rate of a video:

```
ffprobe {{[-v|-loglevel]}} error -select_streams v:0 -
show_entries stream=bit_rate {{[-of|-output_format]}}
default=noprint_wrappers=1:nokey=1 {{input.mp4}}
```

ffsend

Easily and securely share files.

More information: https://gitlab.com/timvisee/ffsend.

- Upload a file:

```
ffsend upload {{path/to/file}}
```

- Download a file:

```
ffsend download {{url}}
```

- Upload a file with password:

```
ffsend upload {{path/to/file}} {{[-p|--password]}}
{{password}}
```

- Download a file protected by password:

```
ffsend download {{url}} {{[-p|--password]}} {{password}}
```

- Upload a file and allow 4 downloads:

```
ffsend upload {{path/to/file}} {{[-d|--downloads]}} {{4}}
```

ffuf

A fast web fuzzer written in Go.

The **FUZZ** keyword is used as a placeholder. **`ffuf`** will try to hit the URL by replacing the word **FUZZ** with every word in the wordlist.

More information: https://github.com/ffuf/ffuf#usage.

- Enumerate directories using [c]olored output and a [w]ordlist specifying a target [u]RL:

```
ffuf -c -w {{path/to/wordlist.txt}} -u {{http://example.com/FUZZ}}
```

- Enumerate webservers of subdomains by changing the position of the keyword:

```
ffuf -w {{path/to/subdomains.txt}} -u {{http://FUZZ.example.com}}
```

- Fuzz with specified [t]hreads (default: 40) and pro[x]ying the traffic and save [o]utput to a file:

```
ffuf -o -w {{path/to/wordlist.txt}} -u {{http://example.com/FUZZ}} -t {{500}} -x {{http://127.0.0.1:8080}}
```

- Fuzz a specific [H]eader ("Name: Value") and [m]atch HTTP status [c]odes:

```
ffuf -w {{path/to/wordlist.txt}} -u {{http://example.com}} -H "{{Host: FUZZ}}" -mc {{200}}
```

- Fuzz with specified HTTP method and [d]ata, while [f]iltering out comma separated status [c]odes:

```
ffuf -w {{path/to/postdata.txt}} -X {{POST}} -d "{{username=admin\&password=FUZZ}}" -u {{http://example.com/login.php}} -fc {{401,403}}
```

- Fuzz multiple positions with multiple wordlists using different modes:

```
ffuf -w {{path/to/keys:KEY}} -w {{path/to/values:VALUE}} -mode {{pitchfork|clusterbomb}} -u {{http://example.com/id?KEY=VALUE}}
```

- Proxy requests through a HTTP MITM pro[x]y (such as Burp Suite or `mitmproxy`):

```
ffuf -w {{path/to/wordlist}} -x {{http://127.0.0.1:8080}} -u
{{http://example.com/FUZZ}}
```

fg

Run jobs in foreground.

See also: **`jobs`**, **`bg`**, **`disown`**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-fg.

- Bring most recently suspended or running background job to foreground:

```
fg
```

- Bring a specific job to foreground (run `jobs` to find the job number):

```
fg %{{job_number}}
```

fgrep

Matches fixed strings in files.

Equivalent to **`grep -F`**.

More information: https://www.gnu.org/software/grep/manual/grep.html.

- Search for an exact string in a file:

```
fgrep {{search_string}} {{path/to/file}}
```

- Search only lines that match entirely in one or more files:

```
fgrep {{[-x|--line-regexp]}} {{search_string}} {{path/to/
file1 path/to/file2 ...}}
```

- Count the number of lines that match the given string in a file:

```
fgrep {{[-c|--count]}} {{search_string}} {{path/to/file}}
```

- Show the line number in the file along with the line matched:

```
fgrep {{[-n|--line-number]}} {{search_string}} {{path/to/
file}}
```

- Display all lines except those that contain the search string:

```
fgrep {{[-v|--invert-match]}} {{search_string}} {{path/to/
file}}
```

- Display filenames whose content matches the search string at least once:

```
fgrep {{[-l|--files-with-matches]}} {{search_string}} {{path/
to/file1 path/to/file2 ...}}
```

fi

This shell keyword is used to end an **`if`** statement.

- View documentation for the `if` keyword:

  ```
  tldr if
  ```

fiascotopnm

Convert a compressed FIASCO file to a PNM image.

More information: https://netpbm.sourceforge.net/doc/fiascotopnm.html.

- Convert a compressed FIASCO file to a PNM file or in the case of video streams multiple PNM files:

```
fiascotopnm {{path/to/file.fiasco}} {{[-o|--output]}} {{output_file_basename}}
```

- Use fast decompression, resulting in a slightly decreased quality of the output file(s):

```
fiascotopnm {{[-z|--fast]}} {{path/to/file.fiasco}} {{[-o|--output]}} {{output_file_basename}}
```

- Load the options to be used from the specified configuration file:

```
fiascotopnm {{[-f|--config]}} {{path/to/fiascorc}} {{path/to/file.fiasco}} {{[-o|--output]}} {{output_file_basename}}
```

- Magnify the decompressed image(s) by a factor of 2^n:

```
fiascotopnm {{[-m|--magnify]}} {{n}} {{path/to/file.fiasco}} {{[-o|--output]}} {{output_file_basename}}
```

- Smooth the decompressed image by the specified amount:

```
fiascotopnm {{[-s|--smoothing]}} {{n}} {{path/to/file.fiasco}} {{[-o|--output]}} {{output_file_basename}}
```

figlet

Generate ASCII banners from user input.

See also: **`showfigfonts`**.

More information: http://www.figlet.org/figlet-man.html.

- Generate by directly inputting text:

```
figlet {{input_text}}
```

- Use a custom [f]ont file:

```
figlet -f {{path/to/font_file.flf}} {{input_text}}
```

- Use a [f]ont from the default font directory (the extension can be omitted):

```
figlet -f {{font_filename}} {{input_text}}
```

- Pipe command output through FIGlet:

```
{{command}} | figlet
```

- Show available FIGlet fonts:

```
showfigfonts {{optional_string_to_display}}
```

- Use the full width of the [t]erminal and [c]enter the input text:

```
figlet -t -c {{input_text}}
```

- Display all characters at full [W]idth to avoid overlapping:

```
figlet -W {{input_text}}
```

figlist

List the figlet fonts and control files.

See also: **`figlet`**, **`showfigfonts`**, **`chkfont`**.

More information: https://manned.org/figlist.

- List all available fonts using the default font directory:

```
figlist
```

- List fonts from a custom directory:

```
figlist -d {{path/to/directory}}
```

- Search for a font by keyword:

```
figlist -d {{path/to/directory}} | grep {{keyword}}
```

- Count the total number of available fonts in a specified directory:

```
figlist -d {{path/to/directory}} | wc {{[-l|--lines]}}
```

file-rename

This command is an alias of **`rename`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common rename
```

file

Determine file type.

See also: **`stat`**.

More information: https://manned.org/file.

- Give a description of the type of the specified file:

```
file {{path/to/file}}
```

- Look inside a zipped file and determine the file type(s) inside:

```
file {{[-z|--uncompress]}} {{path/to/file.zip}}
```

- Allow `file` to work with special or device files:

```
file {{[-s|--special-files]}} {{path/to/file}}
```

- Don't stop at first file type match; keep going until the end of the file:

```
file {{[-k|--keep-going]}} {{path/to/file}}
```

- Determine the MIME encoding type of a file:

```
file {{[-i|--mime]}} {{path/to/file}}
```

filebrowser

Simple HTTP web server to manage files and directories.

More information: https://filebrowser.org.

- Start a new server instance serving the current directory:

```
filebrowser
```

- Start a new server instance serving a specific root directory:

```
filebrowser {{[-r|--root]}} {{path/to/directory}}
```

- Start an instance with different host address (defaults to `127.0.0.1`) and port (defaults to `8080`):

```
filebrowser {{[-a|--address]}} {{host}} {{[-p|--port]}} {{port}} {{[-r|--root]}} {{path/to/directory}}
```

- Start an instance with a specified configuration file, storing the application database in a specific location (defaults to `filebrowser.db` on the current directory):

```
filebrowser {{[-c|--config]}} {{path/to/file}} {{[-d|--database]}} {{path/to/database.db}} {{[-r|--root]}} {{path/to/directory}}
```

- Set up a different default first-time account username and password (both default to `admin`) when setting up a new instance:

```
filebrowser --username {{username}} --password {{password}} {{[-r|--root]}} {{path/to/directory}}
```

- Set up the maximum amount of image processors used when generating thumbnails (defaults to `4`):

```
filebrowser --img-processors {{4}} {{[-r|--root]}} {{path/to/directory}}
```

- Disable image thumbnails as well as the Command Runner feature, limiting access for hosted script files from being executed inside the app:

```
filebrowser --disable-exec --disable-thumbnails {{[-r|--root]}} {{path/to/directory}}
```

- Disable resizing of image previews as well as detecting file types by reading their headers:

```
filebrowser --disable-preview-resize --disable-type-
detection-by-header {{[-r|--root]}} {{path/to/directory}}
```

FileCheck

Flexible pattern matching file verifier.

It is typically used from LLVM regression tests and forms a part of a **`lit`** test.

More information: https://llvm.org/docs/CommandGuide/FileCheck.html.

- Match `input_file` content with pattern file `check_file`:

```
FileCheck --input-file={{path/to/input_file}} {{path/to/check_file}}
```

- Match input from the `stdin` with pattern file `check_file`:

```
echo "{{some_text}}" | FileCheck {{path/to/check_file}}
```

- Match with the specified custom check `prefix` (Note: The default prefix is `CHECK`):

```
echo "{{some_text}}" | FileCheck --check-prefix={{prefix}} {{path/to/check_file}}
```

- Print good directive pattern matches:

```
echo "{{some_text}}" | FileCheck -v {{path/to/check_file}}
```

- Input `llvm_code.ll` into llvm-as, then pipe the output into FileCheck to match:

```
llvm-as {{path/to/llvm_code.ll}} | FileCheck {{path/to/check_file}}
```

filen

Interface with Filen, an end-to-end encrypted cloud storage service.

More information: https://github.com/FilenCloudDienste/filen-cli.

- Enter interactive mode:

```
filen
```

- Upload a local file to a specific remote folder:

```
filen upload {{path/to/local_file}} {{remote_folder_id}}
```

- Download a file or folder using its remote ID:

```
filen download {{remote_id}} {{path/to/local_destination}}
```

- List files and folders inside a remote folder:

```
filen ls {{remote_folder}}
```

- Delete a remote file or folder (move it to trash):

```
filen rm {{remote_id}}
```

- Restore a trashed item:

```
filen trash restore {{remote_id}}
```

- Synchronize a local folder with a remote folder (two-way sync):

```
filen sync {{path/to/local_folder}}:/{{remote_folder}} --continuous
```

- Download changes from the cloud to a local folder (one-way sync):

```
filen sync {{path/to/local_folder}}:ctl:/{{remote_folder}}
```

fin

Docksal utility.

More information: https://docs.docksal.io/fin/fin/.

- Start the project in the current directory:

```
fin project start
```

- Stop the project in the current directory:

```
fin project stop
```

- Open a shell into a specific container:

```
fin bash {{container_name}}
```

- Display logs of a specific container:

```
fin logs {{container_name}}
```

- Display logs of a specific container and follow the log:

```
fin logs {{[-f|--follow]}} {{container_name}}
```

find

Find files or directories under a directory tree, recursively.

See also: **fd**.

More information: https://manned.org/find.

- Find files by extension:

```
find {{root_path}} -name '{{*.ext}}'
```

- Find files matching multiple path/name patterns:

```
find {{root_path}} -path '{{*/path/*/*.ext}}' -or -name
'{{*pattern*}}'
```

- Find directories matching a given name, in case-insensitive mode:

```
find {{root_path}} -type d -iname '{{*lib*}}'
```

- Find files matching a given pattern, excluding specific paths:

```
find {{root_path}} -name '{{*.py}}' -not -path '{{*/site-
packages/*}}'
```

- Find files matching a given size range, limiting the recursive depth to "1":

```
find {{root_path}} -maxdepth 1 -size {{+500k}} -size {{-10M}}
```

- Run a command for each file (use `{}` within the command to access the filename):

```
find {{root_path}} -name '{{*.ext}}' -exec {{wc -l}} {} \;
```

- Find all files modified today and pass the results to a single command as arguments:

```
find {{root_path}} -daystart -mtime {{-1}} -exec {{tar -cvf
archive.tar}} {} \+
```

- Search for either empty files or directories and delete them verbosely:

```
find {{root_path}} -type {{f|d}} -empty -delete -print
```

finger

User information lookup program.

More information: https://manned.org/finger.

- Display information about currently logged in users:

  ```
  finger
  ```

- Display information about a specific user:

  ```
  finger {{username}}
  ```

- Display the user's login name, real name, terminal name, and other information:

  ```
  finger -s
  ```

- Produce multiline output format displaying same information as `-s` as well as user's home directory, home phone number, login shell, mail status, etc.:

  ```
  finger -l
  ```

- Prevent matching against user's names and only use login names:

  ```
  finger -m
  ```

fio

Flexible I/O tester: do an I/O action spawning multiple threads or processes.

More information: https://fio.readthedocs.io/en/latest/fio_doc.html.

- Test random reads:

```
fio --filename={{path/to/file}} --direct=1 --rw=randread --
bs=4k --ioengine=libaio --iodepth=256 --runtime=120 --
numjobs=4 --time_based --group_reporting --name={{job_name}}
--eta-newline=1 --readonly
```

- Test sequential reads:

```
fio --filename={{path/to/file}} --direct=1 --rw=read --bs=4k
--ioengine=libaio --iodepth=256 --runtime=120 --numjobs=4 --
time_based --group_reporting --name={{job_name}} --eta-
newline=1 --readonly
```

- Test random read/write:

```
fio --filename={{path/to/file}} --direct=1 --rw=randrw --
bs=4k --ioengine=libaio --iodepth=256 --runtime=120 --
numjobs=4 --time_based --group_reporting --name={{job_name}}
--eta-newline=1
```

- Test with parameters from a job file:

```
fio {{path/to/job_file}}
```

- Convert a specific job file to command-line options:

```
fio --showcmd {{path/to/job_file}}
```

firebase

Test, manage, and deploy Firebase projects.

More information: https://github.com/firebase/firebase-tools.

- Log in to https://console.firebase.google.com:

  ```
  firebase login
  ```

- List existing Firebase projects:

  ```
  firebase projects:list
  ```

- Start an interactive wizard to create a Firebase project in the current directory:

  ```
  firebase init
  ```

- Deploy code and assets to the current Firebase project:

  ```
  firebase deploy
  ```

- Start a local server to statically host the current Firebase project's assets:

  ```
  firebase serve
  ```

- Start an interactive wizard to open one of many links of the current Firebase project in the default web browser:

  ```
  firebase open
  ```

firefox

A free and open source web browser.

More information: https://wiki.mozilla.org/Firefox/CommandLineOptions.

- Launch Firefox and open a web page:

```
firefox {{https://www.duckduckgo.com}}
```

- Open a new window:

```
firefox --new-window {{https://www.duckduckgo.com}}
```

- Open a private (incognito) window:

```
firefox --private-window
```

- Search for "wikipedia" using the default search engine:

```
firefox --search "{{wikipedia}}"
```

- Launch Firefox in safe mode, with all extensions disabled:

```
firefox --safe-mode
```

- Take a screenshot of a web page in headless mode:

```
firefox --headless --screenshot {{path/to/output_file.png}} {{https://example.com/}}
```

- Use a specific profile to allow multiple separate instances of Firefox to run at once:

```
firefox --profile {{path/to/directory}} {{https://example.com/}}
```

- Set Firefox as the default browser:

```
firefox --setDefaultBrowser
```

fish

The Friendly Interactive SHell, a command-line interpreter designed to be user friendly.

More information: https://fishshell.com/docs/current/cmds/fish.html.

- Start an interactive shell session:

```
fish
```

- Start an interactive shell session without loading startup configs:

```
fish {{[-N|--no-config]}}
```

- Execute specific commands:

```
fish {{[-c|--command]}} "{{echo 'fish is executed'}}"
```

- Execute a specific script:

```
fish {{path/to/script.fish}}
```

- Check a specific script for syntax errors:

```
fish {{[-N|--no-execute]}} {{path/to/script.fish}}
```

- Execute specific commands from `stdin`:

```
{{echo "echo 'fish is executed'"}} | fish
```

- Start an interactive shell session in private mode, where the shell does not access old history or save new history:

```
fish {{[-P|--private]}}
```

- Define and export an environmental variable that persists across shell restarts (builtin):

```
set {{[-U|--universal]}} {{[-x|--export]}} {{variable_name}} {{variable_value}}
```

fisher

Fisher, a fish-shell plugin manager.

Install plugins by name or from a managed **`fishfile`** for bundled installs.

More information: https://github.com/jorgebucaran/fisher.

- Install one or more plugins:

```
fisher {{plugin1}} {{plugin2}}
```

- Install a plugin from a GitHub gist:

```
fisher {{gist_url}}
```

- Edit `fishfile` manually with your favorite editor and install multiple plugins:

```
{{editor}} ~/.config/fish/fishfile; fisher
```

- List installed plugins:

```
fisher ls
```

- Update plugins:

```
fisher update
```

- Remove one or more plugins:

```
fisher remove {{plugin1}} {{plugin2}}
```

fitstopnm

Convert a Flexible Image Transport System (FITS) file to a PNM image.

See also: **pamtofits**.

More information: https://netpbm.sourceforge.net/doc/fitstopnm.html.

- Convert a FITS file to a PNM image:

```
fitstopnm {{path/to/file.fits}} > {{path/to/output.pnm}}
```

- Convert the image on the specified position of the third axis in the FITS file:

```
fitstopnm {{[-i|-image]}} {{z_position}} {{path/to/
file.fits}} > {{path/to/output.pnm}}
```

fkill

Fabulously kill processes. Cross-platform.

More information: https://github.com/sindresorhus/fkill.

- Run without arguments to use the interactive interface:

```
fkill
```

- Kill the process by PID, name or port:

```
fkill {{pid|name|:port}}
```

flac

Encodes, decodes and tests FLAC files.

More information: https://xiph.org/flac.

- Encode a WAV file to FLAC (this will create a FLAC file in the same location as the WAV file):

```
flac {{path/to/file.wav}}
```

- Encode a WAV file to FLAC, specifying the output file:

```
flac {{[-o|--output-name]}} {{path/to/output.flac}} {{path/to/file.wav}}
```

- Decode a FLAC file to WAV, specifying the output file:

```
flac {{[-d|--decode]}} {{[-o|--output-name]}} {{path/to/output.wav}} {{path/to/file.flac}}
```

- Test a FLAC file for the correct encoding:

```
flac {{[-t|--test]}} {{path/to/file.flac}}
```

flake8

Check the style and quality of Python code.

More information: https://flake8.pycqa.org/en/latest/user/options.html.

- Lint a file or directory recursively:

```
flake8 {{path/to/file_or_directory}}
```

- Lint a file or directory recursively and show the line on which each error occurred:

```
flake8 --show-source {{path/to/file_or_directory}}
```

- Lint a file or directory recursively and ignore a list of rules. (All available rules can be found at https://www.flake8rules.com/):

```
flake8 --ignore {{rule1,rule2,...}} {{path/to/
file_or_directory}}
```

- Lint a file or directory recursively but exclude files matching the given globs or substrings:

```
flake8 --exclude {{substring1,glob2}} {{path/to/
file_or_directory}}
```

flarectl

Official CLI for Cloudflare.

More information: https://github.com/cloudflare/cloudflare-go/blob/master/cmd/flarectl/README.md.

- Block a specific IP:

```
flarectl firewall rules create --zone="{{example.com}}" --value="{{8.8.8.8}}" --mode="{{block}}" --notes="{{Block bad actor}}"
```

- Add a DNS record:

```
flarectl dns create --zone="{{example.com}}" --name="{{app}}" --type="{{CNAME}}" --content="{{myapp.herokuapp.com}}" --proxy
```

- List all Cloudflare IPv4/IPv6 ranges:

```
flarectl ips --ip-type {{ipv4|ipv6|all}}
```

- Create many new Cloudflare zones automatically with names from `domains.txt`:

```
for domain in $(cat {{domains.txt}}); do flarectl zone info --zone=$domain; done
```

- List all firewall rules:

```
flarectl firewall rules list
```

flask-unsign

A tool to brute-force, decode and craft **`Flask`** session cookies.

More information: https://github.com/Paradoxis/Flask-Unsign.

- Decode a Flask session cookie:

```
flask-unsign {{[-d|--decode]}} {{[-c|--cookie]}} {{cookie}}
```

- Decode a session cookie fetched from a URL which returns a `Set-Cookie` header:

```
flask-unsign {{[-d|--decode]}} --server {{URL}}
```

- Brute-force a secret key using the default flask-unsign-wordlist (requires `flask-unsign-wordlist`):

```
flask-unsign {{[-u|--unsign]}} {{[-c|--cookie]}} {{cookie}}
```

- Brute-force a secret key with a custom wordlist (use `--no-literal-eval` for unquoted entries):

```
flask-unsign {{[-u|--unsign]}} {{[-c|--cookie]}} {{cookie}}
{{[-w|--wordlist]}} {{path/to/wordlist.txt}}
```

- Sign a new session cookie with a secret key:

```
flask-unsign {{[-s|--sign]}} {{[-c|--cookie]}}
"{{{'logged_in': False}}}" {{[-S|--secret]}} {{secret}}
```

- Sign a session cookie using legacy timestamp (useful for old versions):

```
flask-unsign {{[-s|--sign]}} {{[-c|--cookie]}}
"{{{'logged_in': False}}}" {{[-S|--secret]}} {{secret}} {{[-
l|--legacy]}}
```

- Brute-force a session cookie with custom threads and no literal evaluation:

```
flask-unsign {{[-u|--unsign]}} {{[-c|--cookie]}} {{cookie}}
{{[-w|--wordlist]}} {{path/to/wordlist.txt}} {{[-t|--
threads]}} {{threads}} {{[-nE|--no-literal-eval]}}
```

flask

A general utility script for Flask applications. Loads the application defined in the **`FLASK_APP`** environment variable.

More information: https://flask.palletsprojects.com/en/stable/cli/.

- Run a development server:

  ```
  flask run
  ```

- Show the routes for the app:

  ```
  flask routes
  ```

- Run a Python interactive shell in the app's context:

  ```
  flask shell
  ```

flexget

A multipurpose automation tool for content like torrents, nzbs, podcasts, comics, series, movies, etc.

More information: https://flexget.com/en/CLI.

- Run all Flexget tasks now:

```
flexget execute --now
```

- Start the Flexget daemon and daemonize its process:

```
flexget daemon start --daemonize
```

- List all series recorded in Flexget:

```
flexget series list
```

- Run a task from a configuration file:

```
flexget -c {{path/to/config.yml}} execute --task
{{task_name}}
```

flips

Create and apply patches for IPS and BPS files.

More information: https://github.com/Alcaro/Flips.

- Start Flips to create and apply patches interactively:

```
flips
```

- Apply a patch and create a new ROM file:

```
flips --apply {{patch.bps}} {{rom.smc}} {{hack.smc}}
```

- Create a patch from two ROMs:

```
flips --create {{rom.smc}} {{hack.smc}} {{patch.bps}}
```

flite

Speech synthesis engine.

More information: http://www.festvox.org/flite/doc/.

- List all available voices:

```
flite -lv
```

- Convert a text string to speech:

```
flite -t "{{string}}"
```

- Convert the contents of a file to speech:

```
flite -f {{path/to/file.txt}}
```

- Use the specified voice:

```
flite -voice {{file://path/to/filename.flitevox|url}}
```

- Store output into a wav file:

```
flite -voice {{file://path/to/filename.flitevox|url}} -f
{{path/to/file.txt}} -o {{output.wav}}
```

- Display version:

```
flite --version
```

flow

A static type checker for JavaScript.

More information: https://flow.org/en/docs/cli/.

- Run a flow check:

```
flow
```

- Check which files are being checked by flow:

```
flow ls
```

- Run a type coverage check on all files in a directory:

```
flow batch-coverage --show-all --strip-root {{path/to/
directory}}
```

- Display line-by-line type coverage stats:

```
flow coverage --color {{path/to/file.jsx}}
```

flox

Easy to use Nix package and environment manager.

More information: https://flox.dev/docs/reference/command-reference/flox/.

- Create a new environment in the current directory:

```
flox init
```

- Enter an environment, or create one if it doesn't exist:

```
flox activate
```

- Search for packages in the FloxHub catalog:

```
flox search {{package}}
```

- Install a package into the current environment:

```
flox {{[i|install]}} {{package}}
```

- Uninstall a package from the current environment:

```
flox uninstall {{package}}
```

- View a list of all packages installed in the current environment:

```
flox {{[l|list]}}
```

- Push a local environment to FloxHub to share with others:

```
flox push
```

- Pull a shared environment from FloxHub:

```
flox pull {{environment_name}}
```

fls

List files and directories in an image file or device.

More information: https://wiki.sleuthkit.org/index.php?title=Fls.

- Build a recursive fls list over a device, output paths will start with C:

```
fls -r -m {{C:}} {{/dev/loop1p1}}
```

- Analyze a single partition, providing the sector offset at which the filesystem starts in the image:

```
fls -r -m {{C:}} -o {{sector}} {{path/to/image_file}}
```

- Analyze a single partition, providing the timezone of the original system:

```
fls -r -m {{C:}} -z {{timezone}} {{/dev/loop1p1}}
```

flutter pub

Flutter's package manager.

Note: Packages are available on https://pub.dev.

See also: **`flutter`**.

More information: https://docs.flutter.dev/packages-and-plugins/using-packages.

- Download/Update all packages specified in `pubspec.yaml`:

  ```
  flutter pub get
  ```

- Add a package dependency to an app:

  ```
  flutter pub add {{package1 package2 ...}}
  ```

- Remove a package dependency from an app:

  ```
  flutter pub remove {{package1 package2 ...}}
  ```

- Upgrade to the highest version of a package that is allowed by `pubspec.yaml`:

  ```
  flutter pub upgrade {{package}}
  ```

flutter

Google's free, open source, and cross-platform mobile app SDK.

Some subcommands such as **pub** have their own usage documentation.

More information: https://github.com/flutter/flutter/wiki/The-flutter-tool.

- Initialize a new Flutter project in a directory of the same name:

```
flutter create {{project_name}}
```

- Check if all external tools are correctly installed:

```
flutter doctor
```

- List or change Flutter channel:

```
flutter channel {{stable|beta|dev|master}}
```

- Run Flutter on all started emulators and connected devices:

```
flutter run -d all
```

- Run tests in a terminal from the root of the project:

```
flutter test {{test/example_test.dart}}
```

- Build a release APK targeting most modern smartphones:

```
flutter build apk --target-platform {{android-arm}},
{{android-arm64}}
```

- Delete the `build` and `.dart_tool` directories:

```
flutter clean
```

- Display help about a specific command:

```
flutter help {{command}}
```

flutterfire

Configure a Firebase project for your Flutter app.

Note: Requires Firebase CLI and FlutterFire CLI to be installed.

More information: https://firebase.google.com/docs/flutter/setup.

- Initialize and configure Firebase for your current Flutter app:

  ```
  flutterfire configure
  ```

- Specify platforms (iOS, Android, web) to support:

  ```
  flutterfire configure --platforms {{ios,android,...}}
  ```

- Use a specific Firebase project:

  ```
  flutterfire configure --project {{project_id}}
  ```

fluxctl

Tool for Flux v1.

More information: https://fluxcd.io/legacy/flux/references/fluxctl.

- List workloads currently running in the cluster on specific namespace:

  ```
  fluxctl --k8s-fwd-ns={{namespace}} list-workloads
  ```

- Show deployed and available images:

  ```
  fluxctl list-images
  ```

- Synchronize the cluster with the Git repository:

  ```
  fluxctl sync
  ```

- Turn on automatic deployment for a workload:

  ```
  fluxctl automate
  ```

fly

Tool for concourse-ci.

More information: https://concourse-ci.org/fly.html.

- Authenticate with and save concourse target:

```
fly {{[-t|--target]}} {{target_name}} login {{[-n|--team-name]}} {{team_name}} {{[-c|--concourse-url]}} {{https://ci.example.com}}
```

- List targets:

```
fly targets
```

- List pipelines:

```
fly {{[-t|--target]}} {{target_name}} pipelines
```

- Upload or update a pipeline:

```
fly {{[-t|--target]}} {{target_name}} set-pipeline {{[-c|--config]}} {{pipeline.yml}} {{[-p|--pipeline]}} {{pipeline_name}}
```

- Unpause pipeline:

```
fly {{[-t|--target]}} {{target_name}} unpause-pipeline {{[-p|--pipeline]}} {{pipeline_name}}
```

- Show pipeline configuration:

```
fly {{[-t|--target]}} {{target_name}} get-pipeline {{[-p|--pipeline]}} {{pipeline_name}}
```

- Update local copy of fly:

```
fly {{[-t|--target]}} {{target_name}} sync
```

- Destroy pipeline:

```
fly {{[-t|--target]}} {{target_name}} destroy-pipeline {{[-p|--pipeline]}} {{pipeline_name}}
```

flyctl

Tool for flyctl.io.

More information: https://github.com/superfly/flyctl.

- Sign into a Fly account:

```
flyctl auth login
```

- Launch an application from a specific Dockerfile (the default path is the current working directory):

```
flyctl launch --dockerfile {{path/to/dockerfile}}
```

- Open the current deployed application in the default web browser:

```
flyctl open
```

- Deploy the Fly applications from a specific Dockerfile:

```
flyctl deploy --dockerfile {{path/to/dockerfile}}
```

- Open the Fly Web UI for the current application in a web browser:

```
flyctl dashboard
```

- List all applications in the logged-in Fly account:

```
flyctl apps list
```

- View the status of a specific running application:

```
flyctl status --app {{app_name}}
```

- Display version information:

```
flyctl version
```

fmt

Reformat a text file by joining its paragraphs and limiting the line width to a number of characters (75 by default).

More information: https://www.gnu.org/software/coreutils/manual/html_node/fmt-invocation.html.

- Reformat a file:

```
fmt {{path/to/file}}
```

- Reformat a file producing output lines of (at most) n characters:

```
fmt {{[-w|--width]}} {{n}} {{path/to/file}}
```

- Reformat a file without joining lines shorter than the given width together:

```
fmt {{[-s|--split-only]}} {{path/to/file}}
```

- Reformat a file with uniform spacing (1 space between words and 2 spaces between paragraphs):

```
fmt {{[-u|--uniform-spacing]}} {{path/to/file}}
```

fnm

Fast Node.js version manager.

More information: https://github.com/Schniz/fnm/blob/master/docs/commands.md.

- Install a specific version of Node.js:

```
fnm {{[i|install]}} {{node_version}}
```

- Install Node.js LTS:

```
fnm {{[i|install]}} --lts
```

- List all available Node.js versions and highlight the default one:

```
fnm {{[ls|list]}}
```

- Use a specific version of Node.js in the current shell:

```
fnm use {{node_version}}
```

- Set the default Node.js version:

```
fnm default {{node_version}}
```

- Uninstall a given Node.js version:

```
fnm {{[uni|uninstall]}} {{node_version}}
```

fold

Wrap each line in an input file to fit a specified width and print it to **`stdout`**.

More information: https://manned.org/fold.1p.

- Wrap each line to default width (80 characters):

```
fold {{path/to/file}}
```

- Wrap each line to width "30":

```
fold -w30 {{path/to/file}}
```

- Wrap each line to width "5" and break the line at spaces (puts each space separated word in a new line, words with length > 5 are wrapped):

```
fold -w5 -s {{path/to/file}}
```

for

Perform a command several times.

More information: https://www.gnu.org/software/bash/manual/bash.html#Looping-Constructs.

- Iterate through command line arguments:

```
for {{variable}}; do {{echo $variable}}; done
```

- Execute the given commands for each of the specified items:

```
for {{variable}} in {{item1 item2 ...}}; do {{echo "Loop is executed"}}; done
```

- Iterate over a given range of numbers:

```
for {{variable}} in {{{from..to..step}}}; do {{echo "Loop is executed"}}; done
```

- Iterate over a given list of files:

```
for {{variable}} in {{path/to/file1 path/to/file2 ...}}; do {{echo "Loop is executed"}}; done
```

- Iterate over a given list of directories:

```
for {{variable}} in {{path/to/directory1/ path/to/directory2/ ...}}; do {{echo "Loop is executed"}}; done
```

- Perform a given command in every directory:

```
for {{variable}} in */; do (cd "${{variable}}" || continue; {{echo "Loop is executed"}}) done
```

forever

Server-side JavaScript application that makes sure Node.js applications run indefinitely (restarts after exit).

More information: https://github.com/foreversd/forever#usage.

- Start running a file forever (as a daemon):

```
forever {{script}}
```

- List running "forever" processes (along with IDs and other details of "forever" processes):

```
forever list
```

- Stop a running "forever" process:

```
forever stop {{ID|pid|script}}
```

fortune

Print a random quotation (fortune-cookie style).

More information: https://manned.org/fortune.

- Print a quotation:

  `fortune`

- Print an offensive quotation:

  `fortune -o`

- Print a long quotation:

  `fortune -l`

- Print a short quotation:

  `fortune -s`

- List the available quotation database files:

  `fortune -f`

- Print a quotation from one of the database files listed by `fortune -f`:

  `fortune {{path/to/file}}`

fossa

CLI for the Fossa service - Generate realtime license audits, vulnerability scans and reports about dependencies licenses.

More information: https://github.com/fossas/fossa-cli/blob/master/docs/README.md.

- Initialize a `.fossa.yml` configuration file:

```
fossa init
```

- Run a default project build:

```
fossa build
```

- Analyze built dependencies:

```
fossa analyze
```

- Generate reports:

```
fossa report
```

- Test current revision against the FOSSA scan status and exit with errors if issues are found:

```
fossa test
```

fossil add

Put files or directories under Fossil version control.

More information: https://fossil-scm.org/home/help/add.

- Put a file or directory under version control, so it will be in the current checkout:

```
fossil add {{path/to/file_or_directory}}
```

- Remove all added files from the current checkout:

```
fossil add --reset
```

fossil ci

This command is an alias of **`fossil commit`**.

- View documentation for the original command:

```
tldr fossil commit
```

fossil commit

Commit files to a Fossil repository.

More information: https://fossil-scm.org/home/help/commit.

- Create a new version containing all the changes in the current checkout; user will be prompted for a comment:

```
fossil {{[ci|commit]}}
```

- Create a new version containing all the changes in the current checkout, using the specified [m]essage:

```
fossil {{[ci|commit]}} {{[-m|--comment]}} "{{comment}}"
```

- Create a new version containing all the changes in the current checkout with a comment read from a specific file:

```
fossil {{[ci|commit]}} {{[-M|--message-file]}} {{path/to/commit_message_file}}
```

- Create a new version containing changes from the specified files; user will be prompted for a comment:

```
fossil {{[ci|commit]}} {{path/to/file1 path/to/file2 ...}}
```

fossil delete

Remove files or directories from Fossil version control.

See also: **`fossil forget`**.

More information: https://fossil-scm.org/home/help/delete.

- Remove a file or directory from Fossil version control:

```
fossil {{[rm|delete]}} {{path/to/file_or_directory}}
```

- Remove a file or directory from Fossil version control, and also delete it from the disk:

```
fossil {{[rm|delete]}} --hard {{path/to/file_or_directory}}
```

- Re-add all previously removed and uncommitted files to Fossil version control:

```
fossil {{[rm|delete]}} --reset
```

fossil forget

This command is an alias of **`fossil rm`**, with the exception that it never removes files from the disk.

More information: https://fossil-scm.org/home/help/forget.

- View documentation for the original command:

```
tldr fossil rm
```

fossil init

Initialize a new repository for a project.

See also: **`fossil clone`**.

More information: https://fossil-scm.org/home/help/init.

- Create a new repository in a named file:

```
fossil init {{path/to/file}}
```

fossil new

This command is an alias of **`fossil init`**.

- View documentation for the original command:

```
tldr fossil init
```

fossil rm

This command is an alias of **`fossil delete`**.

- View documentation for the original command:

```
tldr fossil delete
```

fossil

Distributed version control system.

Some subcommands such as **`commit`** have their own usage documentation.

More information: https://fossil-scm.org/home/help.

- Execute a Fossil subcommand:

```
fossil {{subcommand}}
```

- Display help:

```
fossil help
```

- Display help for a specific subcommand (like `add`, `commit`, etc.):

```
fossil help {{subcommand}}
```

- Display version:

```
fossil version
```

fping

A more powerful ping which can ping multiple hosts.

More information: https://fping.org.

- List the status of all hosts within a range:

```
fping {{192.168.1.{1..254}}}
```

- List alive hosts within a subnet generated from a netmask:

```
fping {{[-a|--alive]}} {{[-g|--generate]}} {{192.168.1.0/24}}
```

- List alive hosts within a subnet generated from an IP range and prune per-probe results:

```
fping {{[-q|--quiet]}} {{[-a|--alive]}} {{[-g|--generate]}}
{{192.168.1.1}} {{192.168.1.254}}
```

- List unreachable hosts within a subnet generated from a netmask:

```
fping {{[-u|--unreach]}} {{[-g|--generate]}}
{{192.168.1.0/24}}
```

fpsync

Execute several synchronization processes locally or on several remote workers through SSH.

More information: https://manned.org/fpsync.

- Recursively synchronize a directory to another location:

```
fpsync -v /{{path/to/source}}/ /{{path/to/destination}}/
```

- Recursively synchronize a directory with the final pass (It enables rsync's `--delete` option with each synchronization job):

```
fpsync -v -E /{{path/to/source}}/ /{{path/to/destination}}/
```

- Recursively synchronize a directory to a destination using 8 concurrent synchronization jobs:

```
fpsync -v -n 8 -E /{{path/to/source}}/ /{{path/to/destination}}/
```

- Recursively synchronize a directory to a destination using 8 concurrent synchronization jobs spread over two remote workers (machine1 and machine2):

```
fpsync -v -n 8 -E -w login@machine1 -w login@machine2 -d /{{path/to/shared_directory}} /{{path/to/source}}/ /{{path/to/destination}}/
```

- Recursively synchronize a directory to a destination using 4 local workers, each one transferring at most 1000 files and 100 MB per synchronization job:

```
fpsync -v -n 4 -f 1000 -s $((100 * 1024 * 1024)) /{{path/to/source}}/ /{{path/to/destination}}/
```

- Recursively synchronize any directories but exclude specific `.snapshot*` files (Note: Options and values must be separated by a pipe character):

```
fpsync -v -O "-x|.snapshot*" /{{path/to/source}}/ /{{path/to/destination}}/
```

freeramdisk

Free memory used by **`loadlin`** ramdisk on legacy systems.

Largely deprecated by **`umount`**, **`losetup`**, and **`tmpfs`**.

More information: https://manned.org/freeramdisk.

- Free `loadlin` ramdisk memory:

```
sudo freeramdisk
```

freeze

Generate images of code and terminal output.

Supported formats are PNG, SVG, and WebP.

See also: **silicon**.

More information: https://github.com/charmbracelet/freeze#flags.

- Generate an image of code based on a file:

```
freeze {{path/to/file}}
```

- Specify the output path:

```
freeze {{path/to/file}} {{[-o|--output]}} {{path/to/output_image.png}}
```

- Generate an image of terminal output:

```
freeze {{[-x|--execute]}} {{command}}
```

- Interactively customize the output image:

```
freeze {{path/to/file}} {{[-i|--interactive]}}
```

- Select a theme for syntax highlighting:

```
freeze {{path/to/file}} {{[-t|--theme]}} {{dracula}}
```

- Use a base configuration template:

```
freeze {{path/to/file}} {{[-c|--config]}} {{base|full|user}}
```

- Capture a specific range of line numbers:

```
freeze {{path/to/file}} --lines {{start}},{{end}}
```

- Show line numbers:

```
freeze {{path/to/file}} --show-line-numbers
```

freshclam

Update virus definitions for ClamAV antivirus program.

More information: https://www.clamav.net.

- Update virus definitions:

```
freshclam
```

from

Print mail header lines from the current user's mailbox.

More information: https://mailutils.org/manual/html_chapter/Programs.html#frm-and-from.

- List mail:

```
from
```

- Display the number of messages stored:

```
from --count
```

- List mail in the specified mailbox directory:

```
MAIL={{path/to/mailbox}} from
```

- Print the mail from the specified address:

```
from --sender={{me@example.com}}
```

frp

Fast Reverse Proxy: quickly set up network tunnels to expose certain services to the Internet or other external networks.

More information: https://github.com/fatedier/frp.

- View documentation for `frpc`, the `frp` client component:

```
tldr frpc
```

- View documentation for `frps`, the `frp` server component:

```
tldr frps
```

frpc

Connect to a **`frps`** server to start proxying connections on the current host.

Part of **`frp`**.

More information: https://github.com/fatedier/frp.

- Start the service, using the default configuration file (assumed to be `frps.ini` in the current directory):

```
frpc
```

- Start the service, using the newer TOML configuration file (`frps.toml` instead of `frps.ini`) in the current directory:

```
frpc {{[-c|--config]}} ./frps.toml
```

- Start the service, using a specific configuration file:

```
frpc {{[-c|--config]}} {{path/to/file}}
```

- Check if the configuration file is valid:

```
frpc verify {{[-c|--config]}} {{path/to/file}}
```

- Print autocompletion setup script for Bash, fish, PowerShell, or Zsh:

```
frpc completion {{bash|fish|powershell|zsh}}
```

- Display version:

```
frpc {{[-v|--version]}}
```

frps

Quickly set up a reverse proxy service.

Part of **`frp`**.

More information: https://github.com/fatedier/frp.

- Start the service, using the default configuration file (assumed to be `frps.ini` in the current directory):

```
frps
```

- Start the service, using the newer TOML configuration file (`frps.toml` instead of `frps.ini`) in the current directory:

```
frps {{[-c|--config]}} ./frps.toml
```

- Start the service, using a specified configuration file:

```
frps {{[-c|--config]}} {{path/to/file}}
```

- Check if the configuration file is valid:

```
frps verify {{[-c|--config]}} {{path/to/file}}
```

- Print autocompletion setup script for Bash, fish, PowerShell, or Zsh:

```
frps completion {{bash|fish|powershell|zsh}}
```

- Display version:

```
frps {{[-v|--version]}}
```

fselect

Find files with SQL-like queries.

More information: https://github.com/jhspetersson/fselect.

- Select full path and size from temporary or configuration files in a given directory:

```
fselect size, path from {{path/to/directory}} where name = '{{*.cfg}}' or name = '{{*.tmp}}'
```

- Find square images:

```
fselect path from {{path/to/directory}} where width = height
```

- Find old-school rap 320kbps MP3 files:

```
fselect path from {{path/to/directory}} where genre = {{Rap}} and bitrate = {{320}} and mp3_year lt {{2000}}
```

- Select only the first 5 results and output as JSON:

```
fselect size, path from {{path/to/directory}} limit {{5}} into json
```

- Use SQL aggregate functions to calculate minimum, maximum and average size of files in a directory:

```
fselect "{{MIN(size), MAX(size), AVG(size), SUM(size), COUNT(*)}} from {{path/to/directory}}"
```

fstopgm

Convert a Usenix FaceSaver file into a PGM image.

See also: **`pgmtofs`**.

More information: https://netpbm.sourceforge.net/doc/fstopgm.html.

- Convert the specified Usenix FaceSaver file into a PGM image:

```
fstopgm {{path/to/input.fs}} > {{path/to/output.pgm}}
```

fswatch

A cross-platform file change monitor.

More information: https://emcrisostomo.github.io/fswatch.

- Run a Bash command on file creation, update or deletion:

```
fswatch {{path/to/file}} | xargs {{[-n|--max-args]}} 1
{{bash_command}}
```

- Watch one or more files and/or directories:

```
fswatch {{path/to/file}} {{path/to/directory}} {{path/to/
another_directory/**/*.js}} | xargs {{[-n|--max-args]}} 1
{{bash_command}}
```

- Print the absolute paths of the changed files:

```
fswatch {{path/to/directory}} | xargs {{[-n|--max-args]}} 1 -
I _ echo _
```

- Filter by event type:

```
fswatch --event {{Updated|Removed|Created|...}} {{path/to/
directory}} | xargs {{[-n|--max-args]}} 1 {{bash_command}}
```

fswebcam

Small and simple webcam for *nix.

More information: https://www.sanslogic.co.uk/fswebcam.

- Take a picture:

```
fswebcam {{filename}}
```

- Take a picture with custom resolution:

```
fswebcam {{[-r|--resolution]}} {{width}}x{{height}}
{{filename}}
```

- Take a picture from selected device(Default is `/dev/video0`):

```
fswebcam {{[-d|--device]}} {{device}} {{filename}}
```

- Take a picture with timestamp(timestamp string is formatted by strftime):

```
fswebcam --timestamp {{timestamp}} {{filename}}
```

ftp

Tools to interact with a server via File Transfer Protocol.

More information: https://manned.org/ftp.

- Connect to an FTP server:

```
ftp {{ftp.example.com}}
```

- Connect to an FTP server specifying its IP address and port:

```
ftp {{ip_address}} {{port}}
```

- Switch to binary transfer mode (graphics, compressed files, etc):

```
binary
```

- Transfer multiple files without prompting for confirmation on every file:

```
prompt off
```

- Download multiple files (glob expression):

```
mget {{*.png}}
```

- Upload multiple files (glob expression):

```
mput {{*.zip}}
```

- Delete multiple files on the remote server:

```
mdelete {{*.txt}}
```

- Rename a file on the remote server:

```
rename {{original_filename}} {{new_filename}}
```

fuck

Corrects your previous console command.

More information: https://github.com/nvbn/thefuck.

- Set the `fuck` alias to `thefuck` tool:

```
eval "$(thefuck --alias)"
```

- Try to match a rule for the previous command:

```
fuck
```

- Confirm the first choice immediately (correct argument depends on level of annoyance):

```
fuck --{{yes|yeah|hard}}
```

func

Azure Functions Core Tools: develop and test Azure Functions locally.

Local functions can connect to live Azure services, and can deploy a function app to an Azure subscription.

More information: https://learn.microsoft.com/azure/azure-functions/functions-run-local.

- Create a new functions project:

  ```
  func init {{project}}
  ```

- Create a new function:

  ```
  func new
  ```

- Run functions locally:

  ```
  func start
  ```

- Publish your code to a function app in Azure:

  ```
  func azure functionapp publish {{function}}
  ```

- Download all settings from an existing function app:

  ```
  func azure functionapp fetch-app-settings {{function}}
  ```

- Get the connection string for a specific storage account:

  ```
  func azure storage fetch-connection-string
  {{storage_account}}
  ```

function

Define a function.

More information: https://www.gnu.org/software/bash/manual/bash.html#Shell-Functions.

- Define a function with the specified name:

```
function {{func_name}} { {{echo "Function contents here"}}; }
```

- Run a function named `func_name`:

```
func_name
```

- Define a function without the `function` keyword:

```
{{func_name}}() { {{echo "Function contents here"}}; }
```

- Display help:

```
help function
```

funzip

Print the content of the first (non-directory) member in an archive without extraction.

More information: https://manned.org/funzip.

- Print the content of the first member in a Zip archive:

```
funzip {{path/to/archive.zip}}
```

- Print the content in a gzip archive:

```
funzip {{path/to/archive.gz}}
```

- Decrypt a Zip or gzip archive and print the content:

```
funzip -password {{password}} {{path/to/archive}}
```

fusermount

Mount and unmount FUSE filesystems.

More information: https://manned.org/fusermount.

- Unmount a FUSE filesystem:

```
fusermount -u {{path/to/mount_point}}
```

- Unmount a FUSE filesystem as soon as it becomes unused:

```
fusermount -z {{path/to/mount_point}}
```

- Display version:

```
fusermount --version
```

fvm

Flutter version manager.

More information: https://fvm.app/documentation/guides/basic-commands.

- Install a version of the Flutter SDK. Use without `version` for project settings:

```
fvm install {{version}}
```

- Set a specific version of Flutter SDK in a project:

```
fvm use {{version}} {{options}}
```

- Set a global version of the Flutter SDK:

```
fvm global {{version}}
```

- Delete the FVM cache:

```
fvm destroy
```

- Remove a specific version of the Flutter SDK:

```
fvm remove {{version}}
```

- List all installed versions of the Flutter SDK:

```
fvm list
```

- List all releases of the Flutter SDK:

```
fvm releases
```

fx

View and process JSON.

More information: https://fx.wtf/getting-started.

- View a JSON file interactively:

```
fx {{path/to/file.json}}
```

- Pretty-print and colorize JSON from `stdin`:

```
cat {{path/to/file.json}} | fx
```

- Open JSON data from a URL:

```
curl {{url}} | fx
```

- Filter JSON using JavaScript-like expressions:

```
fx {{path/to/file.json}} {{.name}}
```

- Parse a TOML input file into JSON:

```
fx --toml {{path/to/file.toml}}
```

- Parse a YAML input file into JSON:

```
fx --yaml {{path/to/file.yaml}}
```

fzf

Fuzzy finder.

Similar to **sk**.

More information: https://github.com/junegunn/fzf#usage.

- Start `fzf` on all files in the specified directory:

```
find {{path/to/directory}} -type f | fzf
```

- Start `fzf` for running processes:

```
ps aux | fzf
```

- Select multiple files with `<Shift Tab>` and write to a file:

```
find {{path/to/directory}} -type f | fzf {{[-m|--multi]}} > {{path/to/file}}
```

- Start `fzf` with a specified query:

```
fzf {{[-q|--query]}} "{{query}}"
```

- Start `fzf` on entries that start with `core` and end with either `go`, `rb`, or `py`:

```
fzf {{[-q|--query]}} "^core go$ | rb$ | py$"
```

- Start `fzf` on entries that do not match `pyc` and contain `travis`:

```
fzf {{[-q|--query]}} '!pyc travis'
```

g++

Compile C++ source files.

Part of GCC (GNU Compiler Collection).

More information: https://gcc.gnu.org/onlinedocs/gcc/C_002b_002b-Dialect-Options.html.

- Compile a source code file into an executable binary:

```
g++ {{path/to/source1.cpp path/to/source2.cpp ...}} {{[-o|--output]}} {{path/to/output_executable}}
```

- Activate output of all errors and warnings:

```
g++ {{path/to/source.cpp}} -Wall {{[-o|--output]}} {{path/to/output_executable}}
```

- Show common warnings, debug symbols in output, and optimize without affecting debugging:

```
g++ {{path/to/source.cpp}} -Wall {{[-g|--debug]}} -Og {{[-o|--output]}} {{path/to/output_executable}}
```

- Choose a language standard to compile for (C++98/C++11/C++14/C++17):

```
g++ {{path/to/source.cpp}} -std={{c++98|c++11|c++14|c++17}} {{[-o|--output]}} {{path/to/output_executable}}
```

- Include libraries located at a different path than the source file:

```
g++ {{path/to/source.cpp}} {{[-o|--output]}} {{path/to/output_executable}} -I{{path/to/header}} -L{{path/to/library}} -l{{library_name}}
```

- Compile and link multiple source code files into an executable binary:

```
g++ {{[-c|--compile]}} {{path/to/source1.cpp path/to/source2.cpp ...}} && g++ {{[-o|--output]}} {{path/to/output_executable}} {{path/to/source1.o path/to/source2.o ...}}
```

- Optimize the compiled program for performance:

```
g++ {{path/to/source.cpp}} -O{{1|2|3|fast}} {{[-o|--output]}}
{{path/to/output_executable}}
```

- Display version:

```
g++ --version
```

gacutil

Global Assembly Cache (CAG) management utility.

More information: https://manned.org/gacutil.

- Install the specified assembly into GAC:

```
gacutil -i {{path/to/assembly.dll}}
```

- Uninstall the specified assembly from GAC:

```
gacutil -i {{assembly_display_name}}
```

- Print the content of GAC:

```
gacutil -l
```

gallery-dl

Download image galleries and collections from several image hosting sites.

More information: https://github.com/mikf/gallery-dl.

- Download images from the specified URL:

```
gallery-dl "{{url}}"
```

- Save images to a specific directory:

```
gallery-dl --destination {{path/to/directory}} "{{url}}"
```

- Retrieve pre-existing cookies from your web browser (useful for sites that require login):

```
gallery-dl --cookies-from-browser {{browser}} "{{url}}"
```

- Get the direct URL of an image from a site supporting authentication with username and password:

```
gallery-dl --get-urls --username {{username}} --password {{password}} "{{url}}"
```

- Filter manga chapters by chapter number and language:

```
gallery-dl --chapter-filter "{{10 <= chapter < 20}}" --option "lang={{language_code}}" "{{url}}"
```

gammastep

Adjust the screen's color temperature according to the time of day.

More information: https://manned.org/gammastep.

- Turn on Gammastep with a specific [t]emperature during the day (e.g. 5700k) and at night (e.g. 3600k):

```
gammastep -t {{5700}}:{{3600}}
```

- Turn on Gammastep with a manually specified custom [l]ocation:

```
gammastep -l {{latitude}}:{{longitude}}
```

- Turn on Gammastep with a specific screen [b]rightness during the day (e.g. 70%) and at night (e.g. 40%), with minimum brightness 10% and maximum brightness 100%:

```
gammastep -b {{0.7}}:{{0.4}}
```

- Turn on Gammastep with custom [g]amma levels (between 0 and 1):

```
gammastep -g {{red}}:{{green}}:{{blue}}
```

- Turn on Gammastep with a c[O]nstant unchanging color temperature:

```
gammastep -O {{temperature}}
```

- Reset temperature adjustments applied by Gammastep:

```
gammastep -x
```

ganache-cli

Command-line version of Ganache, your personal blockchain for Ethereum development.

More information: https://www.trufflesuite.com/ganache.

- Run Ganache:

```
ganache-cli
```

- Run Ganache with a specific number of accounts:

```
ganache-cli --accounts={{number_of_accounts}}
```

- Run Ganache and lock available accounts by default:

```
ganache-cli --secure
```

- Run Ganache server and unlock specific accounts:

```
ganache-cli --secure --unlock "{{account_private_key1}}" --
unlock "{{account_private_key2}}"
```

- Run Ganache with a specific account and balance:

```
ganache-cli --account="{{account_private_key}},
{{account_balance}}"
```

- Run Ganache with accounts with a default balance:

```
ganache-cli --defaultBalanceEther={{default_balance}}
```

- Run Ganache and log all requests to `stdout`:

```
ganache-cli --verbose
```

gatsby

Static site generator for React.

More information: https://www.gatsbyjs.com/docs/reference/gatsby-cli/.

- Create a new site:

  `gatsby new {{site_name}}`

- Create a new site with a Gatsby `starter`:

  `gatsby new {{site_name}} {{url_of_starter_github_repo}}`

- Start a live-reloading local development server:

  `gatsby develop`

- Perform a production build and generate static HTML:

  `gatsby build`

- Start a local server which serves the production build:

  `gatsby serve`

gau

Get All URLs: fetch known URLs from AlienVault's Open Threat Exchange, the Wayback Machine, and Common Crawl for any domains.

More information: https://github.com/lc/gau.

- Fetch all URLs of a domain from AlienVault's Open Threat Exchange, the Wayback Machine, Common Crawl, and URLScan:

```
gau {{example.com}}
```

- Fetch URLs of multiple domains:

```
gau {{domain1 domain2 ...}}
```

- Fetch all URLs of several domains from an input file, running multiple threads:

```
gau < {{path/to/domains.txt}} --threads {{4}}
```

- Write [o]utput results to a file:

```
gau {{example.com}} --o {{path/to/found_urls.txt}}
```

- Search for URLs from only one specific provider:

```
gau --providers {{wayback|commoncrawl|otx|urlscan}}
{{example.com}}
```

- Search for URLs from multiple providers:

```
gau --providers {{wayback,otx,...}} {{example.com}}
```

- Search for URLs within specific date range:

```
gau --from {{YYYYMM}} --to {{YYYYMM}} {{example.com}}
```

gawk

GNU version of awk, a versatile programming language for working on files.

See also: **awk**.

More information: https://www.gnu.org/software/gawk/manual/gawk.html.

- Print the fifth column (a.k.a. field) in a space-separated file:

```
gawk '{print $5}' {{path/to/file}}
```

- Print the second column of the lines containing "foo" in a space-separated file:

```
gawk '/{{foo}}/ {print $2}' {{path/to/file}}
```

- Print the last column of each line in a file, using a comma (instead of space) as a field separator:

```
gawk {{[-F|--field-separator]}} ',' '{print $NF}' {{path/to/file}}
```

- Sum the values in the first column of a file and print the total:

```
gawk '{s+=$1} END {print s}' {{path/to/file}}
```

- Print every third line starting from the first line:

```
gawk 'NR%3==1' {{path/to/file}}
```

- Print different values based on conditions:

```
gawk '{if ($1 == "foo") print "Exact match foo"; else if ($1 ~ "bar") print "Partial match bar"; else print "Baz"}' {{path/to/file}}
```

- Print all the lines which the 10th column value is between a min and a max:

```
gawk '($10 >= {{min_value}} && $10 <= {{max_value}})'
```

- Print table of users with UID >=1000 with header and formatted output, using colon as separator (`%-20s` mean: 20 left-align string characters, `%6s` means: 6 right-align string characters):

```
gawk 'BEGIN {FS=":";printf "%-20s %6s %25s\n", "Name", "UID", "Shell"} $4 >= 1000 {printf "%-20s %6d %25s\n", $1, $4, $7}' /etc/passwd
```

gc

Count nodes, edges, connected components, or clusters in Graphviz **`.dot`** files.

More information: https://graphviz.org/pdf/gc.1.pdf.

- Count nodes and edges in a file:

```
gc {{path/to/file.dot}}
```

- Count only [n]odes:

```
gc -n {{path/to/file.dot}}
```

- Count only [e]dges:

```
gc -e {{path/to/file.dot}}
```

- Count [c]onnected components:

```
gc -c {{path/to/file.dot}}
```

- Display help:

```
gc -?
```

gcal

Display calendar.

More information: https://www.gnu.org/software/gcal/manual/gcal.html#Invoking-Gcal.

- Display calendar for the current month:

```
gcal
```

- Display calendar for the month of February of the year 2010:

```
gcal 2 2010
```

- Provide calendar sheet with week numbers:

```
gcal --with-week-number
```

- Change starting day of week to 1st day of the week (Monday):

```
gcal --starting-day=1
```

- Display the previous, current and next month surrounding today:

```
gcal .
```

gcalcli

Interact with Google Calendar.

Requests Google API authorization upon first launch.

More information: https://github.com/insanum/gcalcli.

- List your events for all your calendars over the next 7 days:

```
gcalcli agenda
```

- Show events starting from or between specific dates (also takes relative dates e.g. "tomorrow"):

```
gcalcli agenda {{mm/dd}} [{{mm/dd}}]
```

- List events from a specific calendar:

```
gcalcli --calendar {{calendar_name}} agenda
```

- Display an ASCII calendar of events by week:

```
gcalcli calw
```

- Display an ASCII calendar of events for a month:

```
gcalcli calm
```

- Quick-add an event to your calendar:

```
gcalcli --calendar {{calendar_name}} quick "{{mm/dd}} {{HH:MM}} {{event_name}}"
```

- Add an event to calendar. Triggers interactive prompt:

```
gcalcli --calendar "{{calendar_name}}" add
```

gcc

Preprocess and compile C and C++ source files, then assemble and link them together.

Part of GCC (GNU Compiler Collection).

More information: https://gcc.gnu.org/onlinedocs/gcc/.

- Compile multiple source files into an executable:

```
gcc {{path/to/source1.c path/to/source2.c ...}} {{[-o|--output]}} {{path/to/output_executable}}
```

- Activate output of all errors and warnings:

```
gcc {{path/to/source.c}} -Wall {{[-o|--output]}} {{output_executable}}
```

- Show common warnings, debug symbols in output, and optimize without affecting debugging:

```
gcc {{path/to/source.c}} -Wall {{[-g|--debug]}} -Og {{[-o|--output]}} {{path/to/output_executable}}
```

- Include libraries from a different path:

```
gcc {{path/to/source.c}} {{[-o|--output]}} {{path/to/output_executable}} -I{{path/to/header}} -L{{path/to/library}} -l{{library_name}}
```

- Compile source code into Assembler instructions:

```
gcc {{[-S|--assemble]}} {{path/to/source.c}}
```

- Compile source code into an object file without linking:

```
gcc {{[-c|--compile]}} {{path/to/source.c}}
```

- Optimize the compiled program for performance:

```
gcc {{path/to/source.c}} -O{{1|2|3|fast}} {{[-o|--output]}} {{path/to/output_executable}}
```

- Display version:

```
gcc --version
```

gcloud app

Build scalable applications on a managed serverless platform.

See also: **`gcloud`**.

More information: https://cloud.google.com/sdk/gcloud/reference/app.

- Deploy an app's code and configuration to the App Engine server:

```
gcloud app deploy {{deployables}}
```

- List all versions of all services deployed to the App Engine server:

```
gcloud app versions list
```

- Open the current app in a web browser:

```
gcloud app browse
```

- Create an App Engine app within the current project:

```
gcloud app create
```

- Display the latest App Engine app logs:

```
gcloud app logs read
```

gcloud auth

Grant and revoke authorization to **`gcloud`** and manage credentials.

See also: **`gcloud`**.

More information: https://cloud.google.com/sdk/gcloud/reference/auth.

- Authorize Google Cloud access for the `gcloud` CLI with Google Cloud user credentials and set the current account as active:

```
gcloud auth login
```

- Authorize Google Cloud access similar to `gcloud auth login` but with service account credentials:

```
gcloud auth activate-service-account
```

- Manage Application Default Credentials (ADC) for Cloud Client Libraries:

```
gcloud auth application-default
```

- Display a list of Google Cloud accounts currently authenticated on your system:

```
gcloud auth list
```

- Display the current account's access token:

```
gcloud auth print-access-token
```

- Remove access credentials for an account:

```
gcloud auth revoke
```

gcloud components install

Install components of the Google Cloud CLI, along with their dependencies.

Installs components at the current version of the Google Cloud CLI without upgrading the existing installation.

More information: https://cloud.google.com/sdk/gcloud/reference/components/install.

- View available components for installation:

  ```
  gcloud components list
  ```

- Install one or more components (installs any dependencies as well):

  ```
  gcloud components install {{component_id1 component_id2 ...}}
  ```

- Check the current version of Google Cloud CLI:

  ```
  gcloud version
  ```

- Update Google Cloud CLI to the latest version:

  ```
  gcloud components update
  ```

gcloud components update

Update all your installed Google Cloud CLI components to the latest version.

See also: **gcloud**.

More information: https://cloud.google.com/sdk/gcloud/reference/components/update.

- Update all components to the latest version:

```
gcloud components update
```

- Update all components to a specific version:

```
gcloud components update --version={{1.2.3}}
```

- Update components without confirmation (useful for automation scripts):

```
gcloud components update --quiet
```

gcloud compute

Create, run, and manage VMs on Google Cloud infrastructure.

See also: **`gcloud`**.

More information: https://cloud.google.com/sdk/gcloud/reference/compute.

- List Compute Engine zones:

```
gcloud compute zones list
```

- Create a VM instance:

```
gcloud compute instances create {{instance_name}}
```

- Display a VM instance's details:

```
gcloud compute instances describe {{instance_name}}
```

- List all VM instances in a project:

```
gcloud compute instances list
```

- Create a snapshot of a persistent disk:

```
gcloud compute disks snapshot {{disk_name}} --snapshot-names
{{snapshot_name}}
```

- Display a snapshot's details:

```
gcloud compute snapshots describe {{snapshot_name}}
```

- Delete a snapshot:

```
gcloud compute snapshots delete {{snapshot_name}}
```

- Connect to a VM instance using SSH:

```
gcloud compute ssh {{instance_name}}
```

gcloud config set

Set a property in the Google Cloud CLI configuration.

Properties control various aspects of Google Cloud CLI behavior.

More information: https://cloud.google.com/sdk/gcloud/reference/config/set.

- Set the project property in the core section:

```
gcloud config set project {{project_id}}
```

- Set the compute zone for future operations:

```
gcloud config set compute/zone {{zone_name}}
```

- Disable prompting to make gcloud suitable for scripting:

```
gcloud config set disable_prompts true
```

- View the list of properties currently in use:

```
gcloud config list
```

- Unset a previously set property:

```
gcloud config unset {{property_name}}
```

- Create a new configuration profile:

```
gcloud config configurations create {{configuration_name}}
```

- Switch between different configuration profiles:

```
gcloud config configurations activate {{configuration_name}}
```

gcloud config

Manage different configurations of **`gcloud`**.

See also: **`gcloud`**.

More information: https://cloud.google.com/sdk/gcloud/reference/config.

- Define a property (like compute/zone) for the current configuration:

```
gcloud config set {{property}} {{value}}
```

- Fetch the value of a `gcloud` property:

```
gcloud config get {{property}}
```

- Display all the properties for the current configuration:

```
gcloud config list
```

- Create a new configuration with a given name:

```
gcloud config configurations create {{configuration_name}}
```

- Display a list of all available configurations:

```
gcloud config configurations list
```

- Switch to an existing configuration with a given name:

```
gcloud config configurations activate {{configuration_name}}
```

gcloud container

Manage containerized applications on Kubernetes and clusters.

See also: **`gcloud`**.

More information: https://cloud.google.com/sdk/gcloud/reference/container.

- Register `gcloud` as a Docker credential helper:

```
gcloud auth configure-docker
```

- Create a cluster to run GKE containers:

```
gcloud container clusters create {{cluster_name}}
```

- List clusters for running GKE containers:

```
gcloud container clusters list
```

- Update kubeconfig to get `kubectl` to use a GKE cluster:

```
gcloud container clusters get-credentials {{cluster_name}}
```

- List tag and digest metadata for a container image:

```
gcloud container images list-tags {{image}}
```

- Describe an existing cluster for running containers:

```
gcloud container clusters describe {{cluster_name}}
```

gcloud feedback

Provide feedback to the Google Cloud team.

See also: **`gcloud`**.

More information: https://cloud.google.com/sdk/gcloud/reference/feedback.

- Provide feedback to the `gcloud` team:

```
gcloud feedback
```

- Provide feedback to the `gcloud` team and attach a log file:

```
gcloud feedback --log-file {{log_file}}
```

gcloud help

Display help and reference information for **`gcloud`**.

See also: **`gcloud topic`** for supplementary help topics not directly associated with individual commands.

More information: https://cloud.google.com/sdk/gcloud/reference/help.

- Search the `gcloud` CLI reference documents for specific terms:

```
gcloud help
```

gcloud iam

Configure Identity and Access Management (IAM) preferences and service accounts.

See also: **gcloud**.

More information: https://cloud.google.com/sdk/gcloud/reference/iam.

- List IAM grantable roles for a resource:

```
gcloud iam list-grantable-roles {{resource}}
```

- Create a custom role for a organization or project:

```
gcloud iam roles create {{role_name}} --{{organization|
project}} {{organization|project_id}} --file {{path/to/
role.yaml}}
```

- Create a service account for a project:

```
gcloud iam service-accounts create {{name}}
```

- Add an IAM policy binding to a service account:

```
gcloud iam service-accounts add-iam-policy-binding
{{service_account_email}} --member {{member}} --role {{role}}
```

- Replace existing IAM policy binding:

```
gcloud iam service-accounts set-iam-policy
{{service_account_email}} {{policy_file}}
```

- List a service account's keys:

```
gcloud iam service-accounts keys list --iam-account
{{service_account_email}}
```

gcloud info

Display information about the current **`gcloud`** environment.

More information: https://cloud.google.com/sdk/gcloud/reference/info.

- Display `gcloud` environment information:

```
gcloud info
```

- Check network connectivity and hidden properties:

```
gcloud info --run-diagnostics
```

- Print the contents of the most recent log file:

```
gcloud info --show-log
```

gcloud init

Launch an interactive workflow to setup **`gcloud`** or reinitialize configurations.

More information: https://cloud.google.com/sdk/gcloud/reference/init.

- Launch a "Getting Started" workflow:

```
gcloud init
```

- Launch a workflow without diagnostics:

```
gcloud init --skip-diagnostics
```

- Use the console for authentication:

```
gcloud init --console-only
```

gcloud kms decrypt

Decrypt a ciphertext file using a Cloud KMS key.

See also: **`gcloud`**.

More information: https://cloud.google.com/sdk/gcloud/reference/kms/decrypt.

- Decrypt a file using a specified key, key ring, and location:

```
gcloud kms decrypt --key={{key_name}} --
keyring={{keyring_name}} --location={{global}} --ciphertext-
file={{path/to/ciphertext}} --plaintext-file={{path/to/
plaintext}}
```

- Decrypt a file with additional authenticated data (AAD) and write the decrypted plaintext to `stdout`:

```
gcloud kms decrypt --key={{key_name}} --
keyring={{keyring_name}} --location={{global}} --additional-
authenticated-data-file={{path/to/file.aad}} --ciphertext-
file={{path/to/ciphertext}} --plaintext-file=-
```

gcloud logging logs list

List logs in a Google Cloud project.

Useful for identifying available logs for monitoring and analysis.

See also: **`gcloud`**.

More information: https://cloud.google.com/sdk/gcloud/reference/logging/logs/list.

- List all logs in the current project:

```
gcloud logging logs list
```

- List all logs for a specific log bucket and location:

```
gcloud logging logs list --bucket={{bucket_id}} --location={{location}}
```

- List all logs for a specific view in a log bucket:

```
gcloud logging logs list --bucket={{bucket_id}} --location={{location}} --view={{view_id}}
```

- List logs with a filter expression:

```
gcloud logging logs list --filter="{{expression}}"
```

- List a specified number of logs:

```
gcloud logging logs list --limit={{number}}
```

- List logs sorted by a specific field in ascending or descending order (~ for descending):

```
gcloud logging logs list --sort-by="{{field_name}}"
```

- List logs sorted by multiple fields:

```
gcloud logging logs list --sort-by="{{field1}},~{{field2}}"
```

- List logs with verbose output, showing additional details:

```
gcloud logging logs list --verbosity=debug
```

gcloud projects

Manage project access policies in Google Cloud.

See also: **`gcloud`**.

More information: https://cloud.google.com/sdk/gcloud/reference/projects.

- Create a new project:

```
gcloud projects create {{project_id|project_number}}
```

- List all active projects:

```
gcloud projects list
```

- Display metadata for a project:

```
gcloud projects describe {{project_id}}
```

- Delete a project:

```
gcloud projects delete {{project_id|project_number}}
```

- Add an IAM policy binding to a specified project:

```
gcloud projects add-iam-policy-binding {{project_id}} --
member {{principal}} --role {{role}}
```

gcloud sql backups describe

Retrieve information about a backup.

See also: **`gcloud`**.

More information: https://cloud.google.com/sdk/gcloud/reference/sql/backups/describe.

- Retrieve information about a backup:

```
gcloud sql backups describe {{backup_id}} --
instance={{instance_id}}
```

gcloud sql export sql

Export data from a Cloud SQL instance to an SQL file in Google Cloud Storage.

Useful for creating backups or migrating data.

See also: **`gcloud`**.

More information: https://cloud.google.com/sdk/gcloud/reference/sql/export/sql.

- Export data from a specific Cloud SQL instance to a Google Cloud Storage bucket as an SQL dump file:

```
gcloud sql export sql {{instance}} gs://{{bucket_name}}/
{{file_name}}
```

- Export data asynchronously, returning immediately without waiting for the operation to complete:

```
gcloud sql export sql {{instance}} gs://{{bucket_name}}/
{{file_name}} --async
```

- Export data from specific databases within the Cloud SQL instance:

```
gcloud sql export sql {{instance}} gs://{{bucket_name}}/
{{file_name}} --database={{database1,database2,...}}
```

- Export specific tables from a specified database within the Cloud SQL instance:

```
gcloud sql export sql {{instance}} gs://{{bucket_name}}/
{{file_name}} --database={{database}} --
table={{table1,table2,...}}
```

- Export data while offloading the operation to a temporary instance to reduce strain on the source instance:

```
gcloud sql export sql {{instance}} gs://{{bucket_name}}/
{{file_name}} --offload
```

- Export data and compress the output with `gzip`:

```
gcloud sql export sql {{instance}} gs://{{bucket_name}}/
{{file_name}}.gz
```

gcloud topic

Display supplementary help for topics not directly associated with individual commands.

See also: **`gcloud help`** for general help.

More information: https://cloud.google.com/sdk/gcloud/reference/topic.

- View supplementary help material for non-command topics like accessibility, filtering, and formatting:

  ```
  gcloud topic {{topic_name}}
  ```

- List all available topics:

  ```
  gcloud topic --help
  ```

gcloud-version

Print version information for Google Cloud CLI components.

More information: https://cloud.google.com/sdk/gcloud/reference/version.

- Display version information for all installed components, along with available updates to them:

```
gcloud version
```

- Display help:

```
gcloud version --help
```

gcloud

The official CLI tool for Google Cloud Platform.

Note: **`gcloud`** subcommands have their own usage documentation.

More information: https://cloud.google.com/sdk/gcloud.

- List all properties in one's active configuration:

  ```
  gcloud config list
  ```

- Login to a Google account:

  ```
  gcloud auth login
  ```

- Set the active project:

  ```
  gcloud config set project {{project_name}}
  ```

- SSH into a virtual machine instance:

  ```
  gcloud compute ssh {{user}}@{{instance}}
  ```

- Display all Google Compute Engine instances in a project (by default instances from all zones are listed):

  ```
  gcloud compute instances list
  ```

- Update a kubeconfig file with the appropriate credentials to point `kubectl` to a specific cluster in Google Kubernetes Engine (GKE):

  ```
  gcloud container clusters get-credentials {{cluster_name}}
  ```

- Update all `gcloud` components:

  ```
  gcloud components update
  ```

- Display help for a given command:

  ```
  gcloud help {{command}}
  ```

gcpdiag

Google Cloud Platform troubleshooting and diagnostics tool.

Run in a Docker container or in GCP Cloudshell.

More information: https://github.com/GoogleCloudPlatform/gcpdiag.

- Run `gcpdiag` on your project, returning all rules:

```
gcpdiag lint --project={{gcp_project_id}}
```

- Hide rules that are ok:

```
gcpdiag lint --project={{gcp_project_id}} --hide-ok
```

- Authenticate using a service account private key file:

```
gcpdiag lint --project={{gcp_project_id}} --auth-key {{path/to/private_key}}
```

- Search logs and metrics from a number of days back (default: 3 days):

```
gcpdiag lint --project={{gcp_project_id}} --within-days {{number}}
```

- Display help:

```
gcpdiag lint --help
```

gcrane copy

Efficiently copy a remote image from source to target while retaining the digest value.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/gcrane/README.md.

- Copy an image from source to target:

```
gcrane {{[cp|copy]}} {{source}} {{target}}
```

- Set the maximum number of concurrent copies, defaults to 20:

```
gcrane copy {{source}} {{target}} {{[-j|--jobs]}}
{{nr_of_copies}}
```

- Whether to recurse through repositories:

```
gcrane copy {{source}} {{target}} {{[-r|--recursive]}}
```

- Display help:

```
gcrane copy {{[-h|--help]}}
```

gcrane gc

List images that are not tagged.

Will calculate images that can be garbage-collected.

This can be composed with **`gcrane delete`** to actually garbage collect them.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/gcrane/README.md.

- List untagged images:

```
gcrane gc {{repository}}
```

- Whether to recurse through repositories:

```
gcrane gc {{repository}} {{[-r|--recursive]}}
```

- Display help:

```
gcrane gc {{[-h|--help]}}
```

gcrane help

Help provides help for any command in the application.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/gcrane/README.md.

- Display help for a subcommand:

```
gcrane help {{command}}
```

- Display help:

```
gcrane help {{[-h|--help]}}
```

gcrane ls

List the tags in a repository.

More complex form than **`crane ls`**, which allows for listing tags, manifest and sub-repositories.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/gcrane/README.md.

- List the tags:

  ```
  gcrane ls {{repository}}
  ```

- Format response from the registry as JSON:

  ```
  gcrane ls {{repository}} --json
  ```

- Whether to recurse through repositories:

  ```
  gcrane ls {{repository}} {{[-r|--recursive]}}
  ```

- Display help:

  ```
  gcrane ls {{[-h|--help]}}
  ```

gcrane

Container images managing tool.

This tool implements a superset of the **`crane`** commands, with additional commands that are specific to Google Container Registry (**`gcr.io`**).

Some subcommands such as **`append`**, **`auth`**, **`copy`**, etc. have their own usage documentation which can be found under **`crane`**.

Some subcommands such as **`completion`**, **`gc`**, **`help`** are specific to gcrane and have their own usage documentation.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/gcrane/README.md.

- Login to a registry:

```
gcrane auth login {{registry}} {{[-u|--username]}} {{user}}
{{[-p|--password]}} {{password}}
```

- List tags, manifests, and sub-repostiories:

```
gcrane ls {{registry}}/{{project_id}}
```

- Copy images from one registry to another:

```
gcrane cp {{[-r|--recursive]}} {{source_registry}}/
{{project_id}}/{{repository}} {{target_registry}}/
{{project_id}}/{{repository}}
```

- Print images that can be garbage collected:

```
gcrane gc {{registry}}/{{project_id}}/{{repository}}
```

- Delete images that can be garbage collected:

```
gcrane gc {{registry}}/{{project_id}}/{{repository}} | xargs
{{[-n|--max-args]}} 1 gcrane delete
```

- List a specific registry with specific ID:

```
gcrane ls {{gcr.io}}/{{my-project-id}}
```

- Migrate all images from US registry to EU registry:

```
gcrane cp {{[-r|--recursive]}} {{gcr.io}}/{{my-project-id}}/
{{repository}} {{eu.gcr.io}}/{{my-project-id}}/{{repository}}
```

gdal2tiles.py

Generate TMS or XYZ tiles for a raster dataset.

More information: https://gdal.org/programs/gdal2tiles.html.

- Generate TMS tiles for the zoom levels 2 to 5 of a raster dataset:

```
gdal2tiles.py --zoom 2-5 {{path/to/input.tif}} {{path/to/output_directory}}
```

- Generate XYZ tiles for the zoom levels 2 to 5 of a raster dataset:

```
gdal2tiles.py --zoom 2-5 --xyz {{path/to/input.tif}} {{path/to/output_directory}}
```

gdal_contour

Create contour lines and polygons from a digital elevation model.

More information: https://gdal.org/programs/gdal_contour.html.

- Create a vector dataset with contour lines spread over an 100-meter [i]nterval while [a]ttributing the elevation property as "ele":

```
gdal_contour -a {{ele}} -i {{100.0}} {{path/to/input.tif}} {{path/to/output.gpkg}}
```

- Create a vector dataset with [p]olygons spread over an 100-meter [i]nterval:

```
gdal_contour -i {{100.0}} -p {{path/to/input.tif}} {{path/to/output.gpkg}}
```

gdal_translate

Convert raster data between different formats.

More information: https://gdal.org/programs/gdal_translate.html.

- Convert a raster dataset to JPEG format:

```
gdal_translate -of {{JPEG}} {{path/to/input.tif}} {{path/to/output.jpeg}}
```

- Assign a projection to a raster dataset:

```
gdal_translate -a_srs {{EPSG:4326}} {{path/to/input.tif}} {{path/to/output.tif}}
```

- Reduce the size of a raster dataset to a specific fraction:

```
gdal_translate -outsize {{40%}} {{40%}} {{path/to/input.tif}} {{path/to/output.tif}}
```

- Convert a GeoTiff to a Cloud Optimized GeoTiff:

```
gdal_translate {{path/to/input.tif}} {{path/to/output.tif}} -of COG -co COMPRESS=LZW
```

gdaladdo

Build overview images of raster datasets.

More information: https://gdal.org/programs/gdaladdo.html.

- Build overview images of a raster dataset using the "average" [r]esampling method:

```
gdaladdo -r average {{path/to/input.tif}}
```

gdalbuildvrt

Build Virtual Datasets from a list of existing datasets.

More information: https://gdal.org/programs/gdalbuildvrt.html.

- Make a virtual mosaic from all TIFF files contained in a directory:

```
gdalbuildvrt {{path/to/output.vrt}} {{path/to/
input_directory/*.tif}}
```

- Make a virtual mosaic from files whose name is specified in a text file:

```
gdalbuildvrt -input_file_list {{path/to/list.txt}} {{path/to/
output.vrt}}
```

- Make an RGB virtual mosaic from 3 single-band input files:

```
gdalbuildvrt -separate {{path/to/rgb.vrt}} {{path/to/
red.tif}} {{path/to/green.tif}} {{path/to/blue.tif}}
```

- Make a virtual mosaic with blue background color (RGB: 0 0 255):

```
gdalbuildvrt -hidenodata -vrtnodata "{{0 0 255}}" {{path/to/
output.vrt}} {{path/to/input_directory/*.tif}}
```

gdaldem

Analyze and visualize digital elevation models (DEM).

More information: https://gdal.org/programs/gdaldem.html.

- Compute the hillshade of a DEM:

```
gdaldem hillshade {{path/to/input.tif}} {{path/to/
output.tif}}
```

- Compute the slope of a DEM:

```
gdaldem slope {{path/to/input.tif}} {{path/to/output.tif}}
```

- Compute the aspect of a DEM:

```
gdaldem aspect {{path/to/input.tif}} {{path/to/output.tif}}
```

gdalinfo

List various information about a GDAL supported raster dataset.

More information: https://gdal.org/programs/gdalinfo.html.

- List all supported raster formats:

```
gdalinfo --formats
```

- List information about a specific raster dataset:

```
gdalinfo {{path/to/input.tif}}
```

- List information about a specific raster dataset in JSON format:

```
gdalinfo -json {{path/to/input.tif}}
```

- Show histogram values of a specific raster dataset:

```
gdalinfo -hist {{path/to/input.tif}}
```

- List information about a Web Map Service (WMS):

```
gdalinfo WMS:{{https://services.meggsimum.de/geoserver/ows}}
```

- List information about a specific dataset of a Web Map Service (WMS):

```
gdalinfo WMS:{{https://services.meggsimum.de/geoserver/ows}}
-sd {{4}}
```

gdalwarp

Image reprojection and warping utility.

More information: https://gdal.org/programs/gdalwarp.html.

- Reproject a raster dataset:

```
gdalwarp -t_srs {{EPSG:4326}} {{path/to/input.tif}} {{path/to/output.tif}}
```

- Crop a raster dataset by using specific coordinates:

```
gdalwarp -te {{min_x}} {{min_y}} {{max_x}} {{max_y}} -te_srs {{EPSG:4326}} {{path/to/input.tif}} {{path/to/output.tif}}
```

- Crop a raster dataset using a vector layer:

```
gdalwarp -cutline {{path/to/area_to_cut.geojson}} -crop_to_cutline {{path/to/input.tif}} {{path/to/output.tif}}
```

gdb

The GNU Debugger.

More information: https://sourceware.org/gdb/current/onlinedocs/gdb#Invocation.

- Debug an executable:

```
gdb {{path/to/executable}}
```

- Attach a process to `gdb`:

```
gdb {{[-p|--pid]}} {{procID}}
```

- Debug with a core file:

```
gdb {{[-c|--core]}} {{path/to/core}} {{path/to/executable}}
```

- Execute given GDB commands upon start:

```
gdb {{[-ex|--eval-command]}} "{{commands}}" {{path/to/
executable}}
```

- Start `gdb` and pass arguments to the executable:

```
gdb --args {{path/to/executable}} {{argument1 argument2 ...}}
```

- Skip `debuginfod` and pagination prompts and then print the backtrace:

```
gdb {{[-c|--core]}} {{path/to/core}} {{path/to/executable}} -
iex 'set debuginfod enabled on' -iex 'set pagination off' -ex
bt
```

gdc

D compiler using GCC as a backend.

More information: https://wiki.dlang.org/Using_GDC.

- Create an executable:

```
gdc {{path/to/source.d}} -o {{path/to/output_executable}}
```

- Print information about module dependencies:

```
gdc -fdeps
```

- Generate Ddoc documentation:

```
gdc -fdoc
```

- Generate D interface files:

```
gdc -fintfc
```

- Do not link the standard GCC libraries in the compilation:

```
gdc -nostdlib
```

gdm-binary

This command is an alias of **gdm**.

- View documentation for the original command:

```
tldr gdm
```

gdm-restart

Restart the GNOME Display Manager (GDM) daemon.

See also: **gdm**, **gdm-binary**, **gdmsetup**, **gdm-stop**, **gdm-safe-restart**.

More information: https://manned.org/gdm.

- Restart the GNOME Display Manager application:

```
gdm-restart
```

gdm-safe-restart

Restart the GNOME Display Manager (GDM) only when all users have logged out.

See also: **gdm**, **gdm-binary**, **gdmsetup**, **gdm-stop**, **gdm-restart**.

More information: https://manned.org/gdm.

- Restart the GNOME Display Manager application when all users have logged out:

```
gdm-safe-restart
```

gdm-stop

Immediately stop running the GNOME Display Manager (GDM).

See also: **`gdm`**, **`gdm-binary`**, **`gdmsetup`**, **`gdm-restart`**, **`gdm-safe-restart`**.

More information: https://manned.org/gdm.

- Stop the GNOME Display Manager application:

```
gdm-stop
```

gdm

The GNOME Display Manager (GDM) is a replacement for the X Display Manager (XDM).

See also: **`gdm-binary`**, **`gdmsetup`**, **`gdm-stop`**, **`gdm-restart`**, **`gdm-safe-restart`**.

More information: https://manned.org/gdm.

- Run the GNOME Display Manager application:

  `gdm`

- Prevent `gdm` from being run as a daemon background process:

  `gdm --nodaemon`

- Disable `gdm` management of local console X servers for headless or remote environments:

  `gdm --no-console`

- Prevent sanitizing environment variables that start with `LD_`:

  `gdm --preserve-ld-vars`

- Display help:

  `gdm --help`

- Display version:

  `gdm --version`

gdmsetup

Launch a GUI for configuring the GNOME Display Manager (GDM).

Results in edited files in either **`/etc/X11/gdm/gdm.conf`** or **`/etc/gdm/gdm.conf`** depending on distribution and environment.

See also: **`gdm`**, **`gdm-binary`**, **`gdm-stop`**, **`gdm-restart`**, **`gdm-safe-restart`**.

More information: https://manned.org/gdm.

- Run the GNOME Display Manager GUI configuration application:

  ```
  gdmsetup
  ```

gdown

Download files from Google Drive and other URLs.

More information: https://github.com/wkentaro/gdown.

- Download a file from a URL:

```
gdown {{url}}
```

- Download using a file ID:

```
gdown {{file_id}}
```

- Download with fuzzy file ID extraction (also works with https://docs.google.com links):

```
gdown --fuzzy {{url}}
```

- Download a folder using its ID or the full URL:

```
gdown {{folder_id|url}} {{[-O|--output]}} {{path/to/output_directory}} --folder
```

- Download a tar archive, write it to `stdout` and extract it:

```
gdown {{tar_url}} {{[-O|--output]}} - {{[-q|--quiet]}} | tar xvf -
```

gdrive

Interact with Google Drive.

Folder/file ID can be obtained from the Google Drive folder or ID URL.

More information: https://github.com/gdrive-org/gdrive.

- Upload a local path to the parent folder with the specified ID:

```
gdrive upload {{[-p|--parent]}} {{id}} {{path/to/file_or_folder}}
```

- Download file or directory by ID to current directory:

```
gdrive download {{id}}
```

- Download to a given local path by its ID:

```
gdrive download --path {{path/to/folder}} {{id}}
```

- Create a new revision of an ID using a given file or folder:

```
gdrive update {{id}} {{path/to/file_or_folder}}
```

gdu

Disk usage analyzer with console interface.

More information: https://github.com/dundee/gdu.

- Interactively show the disk usage of the current directory:

```
gdu
```

- Interactively show the disk usage of a given directory:

```
gdu {{path/to/directory}}
```

- Interactively show the disk usage of all mounted disks:

```
gdu --show-disks
```

- Interactively show the disk usage of the current directory but ignore some sub-directories:

```
gdu --ignore-dirs {{path/to/directory1,path/to/
directory2,...}}
```

- Ignore paths by regex:

```
gdu --ignore-dirs-pattern '{{.*[abc]+}}'
```

- Ignore hidden directories:

```
gdu --no-hidden
```

- Only print the result, do not enter interactive mode:

```
gdu --non-interactive {{path/to/directory}}
```

- Do not show the progress in non-interactive mode (useful in scripts):

```
gdu --no-progress {{path/to/directory}}
```

gem

A package manager for the Ruby programming language.

More information: https://guides.rubygems.org/command-reference/.

- Search for remote gem(s) and show all available versions:

```
gem search {{regex}} {{[-a|--all]}}
```

- Install the latest version of a gem:

```
gem install {{gem_name}}
```

- Install a specific version of a gem:

```
gem install {{gem_name}} {{[-v|--version]}} {{1.0.0}}
```

- Install the latest matching (SemVer) version of a gem:

```
gem install {{gem_name}} {{[-v|--version]}} '~> {{1.0}}'
```

- Update a gem:

```
gem update {{gem_name}}
```

- List all local gems:

```
gem list
```

- Uninstall a gem:

```
gem uninstall {{gem_name}}
```

- Uninstall a specific version of a gem:

```
gem uninstall {{gem_name}} {{[-v|--version]}} {{1.0.0}}
```

gemini

Launch an interactive prompt with Gemini AI.

More information: https://github.com/google-gemini/gemini-cli.

- Start a REPL session to chat interactively:

```
gemini
```

- Send the output of another command to Gemini and exit immediately:

```
{{echo "Summarize the history of Rome"}} | gemini {{[-p|--prompt]}}
```

- Override the default model (default: gemini-2.5-pro):

```
gemini {{[-m|--model]}} {{gemini-2.5-flash}}
```

- Run inside a sandbox container:

```
gemini {{[-s|--sandbox]}}
```

- Execute a prompt then stay in interactive mode:

```
gemini {{[-i|--prompt-interactive]}} "{{Give me an example of recursion in Python}}"
```

- Include all files in context:

```
gemini {{[-a|--all-files]}}
```

- Show memory usage in status bar:

```
gemini --show-memory-usage
```

gemtopbm

This command has been superseded by **`gemtopnm`**.

More information: https://netpbm.sourceforge.net/doc/gemtopbm.html.

- View documentation for the current command:

```
tldr gemtopnm
```

gemtopnm

Convert a GEM image file into a PNM image.

More information: https://netpbm.sourceforge.net/doc/gemtopnm.html.

- Convert a GEM image file to a PNM image:

```
gemtopnm {{path/to/file.img}} > {{path/to/output.pnm}}
```

- Describe the contents of the specified GEM image:

```
gemtopnm {{[-d|-debug]}} {{path/to/file.img}}
```

- Display version:

```
gemtopnm -version
```

gendesk

Specifies the command to generate a **`.desktop`** file and a download icon with minimal information.

More information: https://manned.org/gendesk.

- Create a `.desktop` file named `app`:

```
gendesk -n --name "{{app}}" --exec "/{{path/to/app}}" --icon "/{{path/to/icon.png}}" --comment "{{This is application}}"
```

- Create a `.desktop` file named `app`, do not display any output, and overwrite it if it exists:

```
gendesk -q -f -n --name "{{app}}" --exec "/{{path/to/app}}" --icon "/{{path/to/icon.png}}" --comment "{{This is application}}"
```

- Display help:

```
gendesk {{[-h|--help]}}
```

Get-NodeInstallLocation

Get the current Node.js installation directory for **`ps-nvm`**.

Part of **`ps-nvm`** and can only be run under PowerShell.

More information: https://github.com/aaronpowell/ps-nvm.

- Get the current Node.js installation directory:

```
Get-NodeInstallLocation
```

Get-NodeVersions

List installed and available Node.js versions for **`ps-nvm`**.

Part of **`ps-nvm`** and can only be run under PowerShell.

More information: https://github.com/aaronpowell/ps-nvm.

- List all installed Node.js versions:

  ```
  Get-NodeVersions
  ```

- List all available Node.js versions:

  ```
  Get-NodeVersions -Remote
  ```

- List all available Node.js 20.x versions:

  ```
  Get-NodeVersions -Remote -Filter ">=20.0.0 <21.0.0"
  ```

get_iplayer

Indexing tool and personal video recorder for BBC iPlayer and BBC Sounds.

More information: https://github.com/get-iplayer/get_iplayer/wiki/manpage.

- Search programmes by name:

```
get_iplayer "{{program_name}}"
```

- Record programme by results of search:

```
get_iplayer "{{program_name}}" {{[-g|--get]}}
```

- Record programme by URL from the BBC iPlayer website:

```
get_iplayer "https://www.bbc.co.uk/iplayer/episode/
{{program_PID}}/{{name-of-show-episode-number-episode-
title}}"
```

- Download subtitles for a programme by results of search:

```
get_iplayer "{{program_name}}" --subtitles-only
```

- Search for a programme, record it and download subtitles:

```
get_iplayer "{{program_name}}" {{[-g|--get]}} --subtitles
```

- Display help:

```
get_iplayer {{[-h|--help]}}
```

GetADUsers.py

Retrieve a list of users from Active Directory, including attributes like last logon timestamp and email.

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- Enumerate all Active Directory users and their attributes:

```
GetADUsers.py -all -dc-ip {{domain_controller_ip}} {{domain}}/{{username}}:{{password}}
```

- Retrieve information only for a specific user:

```
GetADUsers.py -user {{user}} -dc-ip {{domain_controller_ip}} {{domain}}/{{username}}:{{password}}
```

- Extract user details using pass-the-hash authentication:

```
GetADUsers.py -all -dc-ip {{domain_controller_ip}} -hashes {{LM_Hash}}:{{NT_Hash}} {{domain}}/{{username}}
```

- Save output to a file:

```
GetADUsers.py -all -dc-ip {{domain_controller_ip}} {{domain}}/{{username}}:{{password}} > {{path/to/output.txt}}
```

getArch.py

Determine the OS architecture (x86 or x64) of a remote Windows system.

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- Check the architecture of a single target system:

```
getArch.py -target {{target}}
```

- Check the architecture of multiple targets from a file (one per line):

```
getArch.py -targets {{path/to/targets_file}}
```

- Set a custom socket timeout (default is 2 seconds):

```
getArch.py -target {{target}} -timeout {{seconds}}
```

- Enable debug mode for detailed output:

```
getArch.py -target {{target}} -debug
```

geth

The go-ethereum command-line interface.

More information: https://geth.ethereum.org/docs/fundamentals/command-line-options.

- Connect to the main Ethereum network and automatically download the full node:

```
geth
```

- Connect to the Ropsten test network:

```
geth --testnet
```

- Create a new account:

```
geth account new
```

- Enable mining:

```
geth --mine
```

GetNPUsers.py

Enumerate Active Directory accounts with Kerberos pre-authentication disabled, which may be susceptible to AS-REP roasting attacks.

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- Enumerate users with Kerberos pre-authentication disabled (default anonymous enumeration):

```
GetNPUsers.py {{domain}}/ -usersfile {{path/to/userslist}} -dc-ip {{domain_controller_ip}} -no-pass
```

- Perform AS-REP roasting and dump crackable hashes for offline cracking:

```
GetNPUsers.py {{domain}}/ -usersfile {{path/to/userslist}} -dc-ip {{domain_controller_ip}} -no-pass -request
```

- Authenticate with valid credentials (if anonymous binding is disabled):

```
GetNPUsers.py {{domain}}/{{username}}:{{password}} -usersfile {{path/to/userslist}} -dc-ip {{domain_controller_ip}}
```

- Use pass-the-hash authentication instead of a password:

```
GetNPUsers.py {{domain}}/{{username}} -hashes {{LM_Hash}}:{{NT_Hash}} -usersfile {{path/to/userslist}} -dc-ip {{domain_controller_ip}}
```

- Save the output to a file for further analysis:

```
GetNPUsers.py {{domain}}/ -usersfile {{path/to/userslist}} -dc-ip {{domain_controller_ip}} -request > {{path/to/output.txt}}
```

getopts

Parse shell options from arguments.

This command does not support longform options and thus using **`getopt`** is recommended instead.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-getopts.

- Check if an option is the first set option in the current context:

```
getopts {{x}} {{opt}}; echo ${{opt}}
```

- Check if an option is set in a string (specified option must be the first element of the string):

```
getopts {{x}} {{opt}} "{{string text}}"; echo ${{opt}}
```

- Set an option to require an argument and print them:

```
getopts {{x}}: {{opt}}; echo ${{opt}} $OPTARG
```

- Check for multiple options:

```
while getopts {{xyz}} {{opt}}; do case ${{opt}} in x) {{echo
x is set}};; y) {{echo y is set}};; z) {{echo z is set}};;
esac; done
```

- Set `getopts` to silent mode and handle option errors:

```
while getopts :{{x:}} {{opt}}; do case ${{opt}} in x) ;; :)
{{echo "Argument required"}};; ?) {{echo "Invalid argument"}}
esac;; done
```

- Reset `getopts`:

```
OPTIND=1
```

gettext

Translates a string using stored translations in a compiled **`.mo`** file.

Translations are stored in **`/usr/share/locale/locale_name/LC_MESSAGES/`** with **`domain`** being the filename without its extension.

See also: **`msgfmt`**, **`msgunfmt`**.

More information: https://www.gnu.org/software/gettext/manual/gettext.html#gettext-Invocation.

- Get the translation of a string as specified in the domain file (falls back to given `msgid` if no translation exists):

```
LANGUAGE={{locale}} gettext {{[-d|--domain]}} {{domain}} "{{msgid}}"
```

- Display help:

```
gettext {{[-h|--help]}}
```

- Display gettext version:

```
gettext {{[-V|--version]}}
```

GetUserSPNs.py

Retrieve Service Principal Names (SPNs) associated with Active Directory user accounts.

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- Enumerate user accounts with an SPN and request their Kerberos TGS tickets:

```
GetUserSPNs.py {{domain}}/{{username}}:{{password}} -dc-ip
{{domain_controller_ip}}
```

- Use pass-the-hash authentication:

```
GetUserSPNs.py {{domain}}/{{username}} -hashes {{LM_Hash}}:
{{NT_Hash}} -dc-ip {{domain_controller_ip}}
```

- Save the output to a file:

```
GetUserSPNs.py {{domain}}/{{username}}:{{password}} -dc-ip
{{domain_controller_ip}} -outputfile {{path/to/output_file}}
```

- Request only TGS tickets:

```
GetUserSPNs.py {{domain}}/{{username}}:{{password}} -dc-ip
{{domain_controller_ip}} -request
```

- Request only TGS tickets using pass-the-hash authentication:

```
GetUserSPNs.py {{domain}}/{{username}} -dc-ip
{{domain_controller_ip}} -hashes {{LM_Hash}}:{{NT_Hash}} -
request
```

gfortran

Preprocess and compile Fortran source files, then assemble and link them together.

More information: https://gcc.gnu.org/onlinedocs/gfortran/Invoking-GNU-Fortran.html.

- Compile multiple source files into an executable:

```
gfortran {{path/to/source1.f90 path/to/source2.f90 ...}} -o {{path/to/output_executable}}
```

- Show common warnings, debug symbols in output, and optimize without affecting debugging:

```
gfortran {{path/to/source.f90}} -Wall -g -Og -o {{path/to/output_executable}}
```

- Include libraries from a different path:

```
gfortran {{path/to/source.f90}} -o {{path/to/output_executable}} -I{{path/to/mod_and_include}} -L{{path/to/library}} -l{{library_name}}
```

- Compile source code into Assembler instructions:

```
gfortran -S {{path/to/source.f90}}
```

- Compile source code into an object file without linking:

```
gfortran -c {{path/to/source.f90}}
```

gh a11y

This command is an alias of **`gh accessibility`**.

- View documentation for the original command:

```
tldr gh accessibility
```

gh accessibility

Learn about GitHub CLI's accessibility experiences.

More information: https://cli.github.com/manual/.

- Open the GitHub Accessibility site in your browser:

```
gh {{[a11y|accessibility]}} {{[-w|--web]}}
```

gh agent-task

Manage GitHub agent tasks.

More information: https://cli.github.com/manual/gh_agent-task.

- List the most recent agent tasks:

```
gh {{[agent|agent-task]}} list
```

- Create a new agent task for the current repository:

```
gh {{[agent|agent-task]}} create "{{Improve the performance of the data processing pipeline}}"
```

- Create a new agent task from a file:

```
gh {{[agent|agent-task]}} create {{[-F|--from-file]}} {{path/to/file}}
```

- View details about a specific agent task:

```
gh {{[agent|agent-task]}} view {{session_id|pr_number|url|branch}}
```

- Show the log of a specific agent task:

```
gh {{[agent|agent-task]}} view --log {{session_id|pr_number|url|branch}}
```

gh agent

This command is an alias of **`gh agent-task`**.

- View documentation for the original command:

```
tldr gh agent-task
```

gh alias

Manage GitHub CLI command aliases.

More information: https://cli.github.com/manual/gh_alias.

- List all the aliases `gh` is configured to use:

```
gh alias {{[ls|list]}}
```

- Create a `gh` subcommand alias:

```
gh alias set {{pv}} '{{pr view}}'
```

- Set a shell command as a `gh` subcommand:

```
gh alias set {{[-s|--shell]}} {{alias_name}} {{command}}
```

- Delete a command shortcut:

```
gh alias delete {{alias_name}}
```

- Display the subcommand help:

```
gh alias
```

gh api

Make authenticated HTTP requests to the GitHub API and print the response.

More information: https://cli.github.com/manual/gh_api.

- Display the releases for the current repository in JSON format:

```
gh api repos/:owner/:repo/releases
```

- Create a reaction for a specific issue:

```
gh api {{[-H|--header]}} {{Accept:application/
vnd.github.squirrel-girl-preview+json}} {{[-f|--raw-field]}}
'{{content=+1}}' {{repos/:owner/:repo/issues/123/reactions}}
```

- Display the result of a GraphQL query in JSON format:

```
gh api graphql {{[-f|--field]}} {{name=':repo'}} {{[-f|--raw-
field]}} '{{query}}'
```

- Send a request using a custom HTTP method:

```
gh api {{[-X|--method]}} {{POST}} {{endpoint}}
```

- Include the HTTP response headers in the output:

```
gh api {{[-i|--include]}} {{endpoint}}
```

- Do not print the response body:

```
gh api --silent {{endpoint}}
```

- Send a request to a specific GitHub Enterprise Server:

```
gh api --hostname {{github.example.com}} {{endpoint}}
```

- Display the subcommand help:

```
gh api --help
```

gh at

This command is an alias of **`gh attestation`**.

- View documentation for the original command:

```
tldr gh attestation
```

gh attestation

Download and verify artifact attestations to ensure their integrity and provenance.

More information: https://cli.github.com/manual/gh_attestation.

- Download attestations for a local file associated with a specific repository:

```
gh {{[at|attestation]}} download {{path/to/artifact.bin}} {{[-R|--repo]}} {{owner}}/{{repo}}
```

- Download attestations for an OCI container image associated with an organization:

```
gh {{[at|attestation]}} download oci://{{image_uri}} {{[-o|--owner]}} {{organization_name}}
```

- Verify a local artifact online against attestations from a specific repository:

```
gh {{[at|attestation]}} verify {{path/to/artifact.bin}} {{[-R|--repo]}} {{owner}}/{{repo}}
```

- Verify an artifact, requiring it was signed by a specific reusable workflow for enhanced security:

```
gh {{[at|attestation]}} verify {{path/to/artifact.bin}} {{[-o|--owner]}} {{organization_name}} --signer-workflow {{owner}}/{{repo}}/{{path/to/workflow.yml}}
```

- Verify an artifact and output the detailed verification results as JSON for use in policy engines:

```
gh {{[at|attestation]}} verify {{path/to/artifact.bin}} {{[-o|--owner]}} {{organization_name}} --format json
```

- Perform a fully offline verification using a downloaded bundle and a custom trusted root file:

```
gh {{[at|attestation]}} verify {{path/to/artifact.bin}} {{[-b|--bundle]}} {{path/to/bundle.jsonl}} --custom-trusted-root {{path/to/trusted_root.jsonl}}
```

- Save the trusted root of signing certificates to a file for offline verification:

```
gh {{[at|attestation]}} trusted-root > {{path/to/trusted_root.jsonl}}
```

gh auth

Authenticate with a GitHub host.

More information: https://cli.github.com/manual/gh_auth.

- Log in with interactive prompt:

```
gh auth login
```

- Log in with a token from `stdin` (created in https://github.com/settings/tokens):

```
echo {{your_token}} | gh auth login --with-token
```

- Check if you are logged in:

```
gh auth status
```

- Log out:

```
gh auth logout
```

- Log in with a specific GitHub Enterprise Server:

```
gh auth login {{[-h|--hostname]}} {{github.example.com}}
```

- Refresh the session to ensure authentication credentials have the correct minimum scopes (removes additional scopes requested previously):

```
gh auth refresh
```

- Expand the permission scopes:

```
gh auth refresh {{[-s|--scopes]}}
{{repo,admin:repo_hook,admin:org,admin:public_key,admin:org_hook,...}}
```

gh browse

Open a GitHub repository in the browser or print the URL.

More information: https://cli.github.com/manual/gh_browse.

- Open the homepage of the current repository in the default web browser:

```
gh browse
```

- Open the homepage of a specific repository in the default web browser:

```
gh browse {{owner}}/{{repository}}
```

- Open the settings page of the current repository in the default web browser:

```
gh browse {{[-s|--settings]}}
```

- Open the wiki of the current repository in the default web browser:

```
gh browse {{[-w|--wiki]}}
```

- Open a specific issue or pull request in the web browser:

```
gh browse {{issue_number|pull_request_number}}
```

- Open a specific branch in the web browser:

```
gh browse {{[-b|--branch]}} {{branch_name}}
```

- Open a specific file or directory of the current repository in the web browser:

```
gh browse {{path/to/file_or_directory}}
```

- Print the destination URL without open the web browser:

```
gh browse {{[-n|--no-browser]}}
```

gh cache

Manage GitHub Actions caches.

More information: https://cli.github.com/manual/gh_cache.

- List caches for the current repository:

```
gh cache {{[ls|list]}}
```

- List caches for a specific repository:

```
gh cache {{[ls|list]}} {{[-R|--repo]}} {{owner}}/
{{repository}}
```

- List caches with a specific cache key prefix:

```
gh cache {{[ls|list]}} {{[-k|--key]}} {{key_prefix}}
```

- List caches for a specific branch:

```
gh cache {{[ls|list]}} {{[-r|--ref]}} refs/heads/
{{branch_name}}
```

- List caches sorted by least recently accessed:

```
gh cache {{[ls|list]}} {{[-S|--sort]}} last_accessed_at {{[-
O|--order]}} asc
```

- Delete a cache by id:

```
gh cache delete {{cache_id}}
```

- Delete a cache by key:

```
gh cache delete {{cache_key}}
```

- Delete all caches:

```
gh cache delete {{[-a|--all]}}
```

gh codespace

Connect and manage your codespaces in GitHub.

More information: https://cli.github.com/manual/gh_codespace.

- Create a codespace in GitHub interactively:

```
gh {{[cs|codespace]}} create
```

- List all available codespaces:

```
gh {{[cs|codespace]}} {{[ls|list]}}
```

- Connect to a codespace via SSH interactively:

```
gh {{[cs|codespace]}} ssh
```

- Transfer a specific file to a codespace interactively:

```
gh {{[cs|codespace]}} cp {{path/to/source_file}} remote:
{{path/to/remote_file}}
```

- List the ports of a codespace interactively:

```
gh {{[cs|codespace]}} ports
```

- Display the logs from a codespace interactively:

```
gh {{[cs|codespace]}} logs
```

- Delete a codespace interactively:

```
gh {{[cs|codespace]}} delete
```

- Display help for a subcommand:

```
gh {{[cs|codespace]}} {{code|cp|create|delete|edit|...}} {{[-
h|--help]}}
```

gh completion

Generate shell completion scripts for GitHub CLI commands.

More information: https://cli.github.com/manual/gh_completion.

- Print a completion script:

```
gh completion {{[-s|--shell]}} {{bash|zsh|fish|powershell}}
```

- Append the `gh` completion script to `~/.bashrc`:

```
gh completion {{[-s|--shell]}} bash >> ~/.bashrc
```

- Append the `gh` completion script to `~/.zshrc`:

```
gh completion {{[-s|--shell]}} zsh >> ~/.zshrc
```

- Display the subcommand help:

```
gh completion
```

gh config

Change configuration for GitHub CLI.

More information: https://cli.github.com/manual/gh_config.

- Display what Git protocol is being used:

```
gh config get git_protocol
```

- Set protocol to SSH:

```
gh config set git_protocol ssh
```

- Use `delta` in side-by-side mode as the default pager for all `gh` commands:

```
gh config set pager 'delta --side-by-side'
```

- Set text editor to Vim:

```
gh config set editor {{vim}}
```

- Reset to default text editor:

```
gh config set editor ""
```

- Disable interactive prompts:

```
gh config set prompt {{disabled}}
```

- Set a specific configuration value:

```
gh config set {{key}} {{value}}
```

gh copilot

Interact with GitHub Copilot.

More information: https://github.com/github/gh-copilot#usage.

- Suggest a command, given a description:

```
gh copilot suggest "{{Install git}}"
```

- Suggest a Git command:

```
gh copilot suggest {{[-t|--target]}} git "{{Undo the most recent local commits}}"
```

- Explain a command:

```
gh copilot explain "{{traceroute example.com}}"
```

- Generate shell-specific aliases for Bash:

```
echo 'eval "$(gh copilot alias -- bash)"' >> ~/.bashrc
```

- Generate shell-specific aliases for Zsh:

```
echo 'eval "$(gh copilot alias -- zsh)"' >> ~/.zshrc
```

- Configure options:

```
gh copilot config
```

gh cs

This command is an alias of **`gh codespace`**.

- View documentation for the original command:

```
tldr gh codespace
```

gh environment

Display help about environment variables for the GitHub CLI command.

More information: https://cli.github.com/manual/gh_help_environment.

- Display help about environment variables that can be used with `gh`:

```
gh environment
```

gh extension

Manage extensions for the GitHub CLI.

More information: https://cli.github.com/manual/gh_extension.

- Initialize a new GitHub CLI extension project in a directory of the same name:

```
gh {{[ext|extension]}} create {{extension_name}}
```

- Install an extension from a GitHub repository:

```
gh {{[ext|extension]}} install {{owner}}/{{repository}}
```

- List installed extensions:

```
gh {{[ext|extension]}} {{[ls|list]}}
```

- Upgrade a specific extension:

```
gh {{[ext|extension]}} upgrade {{extension_name}}
```

- Upgrade all extensions:

```
gh {{[ext|extension]}} upgrade --all
```

- List installed extensions:

```
gh {{[ext|extension]}} {{[ls|list]}}
```

- Remove an extension:

```
gh {{[ext|extension]}} remove {{extension_name}}
```

- Display help about a subcommand:

```
gh {{[ext|extension]}} {{subcommand}} {{[-h|--help]}}
```

gh formatting

Formatting options for JSON data exported from gh GitHub CLI command.

More information: https://cli.github.com/manual/gh_help_formatting.

- Display help about formatting JSON output from `gh` using `jq`:

```
gh formatting
```

gh gist

Work with GitHub Gists.

More information: https://cli.github.com/manual/gh_gist.

- Create a new Gist from one or more files:

```
gh gist {{[new|create]}} {{path/to/file1 path/to/file2 ...}}
```

- Create a new Gist with a specific [desc]ription:

```
gh gist {{[new|create]}} {{path/to/file1 path/to/file2 ...}}
{{[-d|--desc]}} "{{description}}"
```

- Edit a Gist:

```
gh gist edit {{id|url}}
```

- List up to 42 Gists owned by the currently logged in user:

```
gh gist {{[ls|list]}} {{[-L|--limit]}} 42
```

- View a Gist in the default browser without rendering Markdown:

```
gh gist view {{id|url}} {{[-w|--web]}} {{[-r|--raw]}}
```

gh gpg-key

Manage GPG keys registered with the authorized GitHub account.

See also: **gpg**.

More information: https://cli.github.com/manual/gh_gpg-key.

- List GPG keys in the authorized GitHub account:

```
gh gpg-key {{[ls|list]}}
```

- Add a GPG key to the authorized GitHub account by specifying the key file:

```
gh gpg-key add {{path/to/key_file}}
```

- Add a GPG key to the authorized GitHub account by specifying the key ID:

```
gpg {{[-a|--armor]}} --export {{key_id}} | gh gpg-key add -
```

- Delete a GPG key from the authorized GitHub account:

```
gh gpg-key delete {{key_id}}
```

gh help

Display help about the GitHub CLI command.

More information: https://cli.github.com/manual/gh_help.

- Display general help:

```
gh help
```

- Display help for the `gh help` subcommand:

```
gh help {{[-h|--help]}}
```

- Display help about environment variables that can be used with `gh`:

```
gh help environment
```

- Display a markdown reference of all `gh` commands:

```
gh help reference
```

- Display help about formatting JSON output from `gh` using `jq`:

```
gh help formatting
```

- Display help about using `gh` with MinTTY:

```
gh help mintty
```

- Display help for a subcommand:

```
gh help {{subcommand}}
```

- Display help for a subcommand action:

```
gh help {{pr}} {{create}}
```

gh issue create

Create GitHub issues on a repository.

More information: https://cli.github.com/manual/gh_issue_create.

- Create a new issue against the current repository interactively:

```
gh issue {{[new|create]}}
```

- Create a new issue with the bug label interactively:

```
gh issue {{[new|create]}} {{[-l|--label]}} "bug"
```

- Create a new issue interactively and assign it to the specified users:

```
gh issue {{[new|create]}} {{[-a|--assignee]}}
{{user1,user2,...}}
```

- Create a new issue with a title, body and assign it to the current user:

```
gh issue {{[new|create]}} {{[-t|--title]}} "{{title}}" {{[-
b|--body]}} "{{body}}" {{[-a|--assignee]}} "@me"
```

- Create a new issue interactively, reading the body text from a file:

```
gh issue {{[new|create]}} {{[-F|--body-file]}} {{path/to/
file}}
```

- Create a new issue in the default web browser:

```
gh issue {{[new|create]}} {{[-w|--web]}}
```

- Display help:

```
gh issue {{[new|create]}} {{[-h|--help]}}
```

gh issue

Manage GitHub issues.

More information: https://cli.github.com/manual/gh_issue.

- Display a specific issue:

```
gh issue view {{issue_number}}
```

- Display a specific issue in the default web browser:

```
gh issue view {{issue_number}} {{[-w|--web]}}
```

- Create a new issue in the default web browser:

```
gh issue {{[new|create]}} {{[-w|--web]}}
```

- List the last 10 issues with the bug label:

```
gh issue {{[ls|list]}} {{[-L|--limit]}} 10 {{[-l|--label]}}
"bug"
```

- List closed issues made by a specific user:

```
gh issue {{[ls|list]}} {{[-s|--state]}} closed {{[-A|--
author]}} {{username}}
```

- Display the status of issues relevant to the user, in a specific repository:

```
gh issue status {{[-R|--repo]}} {{owner}}/{{repository}}
```

- Reopen a specific issue:

```
gh issue reopen {{issue_number}}
```

gh label

Work with GitHub labels.

More information: https://cli.github.com/manual/gh_label.

- List labels for the repository in the current directory:

```
gh label {{[ls|list]}}
```

- View labels for the repository in the current directory in the default web browser:

```
gh label {{[ls|list]}} {{[-w|--web]}}
```

- Create a label with a specific name, description and color in hexadecimal format for the repository in the current directory:

```
gh label create {{name}} {{[-d|--description]}} "{{description}}" {{[-c|--color]}} {{color_hex}}
```

- Delete a label for the repository in the current directory, prompting for confirmation:

```
gh label delete {{name}}
```

- Update the name and description for a specific label for the repository in the current directory:

```
gh label edit {{name}} {{[-n|--name]}} {{new_name}} {{[-d|--description]}} "{{description}}"
```

- Clone labels from a specific repository into the repository in the current directory:

```
gh label clone {{owner}}/{{repository}}
```

- Display help for a subcommand:

```
gh label {{subcommand}} {{[-h|--help]}}
```

gh mintty

Display help about MinTTY integration for the GitHub CLI command.

More information: https://cli.github.com/manual/gh_help_mintty.

- Display help about using `gh` with MinTTY:

```
gh mintty
```

gh org

Work with GitHub organizations.

More information: https://cli.github.com/manual/gh_org.

- List organizations for the authenticated user:

```
gh org {{[ls|list]}}
```

- List a certain amount of organizations (defaults to 30):

```
gh org {{[ls|list]}} {{[-L|--limit]}} {{100}}
```

- Display help:

```
gh org --help
```

gh pr checks

View CI checks for a GitHub pull request.

More information: https://cli.github.com/manual/gh_pr_checks.

- Show checks for the pull request of the current branch:

```
gh pr checks
```

- Show checks for a specific pull request:

```
gh pr checks {{pr_number}}
```

- Watch checks and update in real time until completion:

```
gh pr checks {{pr_number}} --watch
```

- Show only required checks:

```
gh pr checks {{pr_number}} --required
```

gh pr comment

Add a comment to a GitHub pull request.

More information: https://cli.github.com/manual/gh_pr_comment.

- Comment on the pull request of the current branch:

```
gh pr comment {{[-b|--body]}} "{{LGTM}}"
```

- Comment on a specific pull request:

```
gh pr comment {{123}} {{[-b|--body]}} "{{Thanks!}}"
```

- Comment from a file:

```
gh pr comment {{123}} {{[-F|--body-file]}} {{path/to/
file.txt}}
```

- Open the editor to write a multi-line comment:

```
gh pr comment {{123}}
```

gh pr create

Manage GitHub pull requests.

More information: https://cli.github.com/manual/gh_pr_create.

- Interactively create a pull request:

```
gh pr {{[new|create]}}
```

- Create a pull request, determining the title and description from the commit messages of the current branch:

```
gh pr {{[new|create]}} {{[-f|--fill]}}
```

- Create a draft pull request:

```
gh pr {{[new|create]}} {{[-d|--draft]}}
```

- Create a pull request specifying the base branch, title, and description:

```
gh pr {{[new|create]}} {{[-B|--base]}} {{base_branch}} {{[-t|--title]}} "{{title}}" {{[-b|--body]}} "{{body}}"
```

- Start opening a pull request in the default web browser:

```
gh pr {{[new|create]}} {{[-w|--web]}}
```

gh pr merge

Merge a GitHub pull request.

More information: https://cli.github.com/manual/gh_pr_merge.

- Merge the pull request associated with the current branch interactively:

```
gh pr merge
```

- Merge the current branch into the specified pull request:

```
gh pr merge {{pr_number}} {{[-m|--merge]}}
```

- Squash and merge a pull request, then delete the branch:

```
gh pr merge {{pr_number}} {{[-sd|--squash --delete-branch]}}
```

- Rebase and merge:

```
gh pr merge {{pr_number}} {{[-r|--rebase]}}
```

- Enable auto-merge (squash):

```
gh pr merge {{pr_number}} --auto {{[-s|--squash]}}
```

- Merge with admin privileges (if allowed):

```
gh pr merge {{pr_number}} --admin
```

gh pr view

View details of a GitHub pull request.

More information: https://cli.github.com/manual/gh_pr_view.

- View the pull request associated with the current branch:

```
gh pr view
```

- View a specific pull request:

```
gh pr view {{123}}
```

- Open the pull request in the default web browser:

```
gh pr view {{[-w|--web]}}
```

- Show comments in the terminal:

```
gh pr view {{123}} {{[-c|--comments]}}
```

gh pr

Manage GitHub pull requests.

Some subcommands such as **`create`** have their own usage documentation.

More information: https://cli.github.com/manual/gh_pr.

- Create a pull request:

```
gh pr {{[new|create]}}
```

- Check out a specific pull request locally:

```
gh {{[co|pr checkout]}} {{pr_number|url|branch}}
```

- View the changes made in the pull request for the current branch:

```
gh pr diff
```

- Approve the pull request for the current branch:

```
gh pr review {{[-a|--approve]}}
```

- Merge the pull request associated with the current branch interactively:

```
gh pr merge
```

- Edit a pull request interactively:

```
gh pr edit
```

- Edit the base branch of a pull request:

```
gh pr edit {{[-B|--base]}} {{branch_name}}
```

- Check the status of the current repository's pull requests:

```
gh pr status
```

gh preview

Preview GitHub CLI programs for testing and development purposes.

These commands are unstable and may change at any time.

More information: https://cli.github.com/manual/gh_preview.

- Run a specific preview command:

```
gh preview {{command}}
```

gh project

Work with GitHub Projects.

More information: https://cli.github.com/manual/gh_project.

- List projects owned by the currently authenticated user:

```
gh project {{[ls|list]}}
```

- List projects owned by a specific user or organization:

```
gh project {{[ls|list]}} --owner {{owner}}
```

- View a project by number:

```
gh project view {{number}} --owner {{owner}}
```

- Create a new project:

```
gh project create --owner {{owner}} --title {{project_title}}
```

- Add an item (issue or pull request) to a project:

```
gh project item-add {{number}} --owner {{owner}} --url
{{issue_or_pr_url}}
```

- List items in a project:

```
gh project item-list {{number}} --owner {{owner}}
```

- Close a project:

```
gh project close {{number}} --owner {{owner}}
```

gh reference

Display a reference about the GitHub CLI command.

More information: https://cli.github.com/manual/gh_help_reference.

- Display a markdown reference of all `gh` commands:

```
gh reference
```

gh release

Manage GitHub releases.

More information: https://cli.github.com/manual/gh_release.

- List releases in a GitHub repository, limited to 30 items:

```
gh release {{[ls|list]}}
```

- Display information about a specific release:

```
gh release view {{tag}}
```

- Create a new release:

```
gh release {{[new|create]}} {{tag}}
```

- Delete a specific release:

```
gh release delete {{tag}}
```

- Download assets from a specific release:

```
gh release download {{tag}}
```

- Upload assets to a specific release:

```
gh release upload {{tag}} {{path/to/file1 path/to/file2 ...}}
```

gh repo create

Create a new GitHub repository.

Note: **`--public`**, **`--private`**, or **`--internal`** is required when not running interactively.

More information: https://cli.github.com/manual/gh_repo_create.

- Create a new repository interactively:

```
gh repo {{[new|create]}}
```

- Create a private repository from the current directory:

```
gh repo {{[new|create]}} {{[-s|--source]}} . --private
```

- Create a public repository from the current directory:

```
gh repo {{[new|create]}} {{[-s|--source]}} . --public
```

- Create a public repository with a specified name and description:

```
gh repo {{[new|create]}} {{repo_name}} {{[-d|--description]}} "{{repo_description}}" --public
```

- Clone the new repository locally after creation:

```
gh repo {{[new|create]}} {{repo_name}} {{[-c|--clone]}} {{--public|--private|--internal}}
```

gh repo

Work with GitHub repositories.

More information: https://cli.github.com/manual/gh_repo.

- Create a new repository interactively:

```
gh repo {{[new|create]}}
```

- Clone a repository:

```
gh repo clone {{owner}}/{{repository}}
```

- Fork and clone a repository:

```
gh repo fork {{owner}}/{{repository}} --clone
```

- View a repository in the default web browser:

```
gh repo view {{repository}} {{[-w|--web]}}
```

- List repositories owned by a specific user or organization (if the owner is not set, the default owner will be the currently logged in user):

```
gh repo {{[ls|list]}} {{owner}}
```

- List only non-forks repositories and limit the number of repositories to list (default: 30):

```
gh repo {{[ls|list]}} {{owner}} --source {{[-L|--limit]}} {{limit}}
```

- List repositories with a specific primary coding language:

```
gh repo {{[ls|list]}} {{owner}} {{[-l|--language]}} {{language_name}}
```

gh rs

This command is an alias of **`gh ruleset`**.

- View documentation for the original command:

```
tldr gh ruleset
```

gh ruleset

Manage GitHub repository rulesets.

More information: https://cli.github.com/manual/gh_ruleset.

- List all rulesets for the current repository:

```
gh {{[rs|ruleset]}} {{[ls|list]}}
```

- List all rulesets for a specific organization:

```
gh {{[rs|ruleset]}} {{[ls|list]}} {{[-o|--org]}}
{{organization_name}}
```

- Check the rules that apply to the current branch:

```
gh {{[rs|ruleset]}} check
```

- Check the rules that apply to a specific branch in another repository:

```
gh {{[rs|ruleset]}} check {{branch_name}} {{[-R|--repo]}}
{{owner}}/{{repo}}
```

- Interactively select and view a ruleset for the current repository:

```
gh {{[rs|ruleset]}} view
```

- View a specific ruleset by its ID:

```
gh {{[rs|ruleset]}} view {{ruleset_id}}
```

- View an organization-level ruleset by its ID:

```
gh {{[rs|ruleset]}} view {{ruleset_id}} {{[-o|--org]}}
{{organization_name}}
```

- Open the list of rulesets for a specific repository in the browser:

```
gh {{[rs|ruleset]}} {{[ls|list]}} {{[-R|--repo]}} {{owner}}/
{{repo}} {{[-w|--web]}}
```

gh run

View, run and watch recent GitHub Actions workflow runs.

More information: https://cli.github.com/manual/gh_run.

- Interactively select a run to see information about the jobs:

```
gh run view
```

- Display information about a specific run:

```
gh run view {{workflow_run_number}}
```

- Display information about the steps of a job:

```
gh run view {{[-j|--job]}} {{job_number}}
```

- Display the log of a job:

```
gh run view {{[-j|--job]}} {{job_number}} --log
```

- Check a specific workflow and exit with a non-zero status if the run failed:

```
gh run view {{workflow_run_number}} --exit-status && {{echo
"run pending or passed"}}
```

- Interactively select an active run and wait until it's done:

```
gh run watch
```

- Display the jobs for a run and wait until it's done:

```
gh run watch {{workflow_run_number}}
```

- Re-run a specific workflow:

```
gh run rerun {{workflow_run_number}}
```

gh screensaver

Extension for GitHub CLI that runs animated terminal screensavers.

See also: **`gh extension`**.

More information: https://github.com/vilmibm/gh-screensaver.

- Run a random screensaver:

```
gh screensaver
```

- Run a specific screensaver:

```
gh screensaver {{[-s|--saver]}} {{fireworks|life|marquee|
pipes|pollock|starfield}}
```

- Run the "marquee" screensaver with a specific text and font:

```
gh screensaver {{[-s|--saver]}} marquee -- --
message="{{message}}" --font={{font_name}}
```

- Run the "starfield" screensaver with a specific density and speed:

```
gh screensaver {{[-s|--saver]}} starfield -- --density
{{500}} --speed {{10}}
```

- List available screensavers:

```
gh screensaver {{[-l|--list]}}
```

gh search

Search across all of GitHub.

More information: https://cli.github.com/manual/gh_search.

- Search for code containing specific keywords:

```
gh search code {{keyword1 keyword2 ...}}
```

- Search for issues with a specific phrase:

```
gh search issues "{{search_phrase}}"
```

- Search commits by a specific author:

```
gh search commits --author {{username}}
```

- Search pull requests assigned to yourself that are still open:

```
gh search prs --assignee @me --state open
```

- Search repositories in an organization by topic:

```
gh search repos --owner {{org_name}} --topic {{topic_name}}
```

- Search issues without a certain label (Unix-like systems):

```
gh search issues -- "{{search_query}} -label:{{label_name}}"
```

- Search issues without a certain label (PowerShell):

```
gh --% search issues -- "{{search_query}} -label:
{{label_name}}"
```

- Open the search query in the web browser:

```
gh search {{subcommand}} {{[-w|--web]}} {{query}}
```

gh secret set

Create or update GitHub secrets.

More information: https://cli.github.com/manual/gh_secret_set.

- Set a secret for the current repository (user will be prompted for the value):

```
gh secret set {{name}}
```

- Set a secret from a file for the current repository:

```
gh < {{path/to/file}} secret set {{name}}
```

- Set a secret for a specific repository:

```
gh secret set {{name}} {{[-b|--body]}} {{value}} {{[-R|--repo]}} {{owner}}/{{repository}}
```

- Set an organization secret for specific repositories:

```
gh secret set {{name}} {{[-o|--org]}} {{organization}} {{[-r|--repos]}} "{{repository1,repository2,...}}"
```

- Set an organization secret with a specific visibility:

```
gh secret set {{name}} {{[-o|--org]}} {{organization}} {{[-v|--visibility]}} {{all|private|selected}}
```

gh secret

Manage GitHub secrets.

More information: https://cli.github.com/manual/gh_secret.

- List secret keys for the current repository:

```
gh secret {{[ls|list]}}
```

- List secret keys for a specific organization:

```
gh secret {{[ls|list]}} {{[-o|--org]}} {{organization}}
```

- List secret keys for a specific repository:

```
gh secret {{[ls|list]}} {{[-R|--repo]}} {{owner}}/
{{repository}}
```

- Set a secret for the current repository (user will be prompted for the value):

```
gh secret set {{name}}
```

- Set a secret from a file for the current repository:

```
gh < {{path/to/file}} secret set {{name}}
```

- Set an organization secret for specific repositories:

```
gh secret set {{name}} {{[-o|--org]}} {{organization}} {{[-
r|--repos]}} {{repository1,repository2}}
```

- Remove a secret for the current repository:

```
gh secret remove {{name}}
```

- Remove a secret for a specific organization:

```
gh secret remove {{name}} {{[-o|--org]}} {{organization}}
```

gh skyline

Generate a 3D model of your GitHub contribution history.

By default, it will create a **`{username}-{year}-github-skyline.stl`** file in the current directory.

More information: https://github.com/github/gh-skyline.

- Generate a skyline STL file for the current year and authenticated user:

```
gh skyline
```

- Generate a skyline for a specific user and year:

```
gh skyline {{[-u|--user]}} {{username}} {{[-y|--year]}} {{year}}
```

- Generate a skyline for a range of years:

```
gh skyline {{[-u|--user]}} {{username}} {{[-y|--year]}} {{first_year}}-{{last_year}}
```

- Generate a full skyline (from the user's join year to the current year):

```
gh skyline {{[-u|--user]}} {{username}} {{[-f|--full]}}
```

- Enable debug logging:

```
gh skyline {{[-d|--debug]}}
```

- Generate a skyline and specify the output file path:

```
gh skyline {{[-o|--output]}} {{path/to/output_file.stl}}
```

- Open the GitHub profile for a specific user:

```
gh skyline {{[-u|--user]}} {{username}} {{[-w|--web]}}
```

- Display help:

```
gh skyline {{[-h|--help]}}
```

gh ssh-key

Manage GitHub SSH keys.

More information: https://cli.github.com/manual/gh_ssh-key.

- List SSH keys for the currently authenticated user:

```
gh ssh-key {{[ls|list]}}
```

- Add an SSH key to the currently authenticated user's account:

```
gh ssh-key add {{path/to/key.pub}}
```

- Add an SSH key to the currently authenticated user's account with a specific title:

```
gh ssh-key add {{[-t|--title]}} {{title}} {{path/to/key.pub}}
```

- Display help:

```
gh ssh-key
```

gh status

Check the status of subscribed repositories on GitHub.

More information: https://cli.github.com/manual/gh_status.

- Print information about all repositories you're subscribed to (issues, pull requests, reviews, mentions, activity, etc.):

```
gh status
```

- Exclude certain repositories:

```
gh status {{[-e|--exclude]}} {{repositories}}
```

- Print status of repositories belonging to an organization:

```
gh status {{[-o|--org]}} {{organization}}
```

gh variable

Manage GitHub Actions and Dependabot variables.

More information: https://cli.github.com/manual/gh_variable.

- List variables for the current repository:

```
gh variable {{[ls|list]}}
```

- List variables for a specific organization:

```
gh variable {{[ls|list]}} {{[-o|--org]}} {{organization}}
```

- Get a variable for the current repository:

```
gh variable get {{name}}
```

- Set a variable for the current repository (user will be prompted for the value):

```
gh variable set {{name}}
```

- Set a variable for a deployment environment in the current repository:

```
gh variable set {{name}} {{[-e|--env]}} {{environment_name}}
```

- Set an organization variable visible to all repositories:

```
gh variable set {{name}} {{[-o|--org]}} {{organization}} {{[-
v|--visibility]}} all
```

- Set multiple variables from a dotenv file:

```
gh variable set {{[-f|--env-file]}} {{path/to/file.env}}
```

- Delete a variable for the current repository:

```
gh variable delete {{name}}
```

gh workflow

List, view, and run GitHub Actions workflows.

More information: https://cli.github.com/manual/gh_workflow.

- Interactively select a workflow to view the latest jobs for:

```
gh workflow view
```

- View a specific workflow in the default browser:

```
gh workflow view {{id|workflow_name|filename.yml}} {{[-w|--web]}}
```

- Display the YAML definition of a specific workflow:

```
gh workflow view {{id|workflow_name|filename.yml}} {{[-y|--yaml]}}
```

- Display the YAML definition for a specific Git branch or tag:

```
gh workflow view {{id|workflow_name|filename.yml}} {{[-r|--ref]}} {{branch|tag_name}} {{[-y|--yaml]}}
```

- List workflow files (use `--all` to include disabled workflows):

```
gh workflow {{[ls|list]}}
```

- Run a manual workflow with parameters:

```
gh workflow run {{id|workflow_name|filename.yml}} {{--raw-field param1=value1 --raw-field param2=value2 ...}}
```

- Run a manual workflow using a specific branch or tag with JSON parameters from `stdin`:

```
echo '{{{"param1": "value1", "param2": "value2", ...}}}' | gh workflow run {{id|workflow_name|filename.yml}} {{[-r|--ref]}} {{branch|tag_name}}
```

- Enable or disable a specific workflow:

```
gh workflow {{enable|disable}} {{id|workflow_name|filename.yml}}
```

gh

Work seamlessly with GitHub.

Some subcommands such as **`config`** have their own usage documentation.

More information: https://cli.github.com/manual/gh.

- Clone a GitHub repository locally:

```
gh repo clone {{owner}}/{{repository}}
```

- Create a new issue:

```
gh issue {{[new|create]}}
```

- View and filter the open issues of the current repository:

```
gh issue {{[ls|list]}}
```

- View an issue in the default web browser:

```
gh issue view {{[-w|--web]}} {{issue_number|url}}
```

- Create a pull request:

```
gh pr {{[new|create]}}
```

- View a pull request in the default web browser:

```
gh pr view {{[-w|--web]}} {{pr_number|url|branch}}
```

- Check out a specific pull request locally:

```
gh {{[co|pr checkout]}} {{pr_number|url|branch}}
```

- Check the status of a repository's pull requests:

```
gh pr status
```

ghc

The Glasgow Haskell Compiler.

Compiles and links Haskell source files.

More information: https://downloads.haskell.org/ghc/latest/docs/users_guide/usage.html.

- Find and compile all modules in the current directory:

```
ghc Main
```

- Compile a single file:

```
ghc {{path/to/file.hs}}
```

- Compile using extra optimization:

```
ghc -O {{path/to/file.hs}}
```

- Stop compilation after generating object files (.o):

```
ghc -c {{path/to/file.hs}}
```

- Start a REPL (interactive shell):

```
ghci
```

- Evaluate a single expression:

```
ghc -e {{expression}}
```

ghci

The Glasgow Haskell Compiler's interactive environment.

More information: https://downloads.haskell.org/ghc/latest/docs/html/users_guide/ghci.html.

- Start a REPL (interactive shell):

```
ghci
```

- Start a REPL and load the specified Haskell source file:

```
ghci {{source_file.hs}}
```

- Start a REPL and enable a language option:

```
ghci -X{{language_option}}
```

- Start a REPL and enable some level of compiler warnings (e.g. `all` or `compact`):

```
ghci -W{{warning_level}}
```

- Start a REPL with a colon-separated list of directories for finding source files:

```
ghci -i{{path/to/directory1:path/to/directory2:...}}
```

ghcid

Simple and efficient CLI IDE for Haskell that reloads code on file changes.

Continuously displays compile errors, warnings, and test results.

More information: https://github.com/ndmitchell/ghcid.

- Start `ghcid` and monitor a Haskell file for changes:

```
ghcid {{path/to/Main.hs}}
```

- Start `ghcid` with a specific command, such as loading a Stack or Cabal project:

```
ghcid --command "{{stack ghci Main.hs}}"
```

- Run an action (default `main`) on each file save:

```
ghcid --run={{action}} {{path/to/Main.hs}}
```

- Set maximum height and width (default to console height and width):

```
ghcid --height={{height}} --width={{width}} {{path/to/
Main.hs}}
```

- Write full GHC compiler output to a file:

```
ghcid --outputfile={{path/to/output_file.txt}} {{path/to/
Main.hs}}
```

- Execute REPL commands (eg. `-- $> 1+1`) on each file save:

```
ghcid --allow-eval {{path/to/Main.hs}}
```

ghcup

Haskell toolchain installer.

Install, manage, and update Haskell toolchains.

More information: https://gitlab.haskell.org/haskell/ghcup-hs.

- Start the interactive TUI:

```
ghcup tui
```

- List available GHC/Cabal versions:

```
ghcup list
```

- Install the recommended GHC version:

```
ghcup install ghc
```

- Install a specific GHC version:

```
ghcup install ghc {{version}}
```

- Activate a specific GHC version:

```
ghcup set ghc {{version}}
```

- Install cabal-install:

```
ghcup install cabal
```

- Update `ghcup` itself:

```
ghcup upgrade
```

ghdl

Open-source simulator for the VHDL language.

More information: https://ghdl.github.io/ghdl/.

- Analyze a VHDL source file and produce an object file:

```
ghdl -a {{filename.vhdl}}
```

- Elaborate a design (where `design` is the name of a configuration unit, entity unit or architecture unit):

```
ghdl -e {{design}}
```

- Run an elaborated design:

```
ghdl -r {{design}}
```

- Run an elaborated design and dump output to a waveform file:

```
ghdl -r {{design}} --wave={{output.ghw}}
```

- Check the syntax of a VHDL source file:

```
ghdl -s {{filename.vhdl}}
```

- Display help:

```
ghdl --help
```

ghost

A blogging platform and headless CMS.

More information: https://ghost.org.

- Install Ghost in the current directory:

```
ghost install
```

- Start an instance of Ghost:

```
ghost start
```

- Restart the Ghost instance:

```
ghost restart
```

- Check the system for any potential hiccups while installing or updating Ghost:

```
ghost doctor
```

- View the logs of a Ghost instance:

```
ghost log {{name}}
```

- Run a Ghost instance directly (used by process managers and for debugging):

```
ghost run
```

- View running Ghost processes:

```
ghost ls
```

- View or edit Ghost configuration:

```
ghost config {{key}} {{value}}
```

ghostty

A fast, feature-rich, and cross-platform terminal emulator that uses platform-native UI and GPU acceleration.

Note: All options from the configuration file can also be used on the command-line (using **`--option=argument`**).

More information: https://ghostty.org/docs/config/reference.

- Open a new Ghostty window (not supported on macOS):

  ```
  ghostty
  ```

- Run a specific command in a new Ghostty window (not supported on macOS):

  ```
  ghostty -e {{command}}
  ```

- List all default and configured keybindings:

  ```
  ghostty +list-keybinds
  ```

- List all actions (i.e. what can be triggered via keybindings):

  ```
  ghostty +list-actions
  ```

- Browse an interactive list of themes:

  ```
  ghostty +list-themes
  ```

- Print the default configuration (including comments):

  ```
  ghostty +show-config --default --docs
  ```

ghq

Manage remote repository clones organized by hostname and path.

More information: https://github.com/x-motemen/ghq.

- Clone a repository under the `ghq` root directory (default is `~/ghq`):

```
ghq get {{repository_url}}
```

- Clone a repository from a user/project format (defaults to GitHub):

```
ghq get {{user}}/{{project}}
```

- Clone a repository and `cd` into it:

```
ghq get {{repository_url}} --look
```

- Clone a repository via SSH:

```
ghq get {{user}}/{{project}} -p
```

- Update an existing repository to the latest version:

```
ghq get {{repository_url}} -u
```

- List all locally cloned repositories:

```
ghq list
```

- List locally cloned repositories with full paths:

```
ghq list --full-path
```

- Remove a locally cloned repository:

```
ghq rm {{user}}/{{project}}
```

gibo

Fetch gitignore boilerplates.

More information: https://github.com/simonwhitaker/gibo.

- List available boilerplates:

```
gibo list
```

- Write a boilerplate to `stdout`:

```
gibo dump {{boilerplate}}
```

- Write a boilerplate to .gitignore:

```
gibo dump {{boilerplate}} >>{{.gitignore}}
```

- Search for boilerplates containing a given string:

```
gibo search {{string}}
```

- Update available local boilerplates:

```
gibo update
```

gifdiff

Compare two GIFs for identical visual appearance.

See also: **`gifsicle`**.

More information: https://www.lcdf.org/gifsicle.

- Check how GIFs differ:

```
gifdiff {{path/to/first.gif}} {{path/to/second.gif}}
```

- Check if GIFs differ:

```
gifdiff --brief {{path/to/first.gif}} {{path/to/second.gif}}
```

gifsicle

Create, edit, manipulate, and get information about GIF files.

More information: https://www.lcdf.org/gifsicle.

- Optimize a GIF as a new file:

```
gifsicle {{path/to/input.gif}} {{[-O|--optimize=]}}3 {{[-o|--output]}} {{path/to/output.gif}}
```

- Use batch mode (modify each given file in place) and unoptimize a GIF:

```
gifsicle {{[-b|--batch]}} {{path/to/input.gif}} {{[-U|--unoptimize]}}
```

- Extract a frame from a GIF:

```
gifsicle {{path/to/input.gif}} '#{{0}}' > {{path/to/first_frame.gif}}
```

- Make a GIF animation from selected GIFs:

```
gifsicle {{*.gif}} {{[-d|--delay]}} {{10}} {{[-l|--loop]}} > {{path/to/output.gif}}
```

- Reduce file size using lossy compression:

```
gifsicle {{[-b|--batch]}} {{path/to/input.gif}} {{[-O|--optimize=]}}3 --lossy={{100}} {{[-k|--colors]}} {{16}} {{[-f|--dither]}}
```

- Delete the first 10 frames and all frames after frame 20 from a GIF:

```
gifsicle {{[-b|--batch]}} {{path/to/input.gif}} --delete '#{{0-9}}' '#{{20-}}'
```

- Modify all frames by cropping them to a rectangle, changing their scale, flipping them, and rotating them:

```
gifsicle {{[-b|--batch]}} --crop {{starting_x}},{{starting_y}}+{{rect_width}}x{{rect_height}} --scale {{0.25}} --flip-horizontal --rotate-{{90|180|270}} {{path/to/input.gif}}
```

giftopnm

Convert a GIF file into a PNM image.

More information: https://netpbm.sourceforge.net/doc/giftopnm.html.

- Convert a GIF image to a Netpbm image pixel-for-pixel:

```
giftopnm {{path/to/input.gif}} > {{path/to/output.pnm}}
```

- Display version:

```
giftopnm {{[-v|-version]}}
```

gimp

GNU image manipulation program.

See also: **krita**.

More information: https://docs.gimp.org/en/gimp-fire-up.html#gimp-concepts-running-command-line.

- Start GIMP:

```
gimp
```

- Open specific files:

```
gimp {{path/to/image1 path/to/image2 ...}}
```

- Open specific files in a new window:

```
gimp --new-instance {{path/to/image1 path/to/image2 ...}}
```

- Start without a splash screen:

```
gimp --no-splash
```

- Print errors and warnings to the console instead of showing them in a dialog box:

```
gimp --console-messages
```

- Enable debugging signal handlers:

```
gimp --debug-handlers
```

gist

Upload code to https://gist.github.com.

More information: https://github.com/defunkt/gist.

- Log in to gist on this computer:

```
gist --login
```

- Create a gist from any number of text files:

```
gist {{file.txt}} {{file2.txt}}
```

- Create a private gist with a description:

```
gist --private --description "{{A meaningful
description}}" {{file.txt}}
```

- Read contents from `stdin` and create a gist from it:

```
{{echo "hello world"}} | gist
```

- List your public and private gists:

```
gist --list
```

- List all public gists for any user:

```
gist --list {{username}}
```

- Update a gist using the ID from URL:

```
gist --update {{GIST_ID}} {{file.txt}}
```

git abort

Abort an ongoing rebase, merge, or cherry-pick.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-abort.

- Abort a Git rebase, merge, or cherry-pick:

```
git abort
```

git add

Adds changed files to the index.

More information: https://git-scm.com/docs/git-add.

- Add a file to the index:

```
git add {{path/to/file}}
```

- Add all files (tracked and untracked):

```
git add {{[-A|--all]}}
```

- Add all files recursively starting from the current folder:

```
git add .
```

- Only add already tracked files:

```
git add {{[-u|--update]}}
```

- Also add ignored files:

```
git add {{[-f|--force]}}
```

- Interactively stage parts of files:

```
git add {{[-p|--patch]}}
```

- Interactively stage parts of a given file:

```
git add {{[-p|--patch]}} {{path/to/file}}
```

- Interactively stage a file:

```
git add {{[-i|--interactive]}}
```

git alias

Create shortcuts for Git commands.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-alias.

- List all aliases:

```
git alias
```

- Create a new alias:

```
git alias "{{name}}" "{{command}}"
```

- Search for an existing alias:

```
git alias ^{{name}}
```

git am

Apply patch files and create a commit. Useful when receiving commits via email.

See also: **`git format-patch`** which can generate patch files.

More information: https://git-scm.com/docs/git-am.

- Apply and commit changes following a local patch file:

```
git am {{path/to/file.patch}}
```

- Apply and commit changes following a remote patch file:

```
curl {{[-L|--location]}} {{https://example.com/file.patch}} |
git apply
```

- Abort the process of applying a patch file:

```
git am --abort
```

- Apply as much of a patch file as possible, saving failed hunks to reject files:

```
git am --reject {{path/to/file.patch}}
```

git annex

Manage files with Git, without checking their contents in.

When a file is annexed, its content is moved into a key-value store, and a symlink is made that points to the content.

More information: https://git-annex.branchable.com/git-annex/.

- Initialize a repo with Git annex:

```
git annex init
```

- Add a file:

```
git annex add {{path/to/file_or_directory}}
```

- Show the current status of a file or directory:

```
git annex status {{path/to/file_or_directory}}
```

- Synchronize a local repository with a remote:

```
git annex {{remote}}
```

- Get a file or directory:

```
git annex get {{path/to/file_or_directory}}
```

- Display help:

```
git annex help
```

git annotate

Show commit hash and last author on each line of a file.

See **`git blame`**, which is preferred over **`git annotate`**.

`git annotate` is provided for those familiar with other version control systems.

More information: https://git-scm.com/docs/git-annotate.

- Print a file with the author name and commit hash prepended to each line:

```
git annotate {{path/to/file}}
```

- Print a file with the author email and commit hash prepended to each line:

```
git annotate {{[-e|--show-email]}} {{path/to/file}}
```

- Print only rows that match a `regex`:

```
git annotate -L :{{regexp}} {{path/to/file}}
```

git apply

Apply a patch to files and/or to the index without creating a commit.

See also: **`git am`** which applies a patch and also creates a commit.

More information: https://git-scm.com/docs/git-apply.

- Print messages about the patched files:

```
git apply {{[-v|--verbose]}} {{path/to/file}}
```

- Apply and add the patched files to the index:

```
git apply --index {{path/to/file}}
```

- Apply a remote patch file:

```
curl {{[-L|--location]}} {{https://example.com/file.patch}} |
git apply
```

- Output diffstat for the input and apply the patch:

```
git apply --stat --apply {{path/to/file}}
```

- Apply the patch in reverse:

```
git apply {{[-R|--reverse]}} {{path/to/file}}
```

- Store the patch result in the index without modifying the working tree:

```
git apply --cache {{path/to/file}}
```

git archive-file

Export all the files of the current Git branch into a Zip archive.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-archive-file.

- Pack the currently checked out commit into a Zip archive:

```
git archive-file
```

git archive

Create an archive of files from a tree.

More information: https://git-scm.com/docs/git-archive.

- Create a tar archive from the contents of the current HEAD and print it to `stdout`:

```
git archive {{[-v|--verbose]}} HEAD
```

- Use the Zip format and report progress verbosely:

```
git archive {{[-v|--verbose]}} --format zip HEAD
```

- Output the Zip archive to a specific file:

```
git archive {{[-v|--verbose]}} {{[-o|--output]}} {{path/to/file.zip}} HEAD
```

- Create a tar archive from the contents of the latest commit of a specific branch:

```
git archive {{[-o|--output]}} {{path/to/file.tar}} {{branch_name}}
```

- Use the contents of a specific directory:

```
git archive {{[-o|--output]}} {{path/to/file.tar}} HEAD:{{path/to/directory}}
```

- Prepend a path to each file to archive it inside a specific directory:

```
git archive {{[-o|--output]}} {{path/to/file.tar}} --prefix {{path/to/prepend}}/ HEAD
```

git authors

Generate a list of committers of a Git repository.

Part of **`git-extras`**.

More information: https://manned.org/git-authors.

- Print a full list of committers to `stdout` instead of to the `AUTHORS` file:

  `git authors {{[-l|--list]}}`

- Append the list of committers to the `AUTHORS` file and open it in the default editor:

  `git authors`

- Append the list of committers, excluding emails, to the `AUTHORS` file and open it in the default editor:

  `git authors --no-email`

git bisect

Use binary search to find the commit that introduced a bug.

Git automatically jumps back and forth in the commit graph to progressively narrow down the faulty commit.

More information: https://git-scm.com/docs/git-bisect.

- Start a bisect session on a commit range bounded by a known buggy commit, and a known clean (typically older) one:

```
git bisect start {{bad_commit}} {{good_commit}}
```

- For each commit that `git bisect` selects, mark it as "bad" or "good" after testing it for the issue:

```
git bisect {{good|bad}}
```

- End the bisect session and return to the previous branch:

```
git bisect reset
```

- Skip a commit during a bisect (e.g. one that fails the tests due to a different issue):

```
git bisect skip
```

- Display a log of what has been done so far:

```
git bisect log
```

git blame-someone-else

Blame someone else for your bad code.

More information: https://github.com/jayphelps/git-blame-someone-else.

- Change the committer and author of a commit:

```
git blame-someone-else "{{author
<someone@example.com>}}" {{commit}}
```

git blame

Show commit hash and last author on each line of a file.

More information: https://git-scm.com/docs/git-blame.

- Print file with author name and commit hash on each line:

```
git blame {{path/to/file}}
```

- Print file with author email and commit hash on each line:

```
git blame {{[-e|--show-email]}} {{path/to/file}}
```

- Print file with author name and commit hash on each line at a specific commit:

```
git blame {{commit}} {{path/to/file}}
```

- Print file with author name and commit hash on each line before a specific commit:

```
git blame {{commit}}~ {{path/to/file}}
```

- Jump to the parent of a specific commit and track a specific text and 10 of its following lines:

```
git blame -L '/{{text}}/',+10 {{a82812aa}}^ {{path/to/file}}
```

- Print author name and commit hash information for a specific line range:

```
git blame -L {{start_line}},{{end_line}} {{path/to/file}}
```

- Ignore whitespaces and line moves:

```
git blame -w -C -C -C {{path/to/file}}
```

git branch

Main Git command for working with branches.

More information: https://git-scm.com/docs/git-branch.

- List all branches (local and remote; the current branch is highlighted by *):

```
git branch {{[-a|--all]}}
```

- List which branches include a specific Git commit in their history:

```
git branch {{[-a|--all]}} --contains {{commit_hash}}
```

- Show the name of the current branch:

```
git branch --show-current
```

- Create new branch based on the current commit:

```
git branch {{branch_name}}
```

- Create new branch based on a specific commit:

```
git branch {{branch_name}} {{commit_hash}}
```

- Rename a branch (you must switch to a different branch before doing this):

```
git branch {{[-m|--move]}} {{old_branch_name}}
{{new_branch_name}}
```

- Delete a local branch (you must switch to a different branch before doing this):

```
git branch {{[-d|--delete]}} {{branch_name}}
```

- Delete a remote branch:

```
git push {{remote_name}} {{[-d|--delete]}}
{{remote_branch_name}}
```

git browse-ci

Open the current Git repository's CI website in the default web browser.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-browse-ci.

- Open the current repository's CI configuration on its upstream website:

  ```
  git browse-ci
  ```

- Open the current repository's CI configuration on its upstream website for a specific remote:

  ```
  git browse-ci {{remote}}
  ```

git browse

View an upstream repository in the default browser.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-browse.

- Open the first upstream in the default browser:

```
git browse
```

- Open a specific upstream in the default browser:

```
git browse {{upstream}}
```

git brv

Print a list of branches, sorted by last commit date.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-brv.

- List each branch showing date, latest commit hash and message:

```
git brv
```

git bug

A distributed bug tracker that uses Git's internal storage, so no files are added in your project.

You may submit your problems to the same Git remote you use to interact with others, much like commits and branches.

More information: https://github.com/MichaelMure/git-bug/blob/master/doc/md/git-bug.md.

- Create a new identity:

```
git bug user create
```

- Create a new bug:

```
git bug add
```

- Push a new bug entry to a remote:

```
git bug push
```

- Pull for updates:

```
git bug pull
```

- List existing bugs:

```
git bug ls
```

- Filter and sort bugs using a query:

```
git bug ls "{{status}}:{{open}} {{sort}}:{{edit}}"
```

- Search for bugs by text content:

```
git bug ls "{{search_query}}" baz
```

git bugreport

Captures debug information from the system and user, generating a text file to aid in the reporting of a bug in Git.

More information: https://git-scm.com/docs/git-bugreport.

- Create a new bug report file in the current directory:

```
git bugreport
```

- Create a new bug report file in the specified directory, creating it if it does not exist:

```
git bugreport {{[-o|--output-directory]}} {{path/to/directory}}
```

- Create a new bug report file with the specified filename suffix in `strftime` format:

```
git bugreport {{[-s|--suffix]}} {{%m%d%y}}
```

git bulk

Execute operations on multiple Git repositories.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-bulk.

- Register the current directory as a workspace:

```
git bulk --addcurrent {{workspace_name}}
```

- Register a workspace for bulk operations:

```
git bulk --addworkspace {{workspace_name}} /{{path/to/repository}}
```

- Clone a repository inside a specific directory, then register the repository as a workspace:

```
git bulk --addworkspace {{workspace_name}} /{{path/to/parent_directory}} --from {{remote_repository_location}}
```

- Clone repositories from a newline-separated list of remote locations, then register them as workspaces:

```
git bulk --addworkspace {{workspace_name}} /{{path/to/root_directory}} --from /{{path/to/file}}
```

- List all registered workspaces:

```
git bulk --listall
```

- Run a Git command on the repositories of the current workspace:

```
git bulk {{command}} {{command_arguments}}
```

- Remove a specific workspace:

```
git bulk --removeworkspace {{workspace_name}}
```

- Remove all workspaces:

```
git bulk --purge
```

git bundle

Package objects and references into an archive.

More information: https://git-scm.com/docs/git-bundle.

- Create a bundle file that contains all objects and references of a specific branch:

```
git bundle create {{path/to/file.bundle}} {{branch_name}}
```

- Create a bundle file of all branches:

```
git bundle create {{path/to/file.bundle}} --all
```

- Create a bundle file of the last 5 commits of the current branch:

```
git bundle create {{path/to/file.bundle}} -5 {{HEAD}}
```

- Create a bundle file of the latest 7 days:

```
git bundle create {{path/to/file.bundle}} --since 7.days {{HEAD}}
```

- Verify that a bundle file is valid and can be applied to the current repository:

```
git bundle verify {{path/to/file.bundle}}
```

- Print to `stdout` the list of references contained in a bundle:

```
git bundle unbundle {{path/to/file.bundle}}
```

- Unbundle a specific branch from a bundle file into the current repository:

```
git pull {{path/to/file.bundle}} {{branch_name}}
```

- Create a new repository from a bundle:

```
git clone {{path/to/file.bundle}}
```

git cat-file

Provide content or type and size information for Git repository objects.

More information: https://git-scm.com/docs/git-cat-file.

- Get the [s]ize of the HEAD commit in bytes:

```
git cat-file -s HEAD
```

- Get the [t]ype (blob, tree, commit, tag) of a given Git object:

```
git cat-file -t {{8c442dc3}}
```

- Pretty-[p]rint the contents of a given Git object based on its type:

```
git cat-file -p {{HEAD~2}}
```

git changelog

Generate a changelog report from repository commits and tags.

Part of **`git-extras`**.

More information: https://manned.org/git-changelog.

- Update existing file or create a new `History.md` file with the commit messages since the latest Git tag:

```
git changelog
```

- List commits from the current version:

```
git changelog {{[-l|--list]}}
```

- List a range of commits from the tag named `2.1.0` to now:

```
git changelog {{[-l|--list]}} {{[-s|--start-tag]}} {{2.1.0}}
```

- List pretty formatted range of commits between the tag `0.5.0` and the tag `1.0.0`:

```
git changelog {{[-s|--start-tag]}} {{0.5.0}} {{[-f|--final-tag]}} {{1.0.0}}
```

- List pretty formatted range of commits between the commit `0b97430` and the tag `1.0.0`:

```
git changelog --start-commit {{0b97430}} {{[-f|--final-tag]}} {{1.0.0}}
```

- Specify `CHANGELOG.md` as the output file:

```
git changelog {{CHANGELOG.md}}
```

- Replace contents of current changelog file entirely:

```
git changelog {{[-p|--prune-old]}}
```

git check-attr

For every pathname, list if each attribute is unspecified, set, or unset as a gitattribute on that pathname.

More information: https://git-scm.com/docs/git-check-attr.

- Check the values of all attributes on a file:

```
git check-attr {{[-a|--all]}} {{path/to/file}}
```

- Check the value of a specific attribute on a file:

```
git check-attr {{attribute}} {{path/to/file}}
```

- Check the values of all attributes on specific files:

```
git check-attr {{[-a|--all]}} {{path/to/file1 path/to/
file2 ...}}
```

- Check the value of a specific attribute on one or more files:

```
git check-attr {{attribute}} {{path/to/file1 path/to/
file2 ...}}
```

git check-ignore

Analyze and debug Git ignore/exclude (".gitignore") files.

More information: https://git-scm.com/docs/git-check-ignore.

- Check whether a file or directory is ignored:

```
git check-ignore {{path/to/file_or_directory}}
```

- Check whether multiple files or directories are ignored:

```
git check-ignore {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Use pathnames, one per line, from `stdin`:

```
git < {{path/to/file_list}} check-ignore --stdin
```

- Do not check the index (used to debug why paths were tracked and not ignored):

```
git check-ignore --no-index {{path/to/file_or_directory1
path/to/file_or_directory2 ...}}
```

- Include details about the matching pattern for each path:

```
git check-ignore {{[-v|--verbose]}} {{path/to/
file_or_directory1 path/to/file_or_directory2 ...}}
```

git check-mailmap

Show canonical names and email addresses of contacts.

More information: https://git-scm.com/docs/git-check-mailmap.

- Look up the canonical name associated with an email address:

```
git check-mailmap "<{{email@example.com}}>"
```

git check-ref-format

Check if a reference name is acceptable, and exit with a non-zero status if it is not.

More information: https://git-scm.com/docs/git-check-ref-format.

- Check the format of the specified reference name:

```
git check-ref-format {{refs/head/refname}}
```

- Print the name of the last branch checked out:

```
git check-ref-format --branch @{-1}
```

- Normalize a refname:

```
git check-ref-format --normalize {{refs/head/refname}}
```

git checkout-index

Copy files from the index to the working tree.

More information: https://git-scm.com/docs/git-checkout-index.

- Restore any files deleted since the last commit:

```
git checkout-index {{[-a|--all]}}
```

- Restore any files deleted or changed since the last commit:

```
git checkout-index {{[-a|--all]}} {{[-f|--force]}}
```

- Restore any files changed since the last commit, ignoring any files that were deleted:

```
git checkout-index {{[-a|--all]}} {{[-f|--force]}} {{[-n|--no-create]}}
```

- Export a copy of the entire tree at the last commit to the specified directory (the trailing slash is important):

```
git checkout-index {{[-a|--all]}} {{[-f|--force]}} --prefix {{path/to/export_directory}}/
```

git checkout

Checkout a branch or paths to the working tree.

More information: https://git-scm.com/docs/git-checkout.

- Create and switch to a new branch:

```
git checkout -b {{branch_name}}
```

- Create and switch to a new branch based on a specific reference (branch, remote/branch, tag are examples of valid references):

```
git checkout -b {{branch_name}} {{reference}}
```

- Switch to an existing local branch:

```
git checkout {{branch_name}}
```

- Switch to the previously checked out branch:

```
git checkout -
```

- Switch to an existing remote branch:

```
git checkout {{[-t|--track]}} {{remote_name}}/{{branch_name}}
```

- Discard all unstaged changes in the current directory (see `git reset` for more undo-like commands):

```
git checkout .
```

- Discard unstaged changes to a given file:

```
git checkout {{path/to/file}}
```

- Replace a file in the current directory with the version of it committed in a given branch:

```
git checkout {{branch_name}} -- {{path/to/file}}
```

git cherry-pick

Apply the changes introduced by existing commits to the current branch.

To apply changes to another branch, first use **`git checkout`** to switch to the desired branch.

More information: https://git-scm.com/docs/git-cherry-pick.

- Apply a commit to the current branch:

  ```
  git cherry-pick {{commit}}
  ```

- Apply a range of commits to the current branch (see also: `git rebase --onto`):

  ```
  git cherry-pick {{start_commit}}~..{{end_commit}}
  ```

- Apply multiple (non-sequential) commits to the current branch:

  ```
  git cherry-pick {{commit1 commit2 ...}}
  ```

- Add the changes of a commit to the working directory, without creating a commit:

  ```
  git cherry-pick {{[-n|--no-commit]}} {{commit}}
  ```

git cherry

Find commits that have yet to be applied upstream.

More information: https://git-scm.com/docs/git-cherry.

- Show commits (and their messages) with equivalent commits upstream:

```
git cherry {{[-v|--verbose]}}
```

- Specify a different upstream and topic branch:

```
git cherry {{origin}} {{topic}}
```

- Limit commits to those within a given limit:

```
git cherry {{origin}} {{topic}} {{base}}
```

git clean

Remove files not tracked by Git from the working tree.

More information: https://git-scm.com/docs/git-clean.

- Delete untracked files:

```
git clean
```

- Interactively delete untracked files:

```
git clean {{[-i|--interactive]}}
```

- Show which files would be deleted without actually deleting them:

```
git clean {{[-n|--dry-run]}}
```

- Forcefully delete untracked files:

```
git clean {{[-f|--force]}}
```

- Forcefully delete untracked [d]irectories:

```
git clean {{[-f|--force]}} -d
```

- Delete untracked files, including e[x]cluded files (files ignored in `.gitignore` and `.git/info/exclude`):

```
git clean -x
```

git clear-soft

Clear a Git working directory as if it was freshly cloned with the current branch excluding files in **`.gitignore`**.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-clear-soft.

- Reset all tracked files and delete all untracked files:

```
git clear-soft
```

git clear

Clear a Git working directory as if it was freshly cloned with the current branch including files in **`.gitignore`**.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-clear.

- Reset all tracked files and delete all untracked files even if they are included in the `.gitignore`:

```
git clear
```

git cliff

A highly customizable changelog generator.

More information: https://git-cliff.org/docs/usage/args.

- Generate a changelog from all commits in a Git repository and save it to `CHANGELOG.md`:

```
git cliff > {{CHANGELOG.md}}
```

- Generate a changelog from commits starting from the latest tag and print it to `stdout`:

```
git cliff {{[-l|--latest]}}
```

- Generate a changelog from commits that belong to the current tag (use `git checkout` on a tag before this):

```
git cliff --current
```

- Generate a changelog from commits that do not belong to a tag:

```
git cliff {{[-u|--unreleased]}}
```

- Write the default config file to `cliff.toml` in the current directory:

```
git cliff {{[-i|--init]}}
```

git clone

Clone an existing repository.

More information: https://git-scm.com/docs/git-clone.

- Clone an existing repository into a new directory (the default directory is the repository name):

```
git clone {{remote_repository_location}} {{path/to/directory}}
```

- Clone an existing repository and its submodules:

```
git clone --recursive {{remote_repository_location}}
```

- Clone only the `.git` directory of an existing repository:

```
git clone {{[-n|--no-checkout]}} {{remote_repository_location}}
```

- Clone a local repository:

```
git clone {{[-l|--local]}} {{path/to/local_repository}}
```

- Clone quietly:

```
git clone {{[-q|--quiet]}} {{remote_repository_location}}
```

- Clone an existing repository only fetching the 10 most recent commits on the default branch (useful to save time):

```
git clone --depth 10 {{remote_repository_location}}
```

- Clone an existing repository only fetching a specific branch:

```
git clone {{[-b|--branch]}} {{name}} --single-branch {{remote_repository_location}}
```

- Clone an existing repository using a specific SSH command:

```
git clone {{[-c|--config]}} core.sshCommand="{{ssh -i path/to/private_ssh_key}}" {{remote_repository_location}}
```

git coauthor

Add another author to the latest commit. Since this command rewrites the Git history, **`--force`** will be needed when pushing next time.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-coauthor.

- Insert an additional author to the last Git commit:

```
git coauthor {{name}} {{name@example.com}}
```

git cola

A powerful Git GUI with a slick and intuitive user interface.

More information: https://git-cola.readthedocs.io.

- Start the GUI:

```
git cola
```

- Start the GUI in amend mode:

```
git cola --amend
```

- Prompt for a Git repository. Defaults to the current directory:

```
git cola --prompt
```

- Open the Git repository at mentioned path:

```
git cola {{[-r|--repo]}} {{path/to/git-repository}}
```

- Apply the path filter to the status widget:

```
git cola {{[-s|--status-filter]}} {{filter}}
```

git column

Display data in columns.

More information: https://git-scm.com/docs/git-column.

- Format `stdin` as multiple columns:

```
ls | git column --mode={{column}}
```

- Format `stdin` as multiple columns with a maximum width of `100`:

```
ls | git column --mode=column --width={{100}}
```

- Format `stdin` as multiple columns with a maximum padding of `30`:

```
ls | git column --mode=column --padding={{30}}
```

git commit-graph

Write and verify Git commit-graph files.

More information: https://git-scm.com/docs/git-commit-graph.

- Write a commit-graph file for the packed commits in the repository's local `.git` directory:

```
git commit-graph write
```

- Write a commit-graph file containing all reachable commits:

```
git show-ref {{[-s|--hash]}} | git commit-graph write --stdin-commits
```

- Write a commit-graph file containing all commits in the current commit-graph file along with those reachable from `HEAD`:

```
git rev-parse {{HEAD}} | git commit-graph write --stdin-commits --append
```

git commit-tree

Low level utility to create commit objects.

See also: **`git commit`**.

More information: https://git-scm.com/docs/git-commit-tree.

- Create a commit object with the specified message:

```
git commit-tree {{tree}} -m "{{message}}"
```

- Create a commit object reading the message from a file (use `-` for `stdin`):

```
git commit-tree {{tree}} -F {{path/to/file}}
```

- Create a GPG-signed commit object:

```
git commit-tree {{tree}} -m "{{message}}" {{[-S|--gpg-sign]}}
```

- Create a commit object with the specified parent commit object:

```
git commit-tree {{tree}} -m "{{message}}" -p
{{parent_commit_sha}}
```

git commit

Commit files to the repository.

More information: https://git-scm.com/docs/git-commit.

- Commit staged files to the repository with a message:

```
git commit {{[-m|--message]}} "{{message}}"
```

- Commit staged files with a message read from a file:

```
git commit {{[-F|--file]}} {{path/to/commit_message_file}}
```

- Auto stage all modified and deleted files and commit with a message:

```
git commit {{[-a|--all]}} {{[-m|--message]}} "{{message}}"
```

- Commit staged files and sign them with the specified GPG key (or the one defined in the configuration file if no argument is specified):

```
git commit {{[-S|--gpg-sign]}} {{key_id}} {{[-m|--message]}} "{{message}}"
```

- Update the last commit by adding the currently staged changes, changing the commit's hash:

```
git commit --amend
```

- Commit only specific (already staged) files:

```
git commit {{path/to/file1 path/to/file2 ...}}
```

- Create a commit, even if there are no staged files:

```
git commit {{[-m|--message]}} "{{message}}" --allow-empty
```

git commits-since

Display commits since a time or date.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-commits-since.

- Display commits since yesterday:

```
git commits-since {{yesterday}}
```

- Display commits since last week:

```
git commits-since {{last week}}
```

- Display commits since last month:

```
git commits-since {{last month}}
```

- Display commits since yesterday 2pm:

```
git commits-since {{yesterday 2pm}}
```

git config

Manage custom configuration options for Git repositories.

These configurations can be local (for the current repository) or global (for the current user).

More information: https://git-scm.com/docs/git-config.

- Globally set your name or email (this information is required to commit to a repository and will be included in all commits):

```
git config --global {{user.name|user.email}} "{{Your Name|email@example.com}}"
```

- List local, global or system configuration entries and show their file location:

```
git config {{[-l|--list]}} --{{local|global|system}} --show-origin
```

- Set the global value of a given configuration entry (in this case an alias):

```
git config --global {{alias.unstage}} "reset HEAD --"
```

- Get the value of a given configuration entry:

```
git config {{alias.unstage}}
```

- Use an alias:

```
git {{unstage}}
```

- Revert a global configuration entry to its default value:

```
git config --global --unset {{alias.unstage}}
```

- Edit the local Git configuration (`.git/config`) in the default editor:

```
git config {{[-e|--edit]}}
```

- Edit the global Git configuration (`~/.gitconfig` by default or `$XDG_CONFIG_HOME/git/config` if such a file exists) in the default editor:

```
git config --global {{[-e|--edit]}}
```

git contrib

Display commits from an author.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-contrib.

- Display all commit hashes and their corresponding commit messages from a specific author:

```
git contrib {{author}}
```

git count-objects

Count the number of unpacked objects and their disk consumption.

More information: https://git-scm.com/docs/git-count-objects.

- Count all objects and display the total disk usage:

```
git count-objects
```

- Display a count of all objects and their total disk usage, displaying sizes in human-readable units:

```
git count-objects {{[-H|--human-readable]}}
```

- Display more verbose information:

```
git count-objects {{[-v|--verbose]}}
```

- Display more verbose information, displaying sizes in human-readable units:

```
git count-objects {{[-H|--human-readable]}} {{[-v|--verbose]}}
```

git count

Print the total number of commits.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-count.

- Print the total number of commits:

```
git count
```

- Print the number of commits per contributor and the total number of commits:

```
git count --all
```

git cp

Copy an existing file to a new location, preserving history.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-cp.

- Copy an existing file in a Git repo, staying in the same directory:

```
git cp {{file}} {{new_file}}
```

- Copy an existing file in a Git repo and place it elsewhere:

```
git cp {{path/to/file}} {{path/to/new_file}}
```

git create-branch

Create a Git branch in a repository.

Part of **`git-extras`**.

More information: https://manned.org/git-create-branch.

- Create a local branch:

```
git create-branch {{branch_name}}
```

- Create a branch locally and on origin:

```
git create-branch {{[-r|--remote]}} {{branch_name}}
```

- Create a branch locally and on upstream (through forks):

```
git create-branch {{[-r|--remote]}} upstream {{branch_name}}
```

git credential-cache

Git helper to temporarily store passwords in memory.

More information: https://git-scm.com/docs/git-credential-cache.

- Store Git credentials for a specific amount of time:

```
git config credential.helper 'cache --
timeout={{time_in_seconds}}'
```

git credential-store

Git helper to store passwords on disk.

More information: https://git-scm.com/docs/git-credential-store.

- Store Git credentials in a specific file:

```
git config credential.helper 'store --file={{path/to/file}}'
```

git credential

Retrieve and store user credentials.

More information: https://git-scm.com/docs/git-credential.

- Display credential information, retrieving the username and password from configuration files:

```
echo "{{url=http://example.com}}" | git credential fill
```

- Send credential information to all configured credential helpers to store for later use:

```
echo "{{url=http://example.com}}" | git credential approve
```

- Erase the specified credential information from all the configured credential helpers:

```
echo "{{url=http://example.com}}" | git credential reject
```

git cvsexportcommit

Export a single **Git** commit to a CVS checkout.

More information: https://git-scm.com/docs/git-cvsexportcommit.

- Merge a specific patch into CVS:

```
git cvsexportcommit -v -c -w {{path/to/project_cvs_checkout}}
{{commit_sha1}}
```

git daemon

A really simple server for Git repositories.

More information: https://git-scm.com/docs/git-daemon.

- Launch a Git daemon with a whitelisted set of directories:

```
git daemon --export-all {{path/to/directory1 path/to/directory2 ...}}
```

- Launch a Git daemon with a specific base directory and allow pulling from all sub-directories that look like Git repositories:

```
git daemon --base-path={{path/to/directory}} --export-all --reuseaddr
```

- Launch a Git daemon for the specified directory, verbosely printing log messages and allowing Git clients to write to it:

```
git daemon {{path/to/directory}} --enable=receive-pack --informative-errors --verbose
```

git delete-branch

Delete local and remote Git branches.

Part of **`git-extras`**. If deleting the checked out branch, only the remote branch will be deleted.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-delete-branch.

- Delete one or more local and remote Git branches:

```
git delete-branch {{branch_name1 branch_name2 ...}}
```

git delete-merged-branches

Delete branches that are listed in **`git branch --merged`** excluding master.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-delete-merged-branches.

- Delete merged branches:

```
git delete-merged-branches
```

git delete-squashed-branches

Delete branches that have been "squashed-merged" into a specified branch and checkout. If no branch is specified, default to the currently checked out branch.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-delete-squashed-branches.

- Delete all branches that were "squash-merged" into the current checked out branch:

```
git delete-squashed-branches
```

- Delete all branches that were "squash-merged" into a specific branch:

```
git delete-squashed-branches {{branch_name}}
```

git delete-submodule

Delete a submodule from a Git repository.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-delete-submodule.

- Delete a specific submodule:

```
git delete-submodule {{path/to/submodule}}
```

git delete-tag

Delete existing local and remote tags.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-delete-tag.

- Delete a tag:

```
git delete-tag {{tag_version}}
```

git delta

List files that differ from another branch.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-delta.

- List files from the current checked out branch that differ from the `main` branch:

```
git delta {{main}}
```

- List files from a specific branch that differ from another specific branch:

```
git delta {{branch_1}} {{branch_2}}
```

git describe

Give an object a human-readable name based on an available ref.

More information: https://git-scm.com/docs/git-describe.

- Create a unique name for the current commit (the name contains the most recent annotated tag, the number of additional commits, and the abbreviated commit hash):

```
git describe
```

- Create a name with 4 digits for the abbreviated commit hash:

```
git describe --abbrev={{4}}
```

- Generate a name with the tag reference path:

```
git describe --all
```

- Describe a Git tag:

```
git describe {{v1.0.0}}
```

- Create a name for the last commit of a given branch:

```
git describe {{branch_name}}
```

git diff-files

Compare files using their sha1 hashes and modes.

More information: https://git-scm.com/docs/git-diff-files.

- Compare all changed files:

```
git diff-files
```

- Compare only specified files:

```
git diff-files {{path/to/file}}
```

- Show only the names of changed files:

```
git diff-files --name-only
```

- Output a summary of extended header information:

```
git diff-files --summary
```

git diff-index

Compare the working directory with a commit or tree object.

More information: https://git-scm.com/docs/git-diff-index.

- Compare the working directory with a specific commit:

```
git diff-index {{commit}}
```

- Compare a specific file or directory in working directory with a commit:

```
git diff-index {{commit}} {{path/to/file_or_directory}}
```

- Compare the working directory with the index (staging area) to check for staged changes:

```
git diff-index --cached {{commit}}
```

- Suppress output and return an exit status to check for differences:

```
git diff-index --quiet {{commit}}
```

git diff-tree

Compares the content and mode of blobs found via two tree objects.

More information: https://git-scm.com/docs/git-diff-tree.

- Compare two tree objects:

```
git diff-tree {{tree-ish1}} {{tree-ish2}}
```

- Show changes between two specific commits:

```
git diff-tree -r {{commit1}} {{commit2}}
```

- Display changes in patch format:

```
git diff-tree {{[-p|--patch]}} {{tree-ish1}} {{tree-ish2}}
```

- Filter changes by a specific path:

```
git diff-tree {{tree-ish1}} {{tree-ish2}} -- {{path/to/
file_or_directory}}
```

git diff

Show changes to tracked files.

More information: https://git-scm.com/docs/git-diff.

- Show unstaged changes:

```
git diff
```

- Show all uncommitted changes (including staged ones):

```
git diff HEAD
```

- Show only staged (added, but not yet committed) changes:

```
git diff --staged
```

- Show changes from all commits since a given date/time (a date expression, e.g. "1 week 2 days" or an ISO date):

```
git diff 'HEAD@{{{3 months|weeks|days|hours|seconds ago}}}'
```

- Show diff statistics, like files changed, histogram, and total line insertions/deletions:

```
git diff --stat {{commit}}
```

- Output a summary of file creations, renames and mode changes since a given commit:

```
git diff --summary {{commit}}
```

- Compare a single file between two branches or commits:

```
git diff {{branch_1}}..{{branch_2}} {{path/to/file}}
```

- Compare different files from the current branch to another branch:

```
git diff {{other_branch}}:{{path/to/file2}} {{path/to/file1}}
```

git difftool

Show file changes using external diff tools. Accepts the same options and arguments as **`git diff`**.

See also: **`git diff`**.

More information: https://git-scm.com/docs/git-difftool.

- List available diff tools:

```
git difftool --tool-help
```

- Set the default diff tool to meld:

```
git config --global diff.tool "{{meld}}"
```

- Use the default diff tool to show staged changes:

```
git difftool --staged
```

- Use a specific tool (opendiff) to show changes since a given commit:

```
git difftool {{[-t|--tool]}} {{opendiff}} {{commit}}
```

git effort

Display how much activity a file has had, showing commits per file and "active days" i.e. total number of days that contributed to the file.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-effort.

- Display each file in the repository, showing commits and active days:

```
git effort
```

- Display files modified by a specific number of commits or more, showing commits and active days:

```
git effort --above {{5}}
```

- Display files modified by a specific author, showing commits and active days:

```
git effort -- --author="{{username}}"
```

- Display files modified since a specific time/date, showing commits and active days:

```
git effort -- --since="{{last month}}"
```

- Display only the specified files or directories, showing commits and active days:

```
git effort {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Display all files in a specific directory, showing commits and active days:

```
git effort {{path/to/directory/*}}
```

git extras

Git extension pack.

More information: https://manned.org/git-extras.

- Install or upgrade `git-extras` commands:

```
git extras update
```

- Display help:

```
git extras {{[-h|--help]}}
```

- Display version:

```
git extras {{[-v|--version]}}
```

git fame

Pretty-print Git repository contributions.

More information: https://manned.org/git-fame.

- Calculate contributions for the current Git repository:

```
git fame
```

- Exclude files/directories that match the specified `regex`:

```
git fame --excl "{{regex}}"
```

- Calculate contributions made after the specified date:

```
git fame --since "{{3 weeks ago|2021-05-13}}"
```

- Display contributions in the specified format:

```
git fame --format {{pipe|yaml|json|csv|tsv}}
```

- Display contributions per file extension:

```
git fame {{[-t|--bytype]}}
```

- Ignore whitespace changes:

```
git fame {{[-w|--ignore-whitespace]}}
```

- Detect inter-file line moves and copies:

```
git fame -C
```

- Detect intra-file line moves and copies:

```
git fame -M
```

git fast-export

Export the contents and history of a Git repository in a streamable, plain-text format.

More information: https://manned.org/git-fast-export.

- Export the entire Git repository history to `stdout`:

```
git fast-export --all
```

- Export the entire repository to a file:

```
git fast-export --all > {{path/to/file}}
```

- Export a specific branch only:

```
git fast-export {{main}}
```

- Export with `progress` statements every `n` objects (for showing progress during `git fast-import`):

```
git fast-export --progress {{n}} --all > {{path/to/file}}
```

- Export only a specific subdirectory's history:

```
git fast-export --all -- {{path/to/directory}} > {{path/to/file}}
```

git feature

Create or merge feature branches.

Feature branches obey the format feature/name.

More information: https://manned.org/git-feature.

- Create and switch to a new feature branch:

```
git feature {{feature_branch}}
```

- Merge a feature branch into the current branch creating a merge commit:

```
git feature finish {{feature_branch}}
```

- Merge a feature branch into the current branch squashing the changes into one commit:

```
git feature finish --squash {{feature_branch}}
```

- Send changes from a specific feature branch to its remote counterpart:

```
git feature {{feature_branch}} {{[-r|--remote]}} {{remote_name}}
```

git fetch

Download objects and refs from a remote repository.

More information: https://git-scm.com/docs/git-fetch.

- Fetch the latest changes from the default remote upstream repository (if set):

```
git fetch
```

- Fetch new branches from a specific remote upstream repository:

```
git fetch {{remote_name}}
```

- Fetch the latest changes from all remote upstream repositories:

```
git fetch --all
```

- Also fetch tags from the remote upstream repository:

```
git fetch {{[-t|--tags]}}
```

- Delete local references to remote branches that have been deleted upstream:

```
git fetch {{[-p|--prune]}}
```

- Deepen current shallow branch by 2 commits:

```
git fetch --deepen 2
```

- Update the `main` branch without switching to it (equivalent to `git pull`):

```
git fetch {{origin}} main:main
```

git filter-branch

Change branch history, like removing files.

More information: https://git-scm.com/docs/git-filter-branch.

- Remove a file from all commits:

```
git filter-branch --tree-filter 'rm {{[-f|--force]}}
{{file}}' HEAD
```

- Update author email:

```
git filter-branch --env-filter
'GIT_AUTHOR_EMAIL={{new_email}}' HEAD
```

- Delete a folder from history:

```
git filter-branch --tree-filter 'rm {{[-rf|--recursive --
force]}} {{folder}}' HEAD
```

git filter-repo

A versatile tool for rewriting Git history.

See also: **`bfg`**.

More information: https://github.com/newren/git-filter-repo.

- Replace a sensitive string in all files:

```
git filter-repo --replace-text <(echo '{{find}}
==>{{replacement}}')
```

- Extract a single folder, keeping history:

```
git filter-repo --path {{path/to/folder}}
```

- Remove a single folder, keeping history:

```
git filter-repo --path {{path/to/folder}} --invert-paths
```

- Move everything from sub-folder one level up:

```
git filter-repo --path-rename {{path/to/folder}}/:
```

git flow

A collection of Git extensions to provide high-level repository operations.

More information: https://github.com/nvie/gitflow.

- Initialize it inside an existing Git repository:

```
git flow init
```

- Start developing on a feature branch based on `develop`:

```
git flow feature start {{feature}}
```

- Finish development on a feature branch, merging it into the `develop` branch and deleting it:

```
git flow feature finish {{feature}}
```

- Publish a feature to the remote server:

```
git flow feature publish {{feature}}
```

- Get a feature published by another user:

```
git flow feature pull origin {{feature}}
```

git for-each-ref

List and optionally format references (branches, tags) in a Git repository.

More information: https://git-scm.com/docs/git-for-each-ref.

- List all refs (branches and tags):

```
git for-each-ref
```

- List only branches:

```
git for-each-ref refs/heads/
```

- List only tags:

```
git for-each-ref refs/tags/
```

- Show branches merged into HEAD:

```
git for-each-ref --merged HEAD refs/heads/
```

- List short names of all refs:

```
git for-each-ref --format "%(refname:short)"
```

- Sort refs by committer date (most recent first):

```
git for-each-ref --sort -committerdate
```

- Sort refs by committer date (oldest first):

```
git for-each-ref --sort committerdate
```

- Limit output to a specified number of refs:

```
git for-each-ref --count {{count}}
```

git for-each-repo

Run a Git command on a list of repositories.

Note: This command is experimental and may change.

More information: https://git-scm.com/docs/git-for-each-repo.

- Run maintenance on each of a list of repositories stored in the `maintenance.repo` user configuration variable:

  ```
  git for-each-repo --config maintenance.repo {{maintenance run}}
  ```

- Run `git pull` on each repository listed in a global configuration variable:

  ```
  git for-each-repo --config {{global_configuration_variable}} {{pull}}
  ```

git force-clone

Get the basic functionality of **`git clone`**, but if the destination Git repository already exists it will force-reset it to resemble a clone of the remote.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-force-clone.

- Clone a Git repository into a new directory:

```
git force-clone {{remote_repository_location}} {{path/to/directory}}
```

- Clone a Git repository into a new directory, checking out an specific branch:

```
git force-clone {{[-b|--branch]}} {{branch_name}} {{remote_repository_location}} {{path/to/directory}}
```

- Clone a Git repository into an existing directory of a Git repository, performing a force-reset to resemble it to the remote and checking out an specific branch:

```
git force-clone {{[-b|--branch]}} {{branch_name}} {{remote_repository_location}} {{path/to/directory}}
```

git fork

Fork a GitHub repo. Like **`git clone`** but forks first.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-fork.

- Fork and clone a GitHub repository by its URL:

```
git fork {{https://github.com/tldr-pages/tldr}}
```

- Fork and clone a GitHub repository by its slug:

```
git fork {{tldr-pages/tldr}}
```

git format-patch

Prepare .patch files. Useful when emailing commits elsewhere.

See also: **`git am`** which can apply generated .patch files.

More information: https://git-scm.com/docs/git-format-patch.

- Create an auto-named `.patch` file for all the unpushed commits:

```
git format-patch {{origin}}
```

- Write a `.patch` file for all the commits between 2 revisions to `stdout`:

```
git format-patch {{revision_1}}..{{revision_2}}
```

- Write a `.patch` file for the `n` latest commits:

```
git format-patch -{{n}}
```

git fresh-branch

Create an empty local branch.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md##git-fresh-branch.

- Create an empty local branch:

```
git fresh-branch {{branch_name}}
```

git fsck

Verify the validity and connectivity of nodes in a Git repository index.

Does not make any modifications.

See also: **`git gc`** for cleaning up dangling blobs.

More information: https://git-scm.com/docs/git-fsck.

- Check the current repository:

```
git fsck
```

- List all tags found:

```
git fsck --tags
```

- List all root nodes found:

```
git fsck --root
```

- Show all unreachable and dangling objects, ignore reflogs, and perform a full integrity check:

```
git fsck --dangling --no-reflogs --unreachable --full
```

- Check connectivity only (skip object integrity verification):

```
git fsck --connectivity-only
```

git gc

Optimise the local repository by cleaning unnecessary files.

More information: https://git-scm.com/docs/git-gc.

- Optimise the repository:

```
git gc
```

- Aggressively optimise, takes more time:

```
git gc --aggressive
```

- Do not prune loose objects (prunes by default):

```
git gc --no-prune
```

- Suppress all output:

```
git gc --quiet
```

- Display help:

```
git gc --help
```

git get-tar-commit-id

Extract commit ID from an archive created using **`git archive`**.

More information: https://git-scm.com/docs/git-get-tar-commit-id.

- Extract commit hash ID or quietly exit with a return code of 1:

```
git < {{path/to/archive.tar}} get-tar-commit-id
```

git gh-pages

Create a new branch inside the current repository called **gh-pages**.

Part of **git-extras**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-gh-pages.

- Create the GitHub pages branch inside the repository in the current directory:

```
git gh-pages
```

git graft

Merge commits from a branch into another branch and delete the source branch.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-graft.

- Merge all commits not present on the target branch from the source branch to target branch, and delete the source branch:

```
git graft {{source_branch}} {{target_branch}}
```

git grep

Find strings inside files anywhere in a repository's history.

Accepts a lot of the same flags as regular **`grep`**.

More information: https://git-scm.com/docs/git-grep.

- Search for a string in tracked files:

```
git grep {{search_string}}
```

- Search for a string in files matching a pattern in tracked files:

```
git grep {{search_string}} -- {{file_glob_pattern}}
```

- Search for a string in tracked files, including submodules:

```
git grep --recurse-submodules {{search_string}}
```

- Search for a string at a specific point in history:

```
git grep {{search_string}} {{HEAD~2}}
```

- Search for a string across all branches:

```
git grep {{search_string}} $(git rev-list --all)
```

git gui

A GUI for Git to manage branches, commits, and remotes, and perform local merges.

See also: **`git-cola`**, **`gitk`**.

More information: https://git-scm.com/docs/git-gui.

- Launch the GUI:

```
git gui
```

- Show a specific file with author name and commit hash on each line:

```
git gui blame {{path/to/file}}
```

- Open `git gui blame` in a specific revision:

```
git gui blame {{revision}} {{path/to/file}}
```

- Open `git gui blame` and scroll the view to center on a specific line:

```
git gui blame --line={{line}} {{path/to/file}}
```

- Open a window to make one commit and return to the shell when it is complete:

```
git gui citool
```

- Open `git gui citool` in the "Amend Last Commit" mode:

```
git gui citool --amend
```

- Open `git gui citool` in a read-only mode:

```
git gui citool --nocommit
```

- Show a browser for the tree of a specific branch, opening the blame tool when clicking on the files:

```
git gui browser maint
```

git guilt

Show total blame count for files with unstaged changes or calculate the change in blame between two revisions.

Part of **`git-extras`**.

More information: https://manned.org/git-guilt.

- Show total blame count:

```
git guilt
```

- Calculate the change in blame between two revisions:

```
git guilt {{first_revision}} {{last_revision}}
```

- Show author emails instead of names:

```
git guilt {{[-e|--email]}}
```

- Ignore whitespace only changes when attributing blame:

```
git guilt {{[-w|--ignore-whitespace]}}
```

- Find blame delta over the last three weeks:

```
git guilt 'git log --until "3 weeks ago" --format "%H" {{[-n|--max-count]}} 1'
```

- Find blame delta over the last three weeks (git 1.8.5+):

```
git guilt @{3.weeks.ago}
```

git hash-object

Computes the unique hash key of content and optionally creates an object with specified type.

More information: https://git-scm.com/docs/git-hash-object.

- Compute the object ID without storing it:

```
git hash-object {{path/to/file}}
```

- Compute the object ID and store it in the Git database:

```
git hash-object -w {{path/to/file}}
```

- Compute the object ID specifying the object type:

```
git hash-object -t {{blob|commit|tag|tree}} {{path/to/file}}
```

- Compute the object ID from `stdin`:

```
cat {{path/to/file}} | git hash-object --stdin
```

git help

Display help information about Git.

More information: https://git-scm.com/docs/git-help.

- Display help about a specific Git subcommand:

```
git help {{subcommand}}
```

- Display help about a specific Git subcommand in a web browser:

```
git help {{[-w|--web]}} {{subcommand}}
```

- Display a list of all available Git subcommands:

```
git help {{[-a|--all]}}
```

- List the available guides:

```
git help {{[-g|--guides]}}
```

- List all possible configuration variables:

```
git help {{[-c|--config]}}
```

git ignore-io

Generate .gitignore files from predefined templates.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-ignore-io.

- List available templates:

```
git ignore-io list
```

- Generate a .gitignore template:

```
git ignore-io {{item_a,item_b,...}}
```

git ignore

Show/update **`.gitignore`** files.

Part of **`git-extras`**.

See also: **`git ignore-io`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-ignore.

- Show the content of all global and local `.gitignore` files:

```
git ignore
```

- Ignore file(s) privately, updating `.git/info/exclude` file:

```
git ignore {{file_pattern}} {{[-p|--private]}}
```

- Ignore file(s) locally, updating local `.gitignore` file:

```
git ignore {{file_pattern}}
```

- Ignore file(s) globally, updating global `.gitignore` file:

```
git ignore {{file_pattern}} {{[-g|--global]}}
```

git imerge

Perform a merge or rebase between two Git branches incrementally.

Conflicts between branches are tracked down to pairs of individual commits, to simplify conflict resolution.

More information: https://github.com/mhagger/git-imerge.

- Start imerge-based rebase (checkout the branch to be rebased, first):

```
git imerge rebase {{branch_to_rebase_onto}}
```

- Start imerge-based merge (checkout the branch to merge into, first):

```
git imerge merge {{branch_to_be_merged}}
```

- Show ASCII diagram of in-progress merge or rebase:

```
git imerge diagram
```

- Continue imerge operation after resolving conflicts (`git add` the conflicted files, first):

```
git imerge continue --no-edit
```

- Wrap up imerge operation, after all conflicts are resolved:

```
git imerge finish
```

- Abort imerge operation, and return to the previous branch:

```
git imerge remove && git checkout {{previous_branch}}
```

git info

Display Git repository information.

Part of **`git-extras`**.

More information: https://manned.org/git-info.

- Display remote locations, remote and local branches, most recent commit data and `.git/config` settings:

```
git info
```

- Display remote locations, remote and local branches and most recent commit data:

```
git info --no-config
```

git init

Initialize a new local Git repository.

More information: https://git-scm.com/docs/git-init.

- Initialize a new local repository:

```
git init
```

- Initialize a repository with the specified name for the initial branch:

```
git init {{[-b|--initial-branch]}} {{branch_name}}
```

- Initialize a repository using SHA256 for object hashes (requires Git version 2.29+):

```
git init --object-format sha256
```

- Initialize a barebones repository, suitable for use as a remote over SSH:

```
git init --bare
```

git instaweb

Helper to launch a GitWeb server.

More information: https://git-scm.com/docs/git-instaweb.

- Launch a GitWeb server for the current Git repository:

```
git instaweb --start
```

- Listen only on localhost:

```
git instaweb --start {{[-l|--local]}}
```

- Listen on a specific port:

```
git instaweb --start {{[-p|--port]}} {{1234}}
```

- Use a specified HTTP daemon:

```
git instaweb --start {{[-d|--httpd]}} {{lighttpd|apache2|
mongoose|plackup|webrick}}
```

- Also auto-launch a web browser:

```
git instaweb --start {{[-b|--browser]}}
```

- Stop the currently running GitWeb server:

```
git instaweb --stop
```

- Restart the currently running GitWeb server:

```
git instaweb --restart
```

git-lfs-transfer

Server-side implementation of the Git LFS pure SSH transfer protocol.

Allows Git LFS to upload and download large files over SSH instead of HTTPS.

More information: https://github.com/charmbracelet/git-lfs-transfer#usage.

- Upload large files tracked by Git LFS to a repository:

  ```
  git-lfs-transfer {{path/to/repo.git}} upload
  ```

- Download large files tracked by Git LFS from a repository:

  ```
  git-lfs-transfer {{path/to/repo.git}} download
  ```

git lfs

Work with large files in Git repositories.

More information: https://github.com/git-lfs/git-lfs/tree/main/docs.

- Initialize Git LFS:

```
git lfs install
```

- Track files that match a glob:

```
git lfs track '{{*.bin}}'
```

- Change the Git LFS endpoint URL (useful if the LFS server is separate from the Git server):

```
git config {{[-f|--file]}} .lfsconfig lfs.url
{{lfs_endpoint_url}}
```

- List tracked patterns:

```
git lfs track
```

- List tracked files that have been committed:

```
git lfs ls-files
```

- Push all Git LFS objects to the remote server (useful if errors are encountered):

```
git lfs push --all {{remote_name}} {{branch_name}}
```

- Fetch all Git LFS objects:

```
git lfs fetch
```

- Replace pointer files with actual Git LFS objects:

```
git lfs checkout
```

git local-commits

Show local commits that haven't been pushed to origin. Any additional arguments will be passed directly to **`git log`**.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-local-commits.

- Show commits that haven't been pushed:

```
git local-commits
```

git lock

Lock a file in a Git repository from being modified by a commit.

Part of **`git-extras`**.

See also: **`git-unlock`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-lock.

- Disable the ability to commit changes of a local file:

```
git lock {{path/to/file}}
```

git locked

List locked files in a Git repository.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-locked.

- List all local locked files:

```
git locked
```

git log

Show a history of commits.

More information: https://git-scm.com/docs/git-log.

- Show the sequence of commits starting from the current one, in reverse chronological order of the Git repository in the current working directory:

```
git log
```

- Show the history of a particular file or directory, including differences:

```
git log {{[-p|--patch]}} {{path/to/file_or_directory}}
```

- Show an overview of which file(s) changed in each commit:

```
git log --stat
```

- Show a graph of commits in the current branch using only the first line of each commit message:

```
git log --oneline --graph
```

- Show a graph of all commits, tags and branches in the entire repo:

```
git log --oneline --decorate --all --graph
```

- Show only commits with messages that include a specific string, ignoring case:

```
git log {{[-i|--regexp-ignore-case]}} --grep
{{search_string}}
```

- Show the last `n` number of commits from a certain author:

```
git log {{[-n|--max-count]}} {{n}} --author "{{author}}"
```

- Show commits between two dates (yyyy-mm-dd):

```
git log --before "{{2017-01-29}}" --after "{{2017-01-17}}"
```

git ls-files

Show information about files in the index and the working tree.

More information: https://git-scm.com/docs/git-ls-files.

- Show deleted files:

```
git ls-files {{[-d|--deleted]}}
```

- Show modified and deleted files:

```
git ls-files {{[-m|--modified]}}
```

- Show ignored and untracked files:

```
git ls-files {{[-o|--others]}}
```

- Show untracked files, not ignored:

```
git ls-files {{[-o|--others]}} --exclude-standard
```

git ls-remote

Git command for listing references in a remote repository based on name or URL.

If no name or URL are given, then the configured upstream branch will be used, or remote origin if the former is not configured.

More information: https://git-scm.com/docs/git-ls-remote.

- Show all references in the default remote repository:

```
git ls-remote
```

- Show only heads references in the default remote repository:

```
git ls-remote --heads
```

- Show only tags references in the default remote repository:

```
git ls-remote {{[-t|--tags]}}
```

- Show all references from a remote repository based on name or URL:

```
git ls-remote {{repository_url}}
```

- Show references from a remote repository filtered by a pattern:

```
git ls-remote {{repository_name}} "{{pattern}}"
```

git ls-tree

List the contents of a tree object.

More information: https://git-scm.com/docs/git-ls-tree.

- List the contents of the tree on a branch:

```
git ls-tree {{branch_name}}
```

- List the contents of the tree on a commit, recursing into subtrees:

```
git ls-tree -r {{commit_hash}}
```

- List only the filenames of the tree on a commit:

```
git ls-tree --name-only {{commit_hash}}
```

- Print the filenames of the current branch head in a tree structure (Note: `tree --fromfile` is not supported on Windows):

```
git ls-tree -r --name-only HEAD | tree --fromfile
```

git magic

Automate add, commit, and push routines.

Part of **`git-extras`**.

More information: https://manned.org/git-magic.

- Commit changes with a generated message:

```
git magic
```

- [a]dd untracked files and commit changes with a generated message:

```
git magic -a
```

- Commit changes with a custom [m]essage:

```
git magic -m "{{custom_commit_message}}"
```

- [e]dit the commit [m]essage before committing:

```
git magic -em "{{custom_commit_message}}"
```

- Commit changes and [p]ush to remote:

```
git magic -p
```

- Commit changes with a [f]orce [p]ush to remote:

```
git magic -fp
```

git mailinfo

Extract patch and authorship information from a single email message.

More information: https://git-scm.com/docs/git-mailinfo.

- Extract the patch and author data from an email message:

```
git mailinfo {{message|patch}}
```

- Extract but remove leading and trailing whitespace:

```
git mailinfo -k {{message|patch}}
```

- Remove everything from the body before a scissors line (e.g. "-->* --") and retrieve the message or patch:

```
git mailinfo --scissors {{message|patch}}
```

git maintenance

Run tasks to optimize Git repository data.

More information: https://git-scm.com/docs/git-maintenance.

- Register the current repository in the user's list of repositories to daily have maintenance run:

```
git maintenance register
```

- Schedule maintenance tasks to run on the current repository every hour:

```
git maintenance start
```

- Halt the background maintenance schedule for the current repository:

```
git maintenance stop
```

- Remove the current repository from the user's maintenance repository list:

```
git maintenance unregister
```

- Run a specific maintenance task on the current repository:

```
git maintenance run --task {{commit-graph|gc|incremental-repack|loose-objects|pack-refs|prefetch}}
```

git merge-base

Find a common ancestor of two commits.

More information: https://git-scm.com/docs/git-merge-base.

- Print the best common ancestor of two commits:

```
git merge-base {{commit_1}} {{commit_2}}
```

- Print all best common ancestors of two commits:

```
git merge-base {{[-a|--all]}} {{commit_1}} {{commit_2}}
```

- Check if a commit is an ancestor of a specific commit:

```
git merge-base --is-ancestor {{ancestor_commit}} {{commit}}
```

git merge-index

Run a merge program on files that need merging.

More information: https://git-scm.com/docs/git-merge-index.

- Merge all files needing resolution using the standard helper:

```
git merge-index git-merge-one-file -a
```

- Merge a specific file:

```
git merge-index git-merge-one-file -- {{path/to/file}}
```

- Merge multiple files, continuing on failures:

```
git merge-index -o git-merge-one-file -- {{path/to/file1 path/to/file2 ...}}
```

- Quietly merge all files with a custom program:

```
git merge-index -q {{merge-program}} -a
```

- Inspect merge inputs for a file using `cat`:

```
git merge-index cat -- {{path}}
```

git merge-into

Merge one branch into another branch.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-merge-into.

- Merge a source branch into a specific destination branch:

```
git merge-into {{source_branch}} {{destination_branch}}
```

- Merge current branch into a specific destination branch:

```
git merge-into {{destination_branch}}
```

git merge-one-file

Resolve a single file merge after a trivial merge.

More information: https://git-scm.com/docs/git-merge-one-file.

- Resolve a simple merge conflict for a file:

```
git merge-one-file {{path/to/file}}
```

- Use as a helper in merge-index for a file:

```
git merge-index git-merge-one-file {{path/to/file}}
```

- Handle a binary file merge:

```
git merge-one-file -p {{path/to/file}}
```

- Apply after read-tree in a scripted merge:

```
git read-tree -m {{branch1}} {{branch2}} && git merge-index
git-merge-one-file {{path/to/file}}
```

git merge-repo

Merge two repository histories.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-merge-repo.

- Merge a repository's branch into the current repository's directory:

```
git merge-repo {{path/to/repo}} {{branch_name}} {{path/to/directory}}
```

- Merge a remote repository's branch into the current repository's directory, not preserving history:

```
git merge-repo {{path/to/remote_repo}} {{branch_name}} .
```

git merge-tree

Merge branches without changing files.

More information: https://git-scm.com/docs/git-merge-tree.

- Show the result of a merge between two branches:

```
git merge-tree {{branch1}} {{branch2}}
```

- Perform a merge and write the resulting tree:

```
git merge-tree --write-tree {{branch1}} {{branch2}}
```

git merge

Merge branches.

More information: https://git-scm.com/docs/git-merge.

- Merge a branch into your current branch:

```
git merge {{branch_name}}
```

- Edit the merge message:

```
git merge {{[-e|--edit]}} {{branch_name}}
```

- Merge a branch and create a merge commit:

```
git merge --no-ff {{branch_name}}
```

- Abort a merge in case of conflicts:

```
git merge --abort
```

- Merge using a specific strategy:

```
git merge {{[-s|--strategy]}} {{strategy}} {{[-X|--strategy-option]}} {{strategy_option}} {{branch_name}}
```

git mergetool

Run merge conflict resolution tools to resolve merge conflicts.

More information: https://git-scm.com/docs/git-mergetool.

- Launch the default merge tool to resolve conflicts:

```
git mergetool
```

- List valid merge tools:

```
git mergetool --tool-help
```

- Launch the merge tool identified by a name:

```
git mergetool {{[-t|--tool]}} {{tool_name}}
```

- Don't prompt before each invocation of the merge tool:

```
git mergetool {{[-y|--no-prompt]}}
```

- Explicitly use the GUI merge tool (see the `merge.guitool` configuration variable):

```
git mergetool {{[-g|--gui]}}
```

- Explicitly use the regular merge tool (see the `merge.tool` configuration variable):

```
git mergetool --no-gui
```

git missing

Show commits which aren't shared between two branches.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-missing.

- Show commits which aren't shared between the currently checked-out branch and another branch:

```
git missing {{branch}}
```

- Show commits which aren't shared between two branches:

```
git missing {{branch_1}} {{branch_2}}
```

git mktree

Build a tree object using **`ls-tree`** formatted text.

More information: https://git-scm.com/docs/git-mktree.

- Build a tree object and verify that each tree entry's hash identifies an existing object:

```
git mktree
```

- Allow missing objects:

```
git mktree --missing
```

- Read the NUL ([z]ero character) terminated output of the tree object (`git ls-tree -z`):

```
git mktree -z
```

- Allow the creation of multiple tree objects:

```
git mktree --batch
```

- Sort and build a tree from `stdin` (non-recursive `git ls-tree` output format is required):

```
git < {{path/to/tree.txt}} mktree
```

git mr

Check out GitLab merge requests locally.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-mr.

- Check out a specific merge request:

```
git mr {{mr_number}}
```

- Check out a merge request from a specific remote:

```
git mr {{mr_number}} {{remote}}
```

- Checkout a merge request from its URL:

```
git mr {{url}}
```

- Clean up old merge request branches:

```
git mr clean
```

git mv

Move or rename files and update the Git index.

More information: https://git-scm.com/docs/git-mv.

- Move a file inside the repo and add the movement to the next commit:

```
git mv {{path/to/file}} {{path/to/destination}}
```

- Rename a file or directory and add the renaming to the next commit:

```
git mv {{path/to/file_or_directory}} {{path/to/destination}}
```

- Overwrite the file or directory in the target path if it exists:

```
git mv {{[-f|--force]}} {{path/to/file_or_directory}} {{path/to/destination}}
```

git name-rev

Describe a commit using existing ref names.

More information: https://git-scm.com/docs/git-name-rev.

- Show the name for HEAD:

```
git name-rev HEAD
```

- Show only the name:

```
git name-rev --name-only HEAD
```

- Enumerate all matching ref names:

```
git name-rev --all
```

- Use only tags to name the commit:

```
git name-rev --tags HEAD
```

- Exit with a non-zero status code instead of printing `undefined` for unknown commits:

```
git name-rev --no-undefined {{commit-ish}}
```

- Show names for multiple commits:

```
git name-rev HEAD~1 HEAD~2 main
```

- Restrict names to branch refs:

```
git name-rev --refs refs/heads/ {{commit-ish}}
```

- Read commit IDs from `stdin`:

```
echo "{{commit-ish}}" | git name-rev --annotate-stdin
```

git notes

Add or inspect object notes.

More information: https://git-scm.com/docs/git-notes.

- List all notes and the objects they are attached to:

```
git notes list
```

- List all notes attached to a given object (defaults to HEAD):

```
git notes list [{{object}}]
```

- Show the notes attached to a given object (defaults to HEAD):

```
git notes show [{{object}}]
```

- Append a note to a specified object (opens the default text editor):

```
git notes append {{object}}
```

- Append a note to a specified object, specifying the message:

```
git notes append --message="{{message_text}}"
```

- Edit an existing note (defaults to HEAD):

```
git notes edit [{{object}}]
```

- Copy a note from one object to another:

```
git notes copy {{source_object}} {{target_object}}
```

- Remove all the notes added to a specified object:

```
git notes remove {{object}}
```

git obliterate

Delete files and erase their history from a Git repository.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-obliterate.

- Erase the existence of specific files:

```
git obliterate {{file_1 file_2 ...}}
```

- Erase the existence of specific files between 2 commits:

```
git obliterate {{file_1 file_2 ...}} -- {{commit_hash_1}}..{{commit_hash_2}}
```

git p4

Import from and submit to Perforce repositories.

More information: https://git-scm.com/docs/git-p4.

- Clone a Perforce depot into a new Git repository:

```
git p4 clone {{path/to/p4_depot}}
```

- Sync changes from Perforce into the current Git repository:

```
git p4 sync {{path/to/p4_depot}}
```

- Rebase local commits on top of the latest Perforce changes:

```
git p4 rebase
```

- Submit Git changes back to Perforce:

```
git p4 submit
```

- Clone the full Perforce history instead of only the latest changelist:

```
git p4 clone {{path/to/p4_depot}}@all
```

git paste

Send commits to a pastebin site using **`pastebinit`**.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-paste.

- Send the patches between the current branch and its upstream to a pastebin using `pastebinit`:

```
git paste
```

- Pass options to `git format-patch` in order to select a different set of commits (`@^` selects the parent of HEAD, and so the currently checked out commit is sent):

```
git paste {{@^}}
```

git pr

Check out GitHub pull requests locally.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-pr.

- Check out a specific pull request:

```
git pr {{pr_number}}
```

- Check out a pull request from a specific remote:

```
git pr {{pr_number}} {{remote}}
```

- Check out a pull request from its URL:

```
git pr {{url}}
```

- Clean up old pull request branches:

```
git pr clean
```

git prune

Git command for pruning all unreachable objects from the object database.

This command is often not used directly but as an internal command that is used by Git gc.

More information: https://git-scm.com/docs/git-prune.

- Report what would be removed by Git prune without removing it:

```
git prune {{[-n|--dry-run]}}
```

- Prune unreachable objects and display what has been pruned to `stdout`:

```
git prune {{[-v|--verbose]}}
```

- Prune unreachable objects while showing progress:

```
git prune --progress
```

git psykorebase

Rebase a branch on top of another using a merge commit and only one conflict handling.

Part of **`git-extras`**.

More information: https://manned.org/git-psykorebase.

- Rebase the current branch on top of another using a merge commit and only one conflict handling:

```
git psykorebase {{upstream_branch}}
```

- Continue after conflicts have been handled:

```
git psykorebase --continue
```

- Specify the branch to rebase:

```
git psykorebase {{upstream_branch}} {{target_branch}}
```

git pull-request

Create a pull request for a project on GitHub.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-pull-request.

- Create a pull request for a project on GitHub:

```
git pull-request {{target_branch}}
```

git pull

Fetch branch from a remote repository and merge it to local repository.

More information: https://git-scm.com/docs/git-pull.

- Download changes from default remote repository and merge it:

```
git pull
```

- Download changes from default remote repository and use fast-forward:

```
git pull {{[-r|--rebase]}}
```

- Download changes from given remote repository and branch, then merge them into HEAD:

```
git pull {{remote_name}} {{branch}}
```

git push

Push commits to a remote repository.

More information: https://git-scm.com/docs/git-push.

- Send local changes in the current branch to its default remote counterpart:

```
git push
```

- Send changes from a specific local branch to its remote counterpart:

```
git push {{remote_name}} {{local_branch}}
```

- Send changes from a specific local branch to its remote counterpart, and set the remote one as the default push/pull target of the local one:

```
git push {{[-u|--set-upstream]}} {{remote_name}} {{local_branch}}
```

- Send changes from a specific local branch to a specific remote branch:

```
git push {{remote_name}} {{local_branch}}:{{remote_branch}}
```

- Send changes on all local branches to their counterparts in a given remote repository:

```
git push --all {{remote_name}}
```

- Delete a branch in a remote repository:

```
git push {{remote_name}} {{[-d|--delete]}} {{remote_branch}}
```

- Remove remote branches that don't have a local counterpart:

```
git push --prune {{remote_name}}
```

- Publish tags that aren't yet in the remote repository:

```
git push --tags
```

git range-diff

Compare two commit ranges (e.g. two versions of a branch).

More information: https://git-scm.com/docs/git-range-diff.

- Diff the changes of two individual commits:

```
git range-diff {{commit_1}}^! {{commit_2}}^!
```

- Diff the changes of ours and theirs from their common ancestor, e.g. after an interactive rebase:

```
git range-diff {{theirs}}...{{ours}}
```

- Diff the changes of two commit ranges, e.g. to check whether conflicts have been resolved appropriately when rebasing commits from `base1` to `base2`:

```
git range-diff {{base1}}..{{rev1}} {{base2}}..{{rev2}}
```

git reauthor

Change details about an author identity. Since this command rewrites the Git history, **`--force`** will be needed when pushing next time.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-reauthor.

- Change an author's email and name across the whole Git repository:

```
git reauthor {{[-o|--old-email]}} {{old@example.com}} {{[-e|--correct-email]}} {{new@example.com}} {{[-n|--correct-name]}} "{{name}}"
```

- Change the email and name to the ones defined in the Git config:

```
git reauthor {{[-o|--old-email]}} {{old@example.com}} {{[-c|--use-config]}}
```

- Change the email and name of all commits, regardless of their original author:

```
git reauthor {{[-a|--all]}} {{[-e|--correct-email]}} {{name@example.com}} {{[-n|--correct-name]}} {{name}}
```

git rebase-patch

Find the commit the patch applies to and do a rebase.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-rebase-patch.

- Find the commit the patch applies to and do a rebase:

```
git rebase-patch {{patch_file}}
```

git rebase

Reapply commits from one branch on top of another branch.

Commonly used to "move" an entire branch to another base, creating copies of the commits in the new location.

More information: https://git-scm.com/docs/git-rebase.

- Rebase the current branch on top of another specified branch:

```
git rebase {{new_base_branch}}
```

- Start an interactive rebase, which allows the commits to be reordered, omitted, combined or modified:

```
git rebase {{[-i|--interactive]}}
{{target_base_branch_or_commit_hash}}
```

- Continue a rebase that was interrupted by a merge failure, after editing conflicting files:

```
git rebase --continue
```

- Continue a rebase that was paused due to merge conflicts, by skipping the conflicted commit:

```
git rebase --skip
```

- Abort a rebase in progress (e.g. if it is interrupted by a merge conflict):

```
git rebase --abort
```

- Move part of the current branch onto a new base, providing the old base to start from:

```
git rebase --onto {{new_base}} {{old_base}}
```

- Reapply the last 5 commits in-place, stopping to allow them to be reordered, omitted, combined or modified:

```
git rebase {{[-i|--interactive]}} {{HEAD~5}}
```

- Auto-resolve any conflicts by favoring the working branch version (`theirs` keyword has reversed meaning in this case):

```
git rebase {{[-X|--strategy-option]}} theirs {{branch_name}}
```

git reflog

Show a log of changes to local references like HEAD, branches or tags.

More information: https://git-scm.com/docs/git-reflog.

- Show the reflog for HEAD:

```
git reflog
```

- Show the reflog for a given branch:

```
git reflog {{branch_name}}
```

- Show only the 5 latest entries in the reflog:

```
git reflog {{[-n|--max-count]}} 5
```

git release

Create a Git tag for a release.

Part of **`git-extras`**.

More information: https://manned.org/git-release.

- Create and push a release:

```
git release {{tag_name}}
```

- Create and push a signed release:

```
git release {{tag_name}} -s
```

- Create and push a release with a message:

```
git release {{tag_name}} -m "{{message}}"
```

git remote

Manage set of tracked repositories ("remotes").

More information: https://git-scm.com/docs/git-remote.

- List existing remotes with their names and URLs:

```
git remote {{[-v|--verbose]}}
```

- Show information about a remote:

```
git remote show {{remote_name}}
```

- Add a remote:

```
git remote add {{remote_name}} {{remote_url}}
```

- Change the URL of a remote (use `--add` to keep the existing URL):

```
git remote set-url {{remote_name}} {{new_url}}
```

- Show the URL of a remote:

```
git remote get-url {{remote_name}}
```

- Remove a remote:

```
git remote remove {{remote_name}}
```

- Rename a remote:

```
git remote rename {{old_name}} {{new_name}}
```

git rename-branch

Rename a Git branch.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-rename-branch.

- Rename the branch you are currently on:

```
git rename-branch {{new_branch_name}}
```

- Rename a specific branch:

```
git rename-branch {{old_branch_name}} {{new_branch_name}}
```

git rename-remote

Change remote for pulling and pushing.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-rename-remote.

- Change the upstream remote to origin:

```
git rename-remote {{upstream}} {{origin}}
```

git rename-tag

Rename a Git tag.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-rename-tag.

- Rename an existing Git tag locally and remotely:

```
git rename-tag {{old_tag_name}} {{new_tag_name}}
```

git repack

Pack unpacked objects in a Git repository.

More information: https://git-scm.com/docs/git-repack.

- Pack unpacked objects in the current directory:

```
git repack
```

- Remove redundant objects after packing:

```
git repack -d
```

- Repack all objects into a single pack:

```
git repack -a
```

- Limit the repack to local objects only:

```
git repack -l
```

git repl

Git REPL (read-evaluate-print-loop) - an interactive Git shell.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-repl.

- Start an interactive Git shell:

```
git repl
```

- Run a Git command while in the interactive Git shell:

```
{{git_subcommand}} {{command_arguments}}
```

- Run an external (non-Git) command while in the interactive Git shell:

```
!{{command}} {{command_arguments}}
```

- Exit the interactive Git shell (or press `<Ctrl d>`):

```
exit
```

git replace

Create, list, and delete refs to replace objects.

More information: https://git-scm.com/docs/git-replace.

- Replace any commit with a different one, leaving other commits unchanged:

```
git replace {{object}} {{replacement}}
```

- Delete existing replace refs for the given objects:

```
git replace {{[-d|--delete]}} {{object}}
```

- Edit an object's content interactively:

```
git replace --edit {{object}}
```

git request-pull

Generate a request asking the upstream project to pull changes into its tree.

More information: https://git-scm.com/docs/git-request-pull.

- Produce a request summarizing the changes between the v1.1 release and a specified branch:

```
git request-pull {{v1.1}} {{https://example.com/project}} {{branch_name}}
```

- Produce a request summarizing the changes between the v0.1 release on the `foo` branch and the local `bar` branch:

```
git request-pull {{v0.1}} {{https://example.com/project}} {{foo:bar}}
```

git rerere

Reuse fixes for merge conflicts.

More information: https://git-scm.com/docs/git-rerere.

- Enable rerere globally:

```
git config --global rerere.enabled true
```

- Forget a file's recorded resolution:

```
git rerere forget {{path/to/file}}
```

- Check the status of recorded resolutions:

```
git rerere status
```

git reset-file

Revert a file to HEAD or a commit.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-reset-file.

- Reset a file to HEAD:

```
git reset-file {{path/to/file}}
```

- Reset a file to a specific commit:

```
git reset-file {{path/to/file}} {{commit_hash}}
```

git reset

Undo commits or unstage changes, by resetting the current Git HEAD to the specified state.

If a path is passed, it works as "unstage"; if a commit hash or branch is passed, it works as "uncommit".

More information: https://git-scm.com/docs/git-reset.

- Unstage everything:

```
git reset
```

- Unstage specific file(s):

```
git reset {{path/to/file1 path/to/file2 ...}}
```

- Interactively unstage portions of a file:

```
git reset {{[-p|--patch]}} {{path/to/file}}
```

- Undo the last commit, keeping its changes (and any further uncommitted changes) in the filesystem:

```
git reset HEAD~
```

- Undo the last two commits, adding their changes to the index, i.e. staged for commit:

```
git reset --soft HEAD~2
```

- Discard any uncommitted changes, staged or not (for only unstaged changes, use `git checkout`):

```
git reset --hard
```

- Reset the repository to a given commit, discarding committed, staged and uncommitted changes since then:

```
git reset --hard {{commit}}
```

git restore

Restore working tree files. Requires Git version 2.23+.

See also: **`git checkout`**, **`git reset`**.

More information: https://git-scm.com/docs/git-restore.

- Restore an unstaged file to the staged version:

```
git restore {{path/to/file}}
```

- Restore an unstaged file to the version of a specific commit:

```
git restore {{[-s|--source]}} {{commit}} {{path/to/file}}
```

- Discard all unstaged changes to tracked files:

```
git restore :/
```

- Unstage a file:

```
git restore {{[-S|--staged]}} {{path/to/file}}
```

- Unstage all files:

```
git restore {{[-S|--staged]}} :/
```

- Discard all changes to files, both staged and unstaged:

```
git restore {{[-W|--worktree]}} {{[-S|--staged]}} :/
```

- Interactively select sections of files to restore:

```
git restore {{[-p|--patch]}}
```

git rev-list

List revisions (commits) in reverse chronological order.

More information: https://git-scm.com/docs/git-rev-list.

- List all commits on the current branch:

```
git rev-list {{HEAD}}
```

- Print the latest commit that changed (add/edit/remove) a specific file on the current branch:

```
git rev-list {{[-n|--max-count]}} 1 HEAD -- {{path/to/file}}
```

- List commits more recent than a specific date, on a specific branch:

```
git rev-list --since "{{2019-12-01
00:00:00}}" {{branch_name}}
```

- List all merge commits on a specific commit:

```
git rev-list --merges {{commit}}
```

- Print the number of commits since a specific tag:

```
git rev-list {{tag_name}}..HEAD --count
```

git rev-parse

Display metadata related to revisions.

More information: https://git-scm.com/docs/git-rev-parse.

- Get the commit hash of a branch:

```
git rev-parse {{branch_name}}
```

- Get the current branch name:

```
git rev-parse --abbrev-ref {{HEAD}}
```

- Get the absolute path to the root directory:

```
git rev-parse --show-toplevel
```

git revert

Create new commits which reverse the effect of earlier ones.

More information: https://git-scm.com/docs/git-revert.

- Revert the most recent commit:

```
git revert {{HEAD}}
```

- Revert the 5th last commit:

```
git revert HEAD~{{4}}
```

- Revert a specific commit:

```
git revert {{0c01a9}}
```

- Revert multiple commits:

```
git revert {{branch_name~5..branch_name~2}}
```

- Don't create new commits, just change the working tree:

```
git revert {{[-n|--no-commit]}} {{0c01a9..9a1743}}
```

- Cancel a Git revert after a merge conflict:

```
git revert --abort
```

git rm

Remove files from repository index and local filesystem.

More information: https://git-scm.com/docs/git-rm.

- Remove file from repository index and filesystem:

```
git rm {{path/to/file}}
```

- Remove directory:

```
git rm -r {{path/to/directory}}
```

- Remove file from repository index but keep it untouched locally:

```
git rm --cached {{path/to/file}}
```

git root

Print the root directory of the current Git repository.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-root.

- Print the absolute path of the current Git repository:

```
git root
```

- Print the current working directory relative to the root of the current Git repository:

```
git root {{[-r|--relative]}}
```

git rscp

Reverse **`git scp`** - copy files from the working directory of a remote repository to the current working tree.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-scp.

- Copy specific files from a remote:

```
git rscp {{remote_name}} {{path/to/file1 path/to/file2 ...}}
```

- Copy a specific directory from a remote:

```
git rscp {{remote_name}} {{path/to/directory}}
```

git scp

Copy files from the current working tree to the working directory of a remote repository.

Part of **`git-extras`**. Uses **`rsync`** to transfer files.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-scp.

- Copy unstaged files to a specific remote:

```
git scp {{remote_name}}
```

- Copy staged and unstaged files to a remote:

```
git scp {{remote_name}} HEAD
```

- Copy files that has been changed in the last commit and any staged or unstaged files to a remote:

```
git scp {{remote_name}} HEAD~1
```

- Copy specific files to a remote:

```
git scp {{remote_name}} {{path/to/file1 path/to/file2 ...}}
```

- Copy a specific directory to a remote:

```
git scp {{remote_name}} {{path/to/directory}}
```

git secret

Stores private data inside a Git repository. Written in Bash.

More information: https://github.com/sobolevn/git-secret.

- Initialize `git-secret` in a local repository:

```
git secret init
```

- Grant access to the current Git user's email:

```
git secret tell -m
```

- Grant access by email:

```
git secret tell {{email}}
```

- Revoke access by email:

```
git secret killperson {{email}}
```

- List emails with access to secrets:

```
git secret whoknows
```

- Register a secret file:

```
git secret add {{path/to/file}}
```

- Encrypt secrets:

```
git secret hide
```

- Decrypt secret files:

```
git secret reveal
```

git sed

Replace patterns in git-controlled files using sed.

Part of **`git-extras`**.

More information: https://manned.org/git-sed.

- Replace the specified text in the current repository:

```
git sed '{{find_text}}' '{{replace_text}}'
```

- Replace the specified text and then commit the resulting changes with a standard commit message:

```
git sed -c '{{find_text}}' '{{replace_text}}'
```

- Replace the specified text, using `regex`:

```
git sed -f g '{{find_text}}' '{{replace_text}}'
```

- Replace a specific text in all files under a given directory:

```
git sed '{{find_text}}' '{{replace_text}}' -- {{path/to/directory}}
```

git send-email

Send a collection of patches as emails.

Patches can be specified as files, directions, or a revision list.

More information: https://git-scm.com/docs/git-send-email.

- Send the last commit in the current branch interactively:

```
git send-email -1
```

- Send a given commit:

```
git send-email -1 {{commit}}
```

- Send multiple (e.g. 10) commits in the current branch:

```
git send-email {{-10}}
```

- Send an introductory email message for the patch series:

```
git send-email -{{number_of_commits}} --compose
```

- Review and edit the email message for each patch you're about to send:

```
git send-email -{{number_of_commits}} --annotate
```

git setup

Create a Git repository in a directory and commit all files.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-setup.

- Create a Git repository in the current directory and commit all files:

```
git setup
```

- Create a Git repository in a specific directory and commit all files:

```
git setup {{path/to/directory}}
```

git shortlog

Summarizes the **`git log`** output.

More information: https://git-scm.com/docs/git-shortlog.

- View a summary of all the commits made, grouped alphabetically by author name:

```
git shortlog
```

- View a summary of all the commits made, sorted by the number of commits made:

```
git shortlog {{[-n|--numbered]}}
```

- View a summary of all the commits made, grouped by the committer identities (name and email):

```
git shortlog {{[-c|--committer]}}
```

- View a summary of the last 5 commits (i.e. specify a revision range):

```
git shortlog HEAD~5..HEAD
```

- View all users, emails and the number of commits in the current branch:

```
git shortlog {{[-s|--summary]}} {{[-n|--numbered]}} {{[-e|--email]}}
```

- View all users, emails and the number of commits in all branches:

```
git shortlog {{[-s|--summary]}} {{[-n|--numbered]}} {{[-e|--email]}} --all
```

git show-branch

Show branches and their commits.

More information: https://git-scm.com/docs/git-show-branch.

- Show a summary of the latest commit on a branch:

```
git show-branch {{branch_name|ref|commit}}
```

- Compare commits in the history of multiple commits or branches:

```
git show-branch {{branch_name1|ref1|commit1 branch_name2|
ref2|commit2 ...}}
```

- Compare all remote tracking branches:

```
git show-branch {{[-r|--remotes]}}
```

- Compare both local and remote tracking branches:

```
git show-branch {{[-a|--all]}}
```

- List the latest commits in all branches:

```
git show-branch {{[-a|--all]}} --list
```

- Compare a given branch with the current branch:

```
git show-branch --current {{commit|branch_name|ref}}
```

- Display the commit name instead of the relative name:

```
git show-branch --sha1-name --current {{current|branch_name|
ref}}
```

- Keep going a given number of commits past the common ancestor:

```
git show-branch --more {{5}} {{branch_name1|ref1|commit1
branch_name2|ref2|commit2 ...}}
```

git show-index

Show the packed archive index of a Git repository.

More information: https://git-scm.com/docs/git-show-index.

- Read an IDX file for a Git packfile and dump its contents to `stdout`:

```
git show-index {{path/to/file.idx}}
```

- Specify the hash algorithm for the index file (experimental):

```
git show-index --object-format {{sha1|sha256}} {{path/to/file}}
```

git show-merged-branches

Print all branches which are merged into the current head.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-show-merged-branches.

- Print all branches which are merged into the current head:

```
git show-merged-branches
```

git show-ref

Git command for listing references.

More information: https://git-scm.com/docs/git-show-ref.

- Show all refs in the repository:

```
git show-ref
```

- Show only heads references:

```
git show-ref --branches
```

- Show only tags references:

```
git show-ref --tags
```

- Verify that a given reference exists:

```
git show-ref --verify {{path/to/ref}}
```

git show-tree

Show a decorated tree graph with all branches of a Git repository, showing annotations.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-show-tree.

- Show a decorated tree graph for all branches annotated with tags and branch names:

```
git show-tree
```

git show-unmerged-branches

Print all branches which are not merged into the current HEAD.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-show-unmerged-branches.

- Print all branches which are not merged into the current HEAD:

```
git show-unmerged-branches
```

git show

Show various types of Git objects (commits, tags, etc.).

More information: https://git-scm.com/docs/git-show.

- Show information about the latest commit (hash, message, changes, and other metadata):

```
git show
```

- Show information about a specific commit, tag, or branch (such as `HEAD` for the latest commit):

```
git show {{commit|tag|branch}}
```

- Show only the list of added, renamed or deleted files:

```
git show --summary {{commit}}
```

- Show information about the 3rd commit from the HEAD of a branch:

```
git show {{branch}}~{{3}}
```

- Show a commit's message in a single line, suppressing the diff output:

```
git show --oneline {{[-s|--no-patch]}} {{commit}}
```

- Show only statistics (added/removed characters) about the changed files:

```
git show --stat {{commit}}
```

- Show a simplified list of all files changed in a commit (modified, added, and deleted):

```
git show --name-only {{commit}}
```

- Show the contents of a file as it was at a given revision (e.g. branch, tag or commit):

```
git show {{revision}}:{{path/to/file}}
```

git-sizer

Computes various Git repository size metrics and alerts you to any that might cause problems or inconvenience.

More information: https://github.com/github/git-sizer.

- Report only statistics that have a level of concern greater than 0:

```
git-sizer
```

- Report all statistics:

```
git-sizer -v
```

- See additional options:

```
git-sizer -h
```

git sparse-checkout

Check out only part of a repository's files instead of cloning or checking out everything.

More information: https://manned.org/git-sparse-checkout.

- Enable sparse checkout:

```
git sparse-checkout init
```

- Disable sparse-checkout and restore full repository:

```
git sparse-checkout disable
```

- Specify which directories (or files) to include:

```
git sparse-checkout set {{path/to/directory}}
```

- Add more paths later:

```
git sparse-checkout add {{path/to/directory}}
```

git squash

Squash multiple commits into a single commit.

Part of **`git-extras`**.

More information: https://manned.org/git-squash.

- Merge all commits from a specific branch into the current branch as a single commit:

```
git squash {{source_branch}}
```

- Squash all commits starting with a specific commit on the current branch:

```
git squash {{commit}}
```

- Squash the n latest commits and commit with a message:

```
git squash HEAD~{{n}} "{{message}}"
```

- Squash the n latest commits and commit concatenating all individual messages:

```
git squash --squash-msg HEAD~{{n}}
```

git stage

This command is an alias of **`git add`**.

- View documentation for the original command:

```
tldr git add
```

git stamp

Stamp the last commit message, with the possibility to reference the issues numbers from your bug tracker or link to its review page.

Part of **`git-extras`**.

More information: https://manned.org/git-stamp.

- Stamp the last commit message referencing it with the issue number from your bug tracker:

```
git stamp {{issue_number}}
```

- Stamp the last commit message linking it to its review page:

```
git stamp {{Review https://example.org/path/to/review}}
```

- Stamp the last commit message replacing previous issues with a new one:

```
git stamp {{[-r|--replace]}} {{issue_number}}
```

git standup

See commits from a specified user.

Part of **`git-extras`**.

More information: https://manned.org/git-standup.

- Show a given author's commits from the last 10 days:

```
git standup -a {{name|email}} -d {{10}}
```

- Show a given author's commits from the last 10 days and whether they are GPG signed:

```
git standup -a {{name|email}} -d {{10}} -g
```

- Show all the commits from all contributors for the last 10 days:

```
git standup -a all -d {{10}}
```

- Display help:

```
git standup -h
```

git stash

Stash local Git changes in a temporary area.

More information: https://git-scm.com/docs/git-stash.

- Stash current changes with a message, except new (untracked) files:

```
git stash push {{[-m|--message]}} {{stash_message}}
```

- Stash current changes, including new untracked files:

```
git stash {{[-u|--include-untracked]}}
```

- Interactively select parts of changed files for stashing:

```
git stash {{[-p|--patch]}}
```

- List all stashes (shows stash name, related branch and message):

```
git stash list
```

- Show the changes as a patch between the stash (default is `stash@{0}`) and the commit back when stash entry was first created:

```
git stash show {{[-p|--patch]}} {{stash@{0}}}
```

- Apply a stash (default is the latest, named `stash@{0}`):

```
git stash apply {{optional_stash_name_or_commit}}
```

- Drop or apply a stash (default is `stash@{0}`) and remove it from the stash list if applying doesn't cause conflicts:

```
git stash pop {{optional_stash_name}}
```

- Drop all stashes:

```
git stash clear
```

git status

Show the changes to files in a Git repository.

List changed, added and deleted files compared to the currently checked-out commit.

More information: https://git-scm.com/docs/git-status.

- Show changed files which are not yet added for commit:

```
git status
```

- Give output in short format:

```
git status {{[-s|--short]}}
```

- Show verbose information on changes in both the staging area and working directory:

```
git status {{[-vv|--verbose --verbose]}}
```

- Show the branch and tracking info:

```
git status {{[-b|--branch]}}
```

- Show output in short format along with branch info:

```
git status {{[-sb|--short --branch]}}
```

- Show the number of entries currently stashed away:

```
git status --show-stash
```

- Don't show untracked files in the output:

```
git status {{[-uno|--untracked-files=no]}}
```

git stripspace

Read text (e.g. commit messages, notes, tags, and branch descriptions) from **`stdin`** and clean it into the manner used by Git.

More information: https://git-scm.com/docs/git-stripspace.

- Trim whitespace from a file:

```
cat {{path/to/file}} | git stripspace
```

- Trim whitespace and Git comments from a file:

```
cat {{path/to/file}} | git stripspace {{[-s|--strip-comments]}}
```

- Convert all lines in a file into Git comments:

```
git < {{path/to/file}} stripspace {{[-c|--comment-lines]}}
```

git submodule

Inspects, updates and manages submodules.

More information: https://git-scm.com/docs/git-submodule.

- Install a repository's specified submodules:

```
git submodule update --init --recursive
```

- Add a Git repository as a submodule:

```
git submodule add {{repository_url}}
```

- Add a Git repository as a submodule at the specified directory:

```
git submodule add {{repository_url}} {{path/to/directory}}
```

- Update every submodule to its latest commit:

```
git submodule update --init --recursive --remote
```

git subtree

Merge subtrees together or split repository into subtrees.

More information: https://manned.org/git-subtree.

- Add a Git repository as a subtree and squash the commits together:

```
git subtree add {{[-P|--prefix]}} {{path/to/directory}} --
squash {{repository_url}} {{branch_name}}
```

- Update subtree repository to its latest commit:

```
git subtree pull {{[-P|--prefix]}} {{path/to/directory}}
{{repository_url}} {{branch_name}}
```

- Merge recent changes up to the latest subtree commit into the subtree:

```
git subtree merge {{[-P|--prefix]}} {{path/to/directory}} --
squash {{repository_url}} {{branch_name}}
```

- Push commits to a subtree repository:

```
git subtree push {{[-P|--prefix]}} {{path/to/directory}}
{{repository_url}} {{branch_name}}
```

- Extract a new project history from the history of a subtree:

```
git subtree split {{[-P|--prefix]}} {{path/to/directory}}
{{repository_url}} {{[-b|--branch]}} {{branch_name}}
```

git summary

Display information about a Git repository.

Part of **`git-extras`**.

More information: https://manned.org/git-summary.

- Display data about a Git repository:

```
git summary
```

- Display data about a Git repository since a commit-ish:

```
git summary {{commit|branch_name|tag_name}}
```

- Display data about a Git repository, merging committers using different emails into 1 statistic for each author:

```
git summary --dedup-by-email
```

- Display data about a Git repository, showing the number of lines modified by each contributor:

```
git summary --line
```

git svn

Bidirectional operation between a Subversion repository and Git.

More information: https://git-scm.com/docs/git-svn.

- Clone an SVN repository:

```
git svn clone {{https://example.com/subversion_repo}} {{local_directory}}
```

- Clone an SVN repository starting at a given revision number:

```
git svn clone {{[-r|--revision]}} {{1234}}:HEAD {{https://svn.example.net/subversion/repo}} {{local_directory}}
```

- Update local clone from the remote SVN repository:

```
git svn rebase
```

- Fetch updates from the remote SVN repository without changing the Git HEAD:

```
git svn fetch
```

- Commit back to the SVN repository:

```
git svn commit
```

git switch

Switch between Git branches. Requires Git version 2.23+.

See also: **`git checkout`**.

More information: https://git-scm.com/docs/git-switch.

- Switch to an existing branch:

```
git switch {{branch_name}}
```

- Create a new branch and switch to it:

```
git switch {{[-c|--create]}} {{branch_name}}
```

- Create a new branch based on an existing commit and switch to it:

```
git switch {{[-c|--create]}} {{branch_name}} {{commit}}
```

- Switch to the previous branch:

```
git switch -
```

- Switch to a branch and update all submodules to match:

```
git switch --recurse-submodules {{branch_name}}
```

- Switch to a branch and automatically merge the current branch and any uncommitted changes into it:

```
git switch {{[-m|--merge]}} {{branch_name}}
```

- Switch to a tag:

```
git switch {{[-d|--detach]}} {{tag}}
```

git symbolic-ref

Read, change, or delete files that store references.

More information: https://git-scm.com/docs/git-symbolic-ref.

- Store a reference by a name:

```
git symbolic-ref refs/{{name}} {{ref}}
```

- Store a reference by name, including a message with a reason for the update:

```
git symbolic-ref -m "{{message}}" refs/{{name}} refs/heads/{{branch_name}}
```

- Read a reference by name:

```
git symbolic-ref refs/{{name}}
```

- Delete a reference by name:

```
git symbolic-ref {{[-d|--delete]}} refs/{{name}}
```

- For scripting, hide errors with `--quiet` and use `--short` to simplify ("refs/heads/X" prints as "X"):

```
git symbolic-ref {{[-q|--quiet]}} --short refs/{{name}}
```

git sync

Sync local branches with remote branches.

Part of **`git-extras`**.

More information: https://manned.org/git-sync.

- Sync the current local branch with its remote branch:

```
git sync
```

- Sync the current local branch with the remote main branch:

```
git sync origin main
```

- Sync without cleaning untracked files:

```
git sync {{[-s|--soft]}} {{remote_name}} {{branch_name}}
```

git tag

Create, list, delete or verify tags.

A tag is a static reference to a commit.

More information: https://git-scm.com/docs/git-tag.

- List all tags:

```
git tag
```

- Create a tag with the given name pointing to the current commit:

```
git tag {{tag_name}}
```

- Create a tag with the given name pointing to a given commit:

```
git tag {{tag_name}} {{commit}}
```

- Create an annotated tag with the given message:

```
git tag {{tag_name}} {{[-m|--message]}} {{tag_message}}
```

- Delete the tag with the given name:

```
git tag {{[-d|--delete]}} {{tag_name}}
```

- Get updated tags from remote:

```
git fetch {{[-t|--tags]}}
```

- Push a tag to remote:

```
git push origin tag {{tag_name}}
```

- List all tags which contain a given commit (HEAD if not specified):

```
git tag --contains {{commit}}
```

git touch

Create new files and add them to the index.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-touch.

- Create new files and add them to the index:

```
git touch {{path/to/file1 path/to/file2 ...}}
```

git undo

Undo recent commits.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-undo.

- Remove the most recent commit:

```
git undo
```

- Remove a specific number of the most recent commits:

```
git undo {{3}}
```

git unlock

Unlock a file in a Git repository so it can be modified by a commit.

Part of **`git-extras`**.

See also: **`git lock`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-unlock.

- Enable the ability to commit changes of a previously-locked local file:

```
git unlock {{path/to/file}}
```

git unpack-file

Create a temporary file with a blob's contents.

More information: https://git-scm.com/docs/git-unpack-file.

- Create a file holding the contents of the blob specified by its ID then print the name of the temporary file:

```
git unpack-file {{blob_id}}
```

git update-index

Git command for manipulating the index.

More information: https://git-scm.com/docs/git-update-index.

- Pretend that a modified file is unchanged (`git status` will not show this as changed):

```
git update-index --skip-worktree {{path/to/modified_file}}
```

git update-ref

Git command for creating, updating, and deleting Git refs.

More information: https://git-scm.com/docs/git-update-ref.

- Delete a ref, useful for soft resetting the first commit:

```
git update-ref -d {{HEAD}}
```

- Update ref with a message:

```
git update-ref -m {{message}} {{HEAD}} {{4e95e05}}
```

git utimes

Change files modification time to their last commit date. Does not touch files that are in the working tree or index.

Part of **`git-extras`**.

More information: https://github.com/tj/git-extras/blob/master/Commands.md#git-utimes.

- Change all files modification time to their last commit date:

  ```
  git utimes
  ```

- Change files modification time that are newer than their last commit date, preserving original modification time of files that were committed from the local repository:

  ```
  git utimes --newer
  ```

git var

Print a Git logical variable's value.

See **`git config`**, which is preferred over **`git var`**.

More information: https://git-scm.com/docs/git-var.

- Print the value of a Git logical variable:

```
git var {{GIT_AUTHOR_IDENT|GIT_COMMITTER_IDENT|GIT_EDITOR|
GIT_PAGER}}
```

- [l]ist all Git logical variables:

```
git var -l
```

git verify-commit

Check for GPG verification of commits.

If no commits are verified, nothing will be printed, regardless of options specified.

More information: https://git-scm.com/docs/git-verify-commit.

- Check commits for a GPG signature:

```
git verify-commit {{commit_hash1 optional_commit_hash2 ...}}
```

- Check commits for a GPG signature and show details of each commit:

```
git verify-commit {{commit_hash1 optional_commit_hash2 ...}}
{{[-v|--verbose]}}
```

- Check commits for a GPG signature and print the raw details:

```
git verify-commit {{commit_hash1 optional_commit_hash2 ...}}
--raw
```

git verify-pack

Verify packed Git archive files.

More information: https://git-scm.com/docs/git-verify-pack.

- Verify a packed Git archive file:

```
git verify-pack {{path/to/pack-file}}
```

- Verify a packed Git archive file and show verbose details:

```
git verify-pack {{[-v|--verbose]}} {{path/to/pack-file}}
```

- Verify a packed Git archive file and only display the statistics:

```
git verify-pack {{[-s|--stat-only]}} {{path/to/pack-file}}
```

git verify-tag

Check for GPG verification of tags.

If a tag wasn't signed, an error will occur.

More information: https://git-scm.com/docs/git-verify-tag.

- Check tags for a GPG signature:

```
git verify-tag {{tag1 optional_tag2 ...}}
```

- Check tags for a GPG signature and show details for each tag:

```
git verify-tag {{tag1 optional_tag2 ...}} {{[-v|--verbose]}}
```

- Check tags for a GPG signature and print the raw details:

```
git verify-tag {{tag1 optional_tag2 ...}} --raw
```

git whatchanged

Show what has changed with recent commits or files.

See also: **`git log`**.

More information: https://git-scm.com/docs/git-whatchanged.

- Display logs and changes for recent commits:

```
git whatchanged
```

- Display logs and changes for recent commits within the specified time frame:

```
git whatchanged --since "{{2 hours ago}}"
```

- Display logs and changes for recent commits for specific files or directories:

```
git whatchanged {{path/to/file_or_directory}}
```

git worktree

Manage multiple working trees attached to the same repository.

More information: https://git-scm.com/docs/git-worktree.

- Create a new directory with the specified branch checked out into it:

```
git worktree add {{path/to/directory}} {{branch}}
```

- Create a new directory with a new branch checked out into it:

```
git worktree add {{path/to/directory}} -b {{new_branch}}
```

- List all the working directories attached to this repository:

```
git worktree list
```

- Remove a worktree (after deleting worktree directory):

```
git worktree prune
```

git write-tree

Low level utility to create a tree object from the current index.

More information: https://git-scm.com/docs/git-write-tree.

- Create a tree object from the current index:

```
git write-tree
```

- Create a tree object without checking whether objects referenced by the directory exist in the object database:

```
git write-tree --missing-ok
```

- Create a tree object that represents a subdirectory (used to write the tree object for a subproject in the named subdirectory):

```
git write-tree --prefix {{subdirectory}}/
```

git

Distributed version control system.

Some subcommands such as **`commit`**, **`add`**, **`branch`**, **`switch`**, **`push`**, etc. have their own usage documentation.

More information: https://git-scm.com/docs/git.

- Create an empty Git repository:

```
git init
```

- Clone a remote Git repository from the internet:

```
git clone {{https://example.com/repo.git}}
```

- View the status of the local repository:

```
git status
```

- Stage all changes for a commit:

```
git add {{[-A|--all]}}
```

- Commit changes to version history:

```
git commit {{[-m|--message]}} {{message_text}}
```

- Push local commits to a remote repository:

```
git push
```

- Pull any changes made to a remote:

```
git pull
```

- Reset everything the way it was in the latest commit:

```
git reset --hard; git clean {{[-f|--force]}}
```

gitea

Administer Gitea, a lightweight Git hosting server.

Requires a configured **`app.ini`** file or environment variables.

More information: https://docs.gitea.com/administration/command-line.

- Run the Gitea web server using the default configuration:

```
gitea web
```

- Create the necessary database schema and tables:

```
gitea migrate
```

- Run administrative subcommands for user management or authentication management:

```
gitea admin {{user list}}
```

- Display help for a specific subcommand:

```
gitea {{admin}} --help
```

- Display help:

```
gitea help
```

- Display version:

```
gitea --version
```

github-label-sync

Synchronize GitHub labels.

More information: https://github.com/Financial-Times/github-label-sync.

- Synchronize labels using a local `labels.json` file:

```
github-label-sync --access-token {{token}}
{{repository_name}}
```

- Synchronize labels using a specific labels JSON file:

```
github-label-sync --access-token {{token}} --labels {{url|
path/to/json_file}} {{repository_name}}
```

- Perform a dry run instead of actually synchronizing labels:

```
github-label-sync --access-token {{token}} --dry-run
{{repository_name}}
```

- Keep labels that aren't in `labels.json`:

```
github-label-sync --access-token {{token}} --allow-added-
labels {{repository_name}}
```

- Synchronize using the `GITHUB_ACCESS_TOKEN` environment variable:

```
github-label-sync {{repository_name}}
```

gitk

Browse Git repositories graphically.

See also: **`git-gui`**, **`git-cola`**, **`tig`**.

More information: https://git-scm.com/docs/gitk.

- Show the repository browser for the current Git repository:

```
gitk
```

- Show repository browser for a specific file or directory:

```
gitk {{path/to/file_or_directory}}
```

- Show commits made since 1 week ago:

```
gitk --since="{{1 week ago}}"
```

- Show commits older than 1/1/2016:

```
gitk --until="{{1/1/2015}}"
```

- Show at most 100 changes in all branches:

```
gitk --max-count=100 --all
```

gitlab-ctl

Manage the GitLab omnibus.

More information: https://docs.gitlab.com/omnibus/maintenance/.

- Display the status of every service:

```
sudo gitlab-ctl status
```

- Display the status of a specific service:

```
sudo gitlab-ctl status {{nginx}}
```

- Restart every service:

```
sudo gitlab-ctl restart
```

- Restart a specific service:

```
sudo gitlab-ctl restart {{nginx}}
```

- Display the logs of every service and keep reading until `<Ctrl c>` is pressed:

```
sudo gitlab-ctl tail
```

- Display the logs of a specific service:

```
sudo gitlab-ctl tail {{nginx}}
```

- Send the SIGKILL signal to specific service:

```
sudo gitlab-ctl kill {{nginx}}
```

- Reconfigure the application:

```
sudo gitlab-ctl reconfigure
```

gitlab-runner

Manage GitLab runners.

More information: https://docs.gitlab.com/runner/.

- Register a runner:

```
sudo gitlab-runner register --url {{https://gitlab.example.com}} --registration-token {{token}} --name {{name}}
```

- Register a runner with a Docker executor:

```
sudo gitlab-runner register --url {{https://gitlab.example.com}} --registration-token {{token}} --name {{name}} --executor {{docker}}
```

- Unregister a runner:

```
sudo gitlab-runner unregister --name {{name}}
```

- Display the status of the runner service:

```
sudo gitlab-runner status
```

- Restart the runner service:

```
sudo gitlab-runner restart
```

- Check if the registered runners can connect to GitLab:

```
sudo gitlab-runner verify
```

gitlab

Ruby wrapper for the GitLab API.

Some subcommands such as **`ctl`** have their own usage documentation.

More information: https://narkoz.github.io/gitlab/.

- Create a new project:

```
gitlab create_project {{project_name}}
```

- Get info about a specific commit:

```
gitlab commit {{project_name}} {{commit_hash}}
```

- Get info about jobs in a CI pipeline:

```
gitlab pipeline_jobs {{project_name}} {{pipeline_id}}
```

- Start a specific CI job:

```
gitlab job_play {{project_name}} {{job_id}}
```

gitleaks

Detect secrets and API keys leaked in Git repositories.

More information: https://github.com/gitleaks/gitleaks.

- Scan a remote repository:

```
gitleaks detect --repo-url {{https://github.com/username/repository.git}}
```

- Scan a local directory:

```
gitleaks detect --source {{path/to/repository}}
```

- Output scan results to a JSON file:

```
gitleaks detect --source {{path/to/repository}} --report {{path/to/report.json}}
```

- Use a custom rules file:

```
gitleaks detect --source {{path/to/repository}} --config-path {{path/to/config.toml}}
```

- Start scanning from a specific commit:

```
gitleaks detect --source {{path/to/repository}} --log-opts {{--since=commit_id}}
```

- Scan uncommitted changes before a commit:

```
gitleaks protect --staged
```

- Display verbose output indicating which parts were identified as leaks during the scan:

```
gitleaks protect --staged --verbose
```

gitlint

Git commit message linter checks your commit messages for style.

More information: https://jorisroovers.com/gitlint/.

- Check the last commit message:

```
gitlint
```

- The range of commits to lint:

```
gitlint --commits {{single_refspec_argument}}
```

- Path to a directory or Python module with extra user-defined rules:

```
gitlint --extra-path {{path/to/directory}}
```

- Start a specific CI job:

```
gitlint --target {{path/to/target_directory}}
```

- Path to a file containing a commit-msg:

```
gitlint --msg-filename {{path/to/file}}
```

- Read staged commit meta-info from the local repository:

```
gitlint --staged
```

gitmoji

Interactively insert emojis on commits.

More information: https://github.com/carloscuesta/gitmoji-cli.

- Start the commit wizard:

```
gitmoji --commit
```

- Initialize the Git hook (so `gitmoji` will be run every time `git commit` is run):

```
gitmoji --init
```

- Remove the Git hook:

```
gitmoji --remove
```

- List all available emojis and their descriptions:

```
gitmoji --list
```

- Search emoji list for a list of keywords:

```
gitmoji --search {{keyword1}} {{keyword2}}
```

- Update cached list of emojis from main repository:

```
gitmoji --update
```

- Configure global preferences:

```
gitmoji --config
```

gitsome

A terminal-based interface for GitHub, accessed via the **`gh`** command.

It also provides menu-style autocomplete suggestions for **`git`** commands.

More information: https://github.com/donnemartin/gitsome.

- Enter the gitsome shell (optional), to enable autocompletion and interactive help for Git (and gh) commands:

```
gitsome
```

- Setup GitHub integration with the current account:

```
gh configure
```

- List notifications for the current account (as would be seen in https://github.com/notifications):

```
gh notifications
```

- List the current account's starred repos, filtered by a given search string:

```
gh starred "{{python 3}}"
```

- View the recent activity feed of a given GitHub repository:

```
gh feed {{tldr-pages/tldr}}
```

- View the recent activity feed for a given GitHub user, using the default pager (e.g. `less`):

```
gh feed {{torvalds}} -p
```

gitstats

Git repository statistics generator.

More information: https://gitstats.sourceforge.net.

- Generate statistics for a local repository:

```
gitstats {{path/to/git_repo/.git}} {{path/to/output_folder}}
```

- View generated statistics in a web browser on Windows (PowerShell)/macOS/Linux:

```
{{Invoke-Item|open|xdg-open}} {{path/to/output_folder/index.html}}
```

gitui

A lightweight keyboard-only TUI for Git.

See also: **`tig`**, **`git-gui`**.

More information: https://github.com/extrawurst/gitui.

- Specify the color theme (defaults to `theme.ron`):

```
gitui --theme {{theme}}
```

- Store logging output into a cache directory:

```
gitui --logging
```

- Use notify-based file system watcher instead of tick-based update:

```
gitui --watcher
```

- Generate a bug report:

```
gitui --bugreport
```

- Use a specific Git directory:

```
gitui --directory {{path/to/directory}}
```

- Use a specific working directory:

```
gitui --workdir {{path/to/directory}}
```

- Display help:

```
gitui --help
```

- Display version:

```
gitui --version
```

gitwatch

Automatically commit file or directory changes to a Git repository.

More information: https://github.com/gitwatch/gitwatch.

- Automatically commit any changes made to a file or directory:

```
gitwatch {{path/to/file_or_directory}}
```

- Automatically commit changes and push them to a remote repository:

```
gitwatch -r {{remote_name}} {{path/to/file_or_directory}}
```

- Automatically commit changes and push them to a specific branch of a remote repository:

```
gitwatch -r {{remote_name}} -b {{branch_name}} {{path/to/file_or_directory}}
```

gixy

Analyze nginx configuration files.

More information: https://github.com/yandex/gixy.

- Analyze nginx configuration (default path: `/etc/nginx/nginx.conf`):

```
gixy
```

- Analyze nginx configuration but skip specific tests:

```
gixy --skips {{http_splitting}}
```

- Analyze nginx configuration with the specific severity level:

```
gixy {{-l|-ll|-lll}}
```

- Analyze nginx configuration files on the specific path:

```
gixy {{path/to/configuration_file_1}} {{path/to/
configuration_file_2}}
```

glab alias

Manage GitLab CLI command aliases.

More information: https://gitlab.com/gitlab-org/cli/-/blob/main/docs/source/alias/index.md.

- List all the aliases `glab` is configured to use:

```
glab alias list
```

- Create a `glab` subcommand alias:

```
glab alias set {{mrv}} '{{mr view}}'
```

- Set a shell command as a `glab` subcommand:

```
glab alias set {{[-s|--shell]}} {{alias_name}} {{command}}
```

- Delete a command shortcut:

```
glab alias delete {{alias_name}}
```

- Display the subcommand help:

```
glab alias
```

glab auth

Authenticate with a GitLab host.

More information: https://gitlab.com/gitlab-org/cli/-/blob/main/docs/source/auth/index.md.

- Log in with interactive prompt:

```
glab auth login
```

- Log in with a token:

```
glab auth login {{[-t|--token]}} {{token}}
```

- Check authentication status:

```
glab auth status
```

- Log in to a specific GitLab instance:

```
glab auth login {{[-h|--hostname]}} {{gitlab.example.com}}
```

glab issue

Manage GitLab issues.

More information: https://gitlab.com/gitlab-org/cli/-/blob/main/docs/source/issue/index.md.

- Display a specific issue:

```
glab issue view {{issue_number}}
```

- Display a specific issue in the default web browser:

```
glab issue view {{issue_number}} {{[-w|--web]}}
```

- Create a new issue in the default web browser:

```
glab issue create --web
```

- List the last 10 issues with the bug label:

```
glab issue list {{[-P|--per-page]}} {{10}} {{[-l|--label]}} "{{bug}}"
```

- List closed issues made by a specific user:

```
glab issue list {{[-c|--closed]}} --author {{username}}
```

- Reopen a specific issue:

```
glab issue reopen {{issue_number}}
```

glab mr create

Manage GitLab merge requests.

More information: https://gitlab.com/gitlab-org/cli/-/blob/main/docs/source/mr/create.md.

- Interactively create a merge request:

```
glab mr create
```

- Create a merge request, determining the title and description from the commit messages of the current branch:

```
glab mr create {{[-f|--fill]}}
```

- Create a draft merge request:

```
glab mr create --draft
```

- Create a merge request specifying the target branch, title, and description:

```
glab mr create {{[-b|--target-branch]}} {{target_branch}}
{{[-t|--title]}} "{{title}}" {{[-d|--description]}}
"{{description}}"
```

- Start opening a merge request in the default web browser:

```
glab mr create {{[-w|--web]}}
```

glab mr merge

Merge GitLab merge requests.

More information: https://gitlab.com/gitlab-org/cli/-/blob/main/docs/source/mr/merge.md.

- Merge the merge request associated with the current branch interactively:

```
glab mr merge
```

- Merge the specified merge request, interactively:

```
glab mr merge {{mr_number}}
```

- Merge the merge request, removing the branch on both the local and the remote:

```
glab mr merge {{[-d|--remove-source-branch]}}
```

- Squash the current merge request into one commit with the message body and merge:

```
glab mr merge {{[-s|--squash]}} {{[-m|--message]}} "{{commit_message_body}}"
```

- Display help:

```
glab mr merge --help
```

glab mr

Manage GitLab merge requests.

Some subcommands such as **`create`** have their own usage documentation.

More information: https://gitlab.com/gitlab-org/cli/-/blob/main/docs/source/mr/index.md.

- Create a merge request:

```
glab mr create
```

- Check out a specific merge request locally:

```
glab mr checkout {{mr_number}}
```

- View the changes made in the merge request:

```
glab mr diff
```

- Approve the merge request for the current branch:

```
glab mr approve
```

- Merge the merge request associated with the current branch interactively:

```
glab mr merge
```

- Edit a merge request interactively:

```
glab mr update
```

- Edit the target branch of a merge request:

```
glab mr update --target-branch {{branch_name}}
```

glab pipeline

List, view, and run GitLab CI/CD pipelines.

More information: https://gitlab.com/gitlab-org/cli/-/blob/main/docs/source/index.md.

- View a running pipeline on the current branch:

```
glab pipeline status
```

- View a running pipeline on a specific branch:

```
glab pipeline status --branch {{branch_name}}
```

- Get the list of pipelines:

```
glab pipeline list
```

- Run a manual pipeline on the current branch:

```
glab pipeline run
```

- Run a manual pipeline on a specific branch:

```
glab pipeline run --branch {{branch_name}}
```

glab release

Manage GitLab releases.

More information: https://gitlab.com/gitlab-org/cli/-/blob/main/docs/source/release/index.md.

- List releases in a Gitlab repository, limited to 30 items:

```
glab release list
```

- Display information about a specific release:

```
glab release view {{tag}}
```

- Create a new release:

```
glab release create {{tag}}
```

- Delete a specific release:

```
glab release delete {{tag}}
```

- Download assets from a specific release:

```
glab release download {{tag}}
```

- Upload assets to a specific release:

```
glab release upload {{tag}} {{path/to/file1 path/to/file2 ...}}
```

glab repo

Work with GitLab repositories.

More information: https://gitlab.com/gitlab-org/cli/-/blob/main/docs/source/repo/index.md.

- Create a new repository (if the repository name is not set, the default name will be the name of the current directory):

```
glab repo create {{name}}
```

- Clone a repository:

```
glab repo clone {{owner}}/{{repository}}
```

- Fork and clone a repository:

```
glab repo fork {{owner}}/{{repository}} {{[-c|--clone]}}
```

- View a repository in the default web browser:

```
glab repo view {{owner}}/{{repository}} {{[-w|--web]}}
```

- Search some repositories in the GitLab instance:

```
glab repo search {{[-s|--search]}} {{search_string}}
```

glab

Work seamlessly with GitLab.

Some subcommands such as **`config`** have their own usage documentation.

More information: https://gitlab.com/gitlab-org/cli/-/tree/main/docs/source.

- Clone a GitLab repository locally:

```
glab repo clone {{owner}}/{{repository}}
```

- Create a new issue:

```
glab issue create
```

- View and filter the open issues of the current repository:

```
glab issue list
```

- View an issue in the default browser:

```
glab issue view {{[-w|--web]}} {{issue_number}}
```

- Create a merge request:

```
glab mr create
```

- View a pull request in the default web browser:

```
glab mr view {{[-w|--web]}} {{pr_number}}
```

- Check out a specific pull request locally:

```
glab mr checkout {{pr_number}}
```

gladtex

A LaTeX formula preprocessor for HTML files.

It converts LaTeX formulas to images.

More information: https://manned.org/gladtex.1.

- Convert to HTML:

```
gladtex {{path/to/input.htex}}
```

- Save the converted file to a specific [o]utput file:

```
gladtex {{path/to/input.htex}} -o {{path/to/output.html}}
```

- Save the generated images to a specific [d]irectory:

```
gladtex {{path/to/input.htex}} -d {{path/to/
image_output_directory}}
```

- Set image [r]esolution (in dpi, default is 100):

```
gladtex {{path/to/input.htex}} -r {{resolution}}
```

- [k]eep LaTeX files after conversion:

```
gladtex {{path/to/input.htex}} -k
```

- Set [b]ackground and [f]oreground color of the images:

```
gladtex {{path/to/input.htex}} -b {{background_color}} -f
{{foreground_color}}
```

- Convert Markdown to HTML using `pandoc` and `gladtex`:

```
pandoc {{[-s|--standalone]}} {{[-t|--to]}} html --gladtex
{{path/to/input.md}} | gladtex -o {{path/to/output.html}}
```

glances

A cross-platform system monitoring tool.

See also: **`htop`**, **`atop`**, **`top`**, **`btm`**, **`btop`**.

More information: https://nicolargo.github.io/glances/.

- Run in terminal:

```
glances
```

- Run in web server mode to show results in browser:

```
glances {{[-w|--webserver]}}
```

- Run in server mode to allow connections from other Glances clients:

```
glances {{[-s|--server]}}
```

- Connect to a Glances server:

```
glances {{[-c|--client]}} {{hostname}}
```

- Require a password in (web) server mode:

```
glances {{[-s|--server]}} --password
```

- Exit Glances:

```
<q>
```

- Display help:

```
glances {{[-h|--help]}}
```

gleam

The compiler, build tool, package manager and code formatter for Gleam, "a friendly language for building type-safe systems that scale!".

More information: https://gleam.run/writing-gleam/command-line-reference/.

- Create a new gleam project:

  ```
  gleam new {{project_name}}
  ```

- Build and run a gleam project:

  ```
  gleam run
  ```

- Build the project:

  ```
  gleam build
  ```

- Run a project for a particular platform and runtime:

  ```
  gleam run --target {{platform}} --runtime {{runtime}}
  ```

- Add a hex dependency to your project:

  ```
  gleam add {{dependency_name}}
  ```

- Run project tests:

  ```
  gleam test
  ```

- Format source code:

  ```
  gleam format
  ```

- Type check the project:

  ```
  gleam check
  ```

glib-compile-resources

Compiles resource files (e.g. images) into a binary resource bundle.

These may be linked into GTK applications using the GResource API.

More information: https://manned.org/glib-compile-resources.

- Compile resources referenced in `file.gresource.xml` to a .gresource binary:

```
glib-compile-resources {{file.gresource.xml}}
```

- Compile resources referenced in `file.gresource.xml` to a C source file:

```
glib-compile-resources --generate-source
{{file.gresource.xml}}
```

- Compile resources in `file.gresource.xml` to a chosen target file, with `.c`, `.h` or `.gresource` extension:

```
glib-compile-resources --generate --target={{file.ext}}
{{file.gresource.xml}}
```

- Print a list of resource files referenced in `file.gresource.xml`:

```
glib-compile-resources --generate-dependencies
{{file.gresource.xml}}
```

glow

Render Markdown in the terminal.

More information: https://github.com/charmbracelet/glow.

- Run glow and select a file to view:

```
glow
```

- Render a Markdown file to the terminal:

```
glow {{path/to/file}}
```

- View a Markdown file using a paginator:

```
glow {{[-p|--pager]}} {{path/to/file}}
```

- View a file from a URL:

```
glow {{https://example.com/file.md}}
```

- View a GitHub/GitLab README:

```
glow {{github.com/owner/repository}}
```

gml2gv

Convert a graph from **`gml`** to **`gv`** format.

Converters: **`gml2gv`**, **`gv2gml`**, **`gv2gxl`**, **`gxl2gv`**, **`graphml2gv`** & **`mm2gv`**.

More information: https://graphviz.org/pdf/gml2gv.1.pdf.

- Convert a graph from `gml` to `gv` format:

```
gml2gv -o {{output.gv}} {{input.gml}}
```

- Convert a graph using `stdin` and `stdout`:

```
cat {{input.gml}} | gml2gv > {{output.gv}}
```

- Display help:

```
gml2gv -?
```

gmssl

GmSSL is a crypto toolkit supporting SM1, SM2, SM3, SM4, SM9, and ZUC/ZUC256.

More information: http://gmssl.org/english.html.

- Generate an SM3 hash for a file:

```
gmssl sm3 {{path/to/file}}
```

- Encrypt a file using the SM4 cipher:

```
gmssl sms4 -e -in {{path/to/file}} -out {{path/to/file.sms4}}
```

- Decrypt a file using the SM4 cipher:

```
gmssl sms4 -d -in {{path/to/file.sms4}}
```

- Generate an SM2 private key:

```
gmssl sm2 -genkey -out {{path/to/file.pem}}
```

- Generate an SM2 public key from an existing private key:

```
gmssl sm2 -pubout -in {{path/to/file.pem}} -out {{path/to/
file.pem.pub}}
```

- Encrypt a file using the ZUC cipher:

```
gmssl zuc -e -in {{path/to/file}} -out {{path/to/file.zuc}}
```

- Decrypt a file using the ZUC cipher:

```
gmssl zuc -d -in {{path/to/file.zuc}}
```

- Display version:

```
gmssl version
```

gnatmake

A low-level build tool for Ada programs (part of the GNAT toolchain).

More information: https://gcc.gnu.org/onlinedocs/gnat_ugn/Switches-for-gnatmake.html.

- Compile an executable:

```
gnatmake {{source_file1.adb source_file2.adb ...}}
```

- Set a custom [o]utput executable name:

```
gnatmake -o {{executable_name}} {{source_file.adb}}
```

- [f]orce recompilation:

```
gnatmake -f {{source_file.adb}}
```

gnatprep

Preprocessor for Ada source code files (part of the GNAT toolchain).

More information: https://gcc.gnu.org/onlinedocs/gnat_ugn/Preprocessing-with-gnatprep.html.

- Use symbol definitions from a file:

```
gnatprep {{source_file}} {{target_file}} {{definitions_file}}
```

- Specify symbol values in the command-line:

```
gnatprep -D{{name}}={{value}} {{source_file}} {{target_file}}
```

gnmic get

Get a snapshot of a gnmi network device operation data.

More information: https://gnmic.kmrd.dev/cmd/get.

- Get a snapshot of the device state at a specific path:

```
gnmic {{[-a|--address]}} {{ip:port}} get --path {{path}}
```

- Query the device state at multiple paths:

```
gnmic {{[-a|--address]}} {{ip:port}} get --path {{path/to/
file_or_directory1}} --path {{path/to/file_or_directory2}}
```

- Query the device state at multiple paths with a common prefix:

```
gnmic {{[-a|--address]}} {{ip:port}} get --prefix {{prefix}}
--path {{path/to/file_or_directory1}} --path {{path/to/
file_or_directory2}}
```

- Query the device state and specify response encoding (json_ietf):

```
gnmic {{[-a|--address]}} {{ip:port}} get --path {{path}} {{[-
e|--encoding]}} json_ietf
```

gnmic set

Modify gnmi network device configuration.

More information: https://gnmic.kmrd.dev/cmd/set.

- Update the value of a path:

```
gnmic {{[-a|--address]}} {{ip:port}} set --update-path
{{path/to/directory}} --update-value {{value}}
```

- Update the value of a path to match the contents of a JSON file:

```
gnmic {{[-a|--address]}} {{ip:port}} set --update-path
{{path/to/directory}} --update-file {{path/to/file}}
```

- Replace the value of a path to match the contents of a JSON file:

```
gnmic {{[-a|--address]}} {{ip:port}} set --replace-path
{{path/to/directory}} --replace-file {{path/to/file}}
```

- Delete the node at a given path:

```
gnmic {{[-a|--address]}} {{ip:port}} set --delete {{path/to/
directory}}
```

gnmic sub

This command is an alias of **`gnmic subscribe`**.

- View documentation for the original command:

```
tldr gnmic subscribe
```

gnmic subscribe

Subscribe to a gnmic network device state updates.

More information: https://gnmic.kmrd.dev/cmd/subscribe.

- Subscribe to target state updates under the subtree of a specific path:

```
gnmic {{[-a|--address]}} {{ip:port}} subscribe --path
{{path}}
```

- Subscribe to a target with a sample interval of 30s (default is 10s):

```
gnmic {{[-a|--address]}} {{ip:port}} subscribe --path
{{path}} --sample-interval 30s
```

- Subscribe to a target with sample interval and updates only on change:

```
gnmic {{[-a|--address]}} {{ip:port}} subscribe --path
{{path}} --stream-mode on-change --heartbeat-interval {{1m}}
```

- Subscribe to a target for only one update:

```
gnmic {{[-a|--address]}} {{ip:port}} subscribe --path
{{path}} --mode once
```

- Subscribe to a target and specify response encoding (json_ietf):

```
gnmic {{[-a|--address]}} {{ip:port}} subscribe --path
{{path}} {{[-e|--encoding]}} json_ietf
```

gnmic

A gNMI client.

Manage gNMI network device configuration and view operational data.

More information: https://gnmic.kmrd.dev.

- Request device capabilities:

```
gnmic {{[-a|--address]}} {{ip:port}} capabilities
```

- Provide a username and password to fetch device capabilities:

```
gnmic {{[-a|--address]}} {{ip:port}} {{[-u|--username]}}
{{username}} {{[-p|--password]}} {{password}} capabilities
```

- Get a snapshot of the device state at a specific path:

```
gnmic {{[-a|--address]}} {{ip:port}} get --path {{path}}
```

- Update device state at a specific path:

```
gnmic {{[-a|--address]}} {{ip:port}} set --update-path
{{path}} --update-value {{value}}
```

- Subscribe to target state updates under the subtree at a specific path:

```
gnmic {{[-a|--address]}} {{ip:port}} subscribe --path
{{path}}
```

gnomon

Utility to annotate console logging statements with timestamps and find slow processes.

More information: https://github.com/paypal/gnomon.

- Use UNIX (or DOS) pipes to pipe `stdout` of any command through gnomon:

```
{{npm test}} | gnomon
```

- Show number of seconds since the start of the process:

```
{{npm test}} | gnomon --type=elapsed-total
```

- Show an absolute timestamp in UTC:

```
{{npm test}} | gnomon --type=absolute
```

- Use a high threshold of 0.5 seconds, exceeding which the timestamp will be colored bright red:

```
{{npm test}} | gnomon --high 0.5
```

- Use a medium threshold of 0.2 seconds, exceeding which the timestamp will be colored bright yellow:

```
{{npm test}} | gnomon --medium {{0.2}}
```

gnucash-cli

A command-line version of GnuCash.

More information: https://gnucash.org/viewdoc.phtml?rev=5&lang=C&doc=help.

- Get quotes for currencies and stocks specified in a file and print them:

```
gnucash-cli {{[-Q|--quotes]}} get {{path/to/file.gnucash}}
```

- Generate a financial report of a specific type, specified by `--name`:

```
gnucash-cli --report run --name "{{Balance Sheet}}" {{path/to/file.gnucash}}
```

gnucash

Personal and small-business financial-accounting software.

More information: https://gnucash.org/viewdoc.phtml.

- Launch GnuCash and load the previously opened file:

```
gnucash
```

- Launch GnuCash and load the specified file:

```
gnucash {{path/to/file.gnucash}}
```

- Launch GnuCash and load an empty file:

```
gnucash --nofile
```

gnuplot

A graph plotter that outputs in several formats.

More information: https://manned.org/gnuplot.

- Start the interactive graph plotting shell:

```
gnuplot
```

- Plot the graph for the specified graph definition file:

```
gnuplot {{path/to/definition.plt}}
```

- Set the output format by executing a command before loading the definition file:

```
gnuplot -e "{{set output "path/to/file.png" size
1024,768}}" {{path/to/definition.plt}}
```

- Persist the graph plot preview window after gnuplot exits:

```
gnuplot {{[-p|--persist]}} {{path/to/definition.plt}}
```

go bug

Report a bug.

More information: https://pkg.go.dev/cmd/go#hdr-Start_a_bug_report.

- Open a web page to start a bug report:

```
go bug
```

go build

Compile Go sources.

More information: https://pkg.go.dev/cmd/go#hdr-Compile_packages_and_dependencies.

- Compile a 'package main' file (output will be the filename without extension):

```
go build {{path/to/main.go}}
```

- Compile, specifying the output filename:

```
go build -o {{path/to/binary}} {{path/to/source.go}}
```

- Compile a package:

```
go build -o {{path/to/binary}} {{path/to/package}}
```

- Compile a main package into an executable, enabling data race detection:

```
go build -race -o {{path/to/executable}} {{path/to/
main_package}}
```

go clean

Remove object files and cached files.

More information: https://pkg.go.dev/cmd/go#hdr-Remove_object_files_and_cached_files.

- Print the remove commands instead of actually removing anything:

```
go clean -n
```

- Delete the build cache:

```
go clean -cache
```

- Delete all cached test results:

```
go clean -testcache
```

- Delete the module cache:

```
go clean -modcache
```

go doc

View documentation for a package or symbol.

More information: https://pkg.go.dev/cmd/go#hdr-Show_documentation_for_package_or_symbol.

- View documentation for the current package:

```
go doc
```

- Show package documentation and exported symbols:

```
go doc {{encoding/json}}
```

- Show also documentation of symbols:

```
go doc -all {{encoding/json}}
```

- Show also sources:

```
go doc -all -src {{encoding/json}}
```

- Show a specific symbol:

```
go doc -all -src {{encoding/json.Number}}
```

go env

Manage environment variables used by the Go toolchain.

More information: https://pkg.go.dev/cmd/go#hdr-Print_Go_environment_information.

- Show all environment variables:

```
go env
```

- Show a specific environment variable:

```
go env {{GOPATH}}
```

- Set an environment variable to a value:

```
go env -w {{GOBIN}}={{path/to/directory}}
```

- Reset an environment variable's value:

```
go env -u {{GOBIN}}
```

go fix

Update packages to use new APIs.

More information: https://pkg.go.dev/cmd/go#hdr-Update_packages_to_use_new_APIs.

- Update packages to use new APIs:

```
go fix {{packages}}
```

go fmt

Format Go source files, printing the changed filenames.

More information: https://pkg.go.dev/cmd/go#hdr-Gofmt__reformat__package_sources.

- Format Go source files in the current directory:

```
go fmt
```

- Format a specific Go package in your import path (`$GOPATH/src`):

```
go fmt {{path/to/package}}
```

- Format the package in the current directory and all subdirectories (note the `...`):

```
go fmt {{./...}}
```

- Print what format commands would've been run, without modifying anything:

```
go fmt -n
```

- Print which format commands are run as they are run:

```
go fmt -x
```

go generate

Generate Go files by running commands within source files.

More information: https://pkg.go.dev/cmd/go#hdr-Generate_Go_files_by_processing_source.

- Generate Go files by running commands within source files:

```
go generate
```

go get

Add a dependency package, or download packages in legacy GOPATH mode.

More information: https://pkg.go.dev/cmd/go#hdr-Add_dependencies_to_current_module_and_install_them.

- Add a specified package to `go.mod` in module-mode or install the package in GOPATH-mode:

  ```
  go get {{example.com/pkg}}
  ```

- Modify the package with a given version in module-aware mode:

  ```
  go get {{example.com/pkg}}@{{v1.2.3}}
  ```

- Remove a specified package:

  ```
  go get {{example.com/pkg}}@{{none}}
  ```

go install

Compile and install packages named by the import paths.

More information: https://pkg.go.dev/cmd/go#hdr-Compile_and_install_packages_and_dependencies.

- Compile and install the current package:

```
go install
```

- Compile and install a specific local package:

```
go install {{path/to/package}}
```

- Install the latest version of a program, ignoring `go.mod` in the current directory:

```
go install {{golang.org/x/tools/gopls}}@{{latest}}
```

- Install a program at the version selected by `go.mod` in the current directory:

```
go install {{golang.org/x/tools/gopls}}
```

go list

List packages or modules.

More information: https://pkg.go.dev/cmd/go#hdr-List_packages_or_modules.

- List packages:

```
go list ./...
```

- List standard packages:

```
go list std
```

- List packages in JSON format:

```
go list -json time net/http
```

- List module dependencies and available updates:

```
go list -m -u all
```

go mod

Module maintenance.

More information: https://pkg.go.dev/cmd/go#hdr-Module_maintenance.

- Initialize new module in current directory:

```
go mod init {{moduleName}}
```

- Download modules to local cache:

```
go mod download
```

- Add missing and remove unused modules:

```
go mod tidy
```

- Verify dependencies have expected content:

```
go mod verify
```

- Copy sources of all dependencies into the vendor directory:

```
go mod vendor
```

go run

Compile and run Go code without saving a binary.

More information: https://pkg.go.dev/cmd/go#hdr-Compile_and_run_Go_program.

- Run a Go file:

```
go run {{path/to/file.go}}
```

- Run a main Go package:

```
go run {{path/to/package}}
```

go test

Tests Go packages (files have to end with **`_test.go`**).

More information: https://pkg.go.dev/cmd/go#hdr-Testing_flags.

- Test the package found in the current directory:

```
go test
```

- [v]erbosely test the package in the current directory:

```
go test -v
```

- Test the packages in the current directory and all subdirectories (note the `...`):

```
go test -v ./...
```

- Test the package in the current directory and run all benchmarks:

```
go test -v -bench .
```

- Test the package in the current directory and run all benchmarks for 50 seconds:

```
go test -v -bench . -benchtime 50s
```

- Test the package with coverage analysis:

```
go test -cover
```

go tool

Run a Go tool or command.

Execute a Go command as a stand-alone binary, typically for debugging.

More information: https://pkg.go.dev/cmd/go#hdr-Run_specified_go_tool.

- List available tools:

```
go tool
```

- Run the go link tool:

```
go tool link {{path/to/main.o}}
```

- Print the command that would be executed, but do not execute it (similar to `whereis`):

```
go tool -n {{command}} {{arguments}}
```

- View documentation for a specified tool:

```
go tool {{command}} --help
```

- List all available cross-compilation targets:

```
go tool dist list
```

go version

Display Go version.

More information: https://pkg.go.dev/cmd/go#hdr-Print_Go_version.

- Display version:

```
go version
```

- Display the Go version used to build a specific executable file:

```
go version {{path/to/executable}}
```

go vet

Check Go source code and report suspicious constructs (e.g. lint your Go source files).

Go vet returns a non-zero exit code if problems are found; returns a zero exit code if no problems are found.

More information: https://pkg.go.dev/cmd/vet.

- Check the Go package in the current directory:

```
go vet
```

- Check the Go package in the specified path:

```
go vet {{path/to/file_or_directory}}
```

- List available checks that can be run with go vet:

```
go tool vet help
```

- View details and flags for a particular check:

```
go tool vet help {{check_name}}
```

- Display offending lines plus n lines of surrounding context:

```
go vet -c={{n}}
```

- Output analysis and errors in JSON format:

```
go vet -json
```

go

Manage Go source code.

Some subcommands such as **`build`** have their own usage documentation.

More information: https://pkg.go.dev/cmd/go.

- Download and install a package, specified by its import path:

```
go get {{path/to/package}}
```

- Compile and run a source file (it has to contain a `main` package):

```
go run {{file}}.go
```

- Compile a source file into a named executable:

```
go build -o {{executable}} {{file}}.go
```

- Compile the package present in the current directory:

```
go build
```

- Execute all test cases of the current package (files have to end with `_test.go`):

```
go test
```

- Compile and install the current package:

```
go install
```

- Initialize a new module in the current directory:

```
go mod init {{module_name}}
```

gobuster

Brute-forces hidden paths on web servers and more.

More information: https://github.com/OJ/gobuster#modes.

- Discover directories and files that match in the wordlist:

```
gobuster dir {{[-u|--url]}} {{https://example.com/}} {{[-w|--wordlist]}} {{path/to/file}}
```

- Discover subdomains:

```
gobuster dns {{[-d|--domain]}} {{example.com}} {{[-w|--wordlist]}} {{path/to/file}}
```

- Discover Amazon S3 buckets:

```
gobuster s3 {{[-w|--wordlist]}} {{path/to/file}}
```

- Discover other virtual hosts on the server:

```
gobuster vhost {{[-u|--url]}} {{https://example.com/}} {{[-w|--wordlist]}} {{path/to/file}}
```

- Fuzz the value of a parameter:

```
gobuster fuzz {{[-u|--url]}} {{https://example.com/?parameter=FUZZ}} {{[-w|--wordlist]}} {{path/to/file}}
```

- Fuzz the name of a parameter:

```
gobuster fuzz {{[-u|--url]}} {{https://example.com/?FUZZ=value}} {{[-w|--wordlist]}} {{path/to/file}}
```

gocr

Optical Character Recognition tool.

Recognize characters using its engine, and prompt the user for unknown patterns to store them in a database.

More information: https://manned.org/gocr.1.

- Recognize characters in the [i]nput image and [o]utput it in the given file. Put the database ([p]) in `path/to/db_directory` (verify that the folder exists or DB usage will silently be skipped). [m]ode 130 means create + use + extend database:

```
gocr -m 130 -p {{path/to/db_directory}} -i {{path/to/
input_image.png}} -o {{path/to/output_file.txt}}
```

- Recognize characters and assume all [C]haracters are numbers:

```
gocr -m 130 -p {{path/to/db_directory}} -i {{path/to/
input_image.png}} -o {{path/to/output_file.txt}} -C
"{{0..9}}"
```

- Recognize characters with a cert[a]inty of 100% (characters have a higher chance to be treated as unknown):

```
gocr -m 130 -p {{path/to/db_directory}} -i {{path/to/
input_image.png}} -o {{path/to/output_file.txt}} -a 100
```

gocryptfs

Encrypted overlay filesystem written in Go.

More information: https://github.com/rfjakob/gocryptfs#use.

- Initialize an encrypted filesystem:

```
gocryptfs -init {{path/to/cipher_directory}}
```

- Mount an encrypted filesystem:

```
gocryptfs {{path/to/cipher_directory}} {{path/to/
mount_point}}
```

- Mount with the explicit master key instead of password:

```
gocryptfs --masterkey {{path/to/cipher_directory}} {{path/to/
mount_point}}
```

- Change the password:

```
gocryptfs --passwd {{path/to/cipher_directory}}
```

- Make an encrypted snapshot of a plain directory:

```
gocryptfs --reverse {{path/to/plain_directory}} {{path/to/
cipher_directory}}
```

godoc

View documentation for go packages.

More information: https://pkg.go.dev/golang.org/x/tools/cmd/godoc.

- Display help for a specific package:

```
godoc {{fmt}}
```

- Display help for the function "Printf" of "fmt" package:

```
godoc {{fmt}} {{Printf}}
```

- Serve documentation as a web server on port 6060:

```
godoc -http=:{{6060}}
```

- Create an index file:

```
godoc -write_index -index_files={{path/to/file}}
```

- Use the given index file to search the docs:

```
godoc -http=:{{6060}} -index -index_files={{path/to/file}}
```

godot

An open source 2D and 3D game engine.

More information: https://docs.godotengine.org/en/latest/tutorials/editor/command_line_tutorial.html.

- Run a project if the current directory contains a `project.godot` file, otherwise open the project manager:

  ```
  godot
  ```

- Edit a project (the current directory must contain a `project.godot` file):

  ```
  godot {{[-e|--editor]}}
  ```

- Open the project manager even if the current directory contains a `project.godot` file:

  ```
  godot {{[-p|--project-manager]}}
  ```

- Export a project for release using a given export preset (the preset must be defined in the project):

  ```
  godot --export-release {{preset}} {{output_path}}
  ```

- Execute a standalone GDScript file (the script must inherit from `SceneTree` or `MainLoop`):

  ```
  godot {{[-s|--script]}} {{script.gd}}
  ```

goenv

Install, uninstall or switch between Golang versions.

Supports version numbers like "1.16.15" or "1.22.8" etc.

More information: https://github.com/go-nv/goenv.

- List all available goenv commands:

```
goenv commands
```

- Install a specific version of Golang:

```
goenv install {{go_version}}
```

- Use a specific version of Golang in the current project:

```
goenv local {{go_version}}
```

- Set the default Golang version:

```
goenv global {{go_version}}
```

- List all available Golang versions and highlight the default one:

```
goenv versions
```

- Uninstall a given Go version:

```
goenv uninstall {{go_version}}
```

- Run an executable with the selected Go version:

```
goenv exec go run {{go_version}}
```

gofmt

Format Go source code.

More information: https://pkg.go.dev/cmd/gofmt.

- Format a file and display the result to the console:

```
gofmt {{source.go}}
```

- Format a file, overwriting the original file in-place:

```
gofmt -w {{source.go}}
```

- Format a file, and then simplify the code, overwriting the original file:

```
gofmt -s -w {{source.go}}
```

- Print all (including spurious) errors:

```
gofmt -e {{source.go}}
```

goimports

Update Go import lines, adding missing ones and removing unreferenced ones.

More information: https://pkg.go.dev/golang.org/x/tools/cmd/goimports.

- Display the completed import source file:

```
goimports {{path/to/file.go}}
```

- Write the result back to the source file instead of `stdout`:

```
goimports -w {{path/to/file.go}}
```

- Display diffs and write the result back to the source file:

```
goimports -w -d {{path/to/file.go}}
```

- Set the import prefix string after 3rd-party packages (comma-separated list):

```
goimports -local {{path/to/package1,path/to/package2,...}}
{{path/to/file.go}}
```

golangci-lint

Parallelized, smart and fast Go linters runner that integrates with all major IDEs and supports YAML configuration.

More information: https://golangci-lint.run/welcome/quick-start/.

- Run linters in the current folder:

  ```
  golangci-lint run
  ```

- List enabled and disabled linters (Note: Disabled linters are shown last, do not mistake them for enabled ones):

  ```
  golangci-lint linters
  ```

- Enable a specific linter for this run:

  ```
  golangci-lint run {{[-E|--enable]}} {{linter}}
  ```

gomi

Manage the trashcan.

See also: **`trash`**, **`rm`**.

More information: https://github.com/babarot/gomi.

- Safely delete specific files or folders:

```
gomi {{path/to/file1 path/to/file2 path/to/folder1 path/to/folder2 ...}}
```

- Open an interactive menu to restore one or more files:

```
gomi {{[-b|--restore]}}
```

- Remove files that have been in the trashcan longer than the specified time ([d]ay, [w]eek, [m]onth, [y]ear):

```
gomi --prune {{1d|1w|1m|1y|...}}
```

- Remove orphaned `.trashinfo` files:

```
gomi --prune {{orphans}}
```

google-chrome

This command is an alias of **`chromium`**.

More information: https://chrome.google.com.

- View documentation for the original command:

```
tldr chromium
```

googler

Search Google from the command-line.

More information: https://github.com/jarun/googler.

- Search Google for a keyword:

```
googler {{keyword}}
```

- Search Google and open the first result in web browser:

```
googler {{[-j|--first]}} {{keyword}}
```

- Show n search results (default: 10):

```
googler {{[-n|--count]}} {{n}} {{keyword}}
```

- Disable automatic spelling correction:

```
googler {{[-x|--exact]}} {{keyword}}
```

- Search one site for a keyword:

```
googler {{[-w|--site]}} {{site}} {{keyword}}
```

- Show Google search result in JSON format:

```
googler --json {{keyword}}
```

- Perform in-place self-upgrade:

```
googler {{[-u|--upgrade]}}
```

- Display help in interactive mode:

```
<?>
```

gopass

Standard Unix Password Manager for Teams. Written in Go.

More information: https://www.gopass.pw.

- Initialize the configuration settings:

```
gopass init
```

- Create a new entry:

```
gopass new
```

- Show all stores:

```
gopass mounts
```

- Mount a shared Git store:

```
gopass mounts add {{store_name}} {{git_repo_url}}
```

- Search interactively using a keyword:

```
gopass show {{keyword}}
```

- Search using a keyword:

```
gopass find {{keyword}}
```

- Sync all mounted stores:

```
gopass sync
```

- Show a particular password entry:

```
gopass {{store_name|path/to/directory|email@email.com}}
```

gops

List and diagnose Go processes currently running on your system.

More information: https://github.com/google/gops.

- Print all go processes running locally:

```
gops
```

- Print more information about a process:

```
gops {{pid}}
```

- Display a process tree:

```
gops tree
```

- Print the current stack trace from a target program:

```
gops stack {{pid|addr}}
```

- Print the current runtime memory statistics:

```
gops memstats {{pid|addr}}
```

goreload

Live reload utility for Go programs.

More information: https://github.com/acoshift/goreload.

- Watch a binary file (defaults to `.goreload`):

```
goreload {{[-b|--bin]}} {{path/to/binary}} {{path/to/file.go}}
```

- Set a custom log prefix (defaults to `goreload`):

```
goreload --logPrefix {{prefix}} {{path/to/file.go}}
```

- Reload whenever any file changes:

```
goreload --all
```

gotelemetry

Manage Go telemetry data and settings.

More information: https://telemetry.go.dev/privacy#collection.

- Enable telemetry uploading:

```
gotelemetry on
```

- Disable telemetry uploading:

```
gotelemetry off
```

- Run a Web Viewer for local telemetry data:

```
gotelemetry view
```

- Print the current telemetry environment:

```
gotelemetry env
```

- Display help for a specific subcommand:

```
gotelemetry help {{subcommand}}
```

gotty

Share your terminal as a web application.

More information: https://github.com/yudai/gotty.

- Share result of command:

```
gotty {{command}}
```

- Share with write permission:

```
gotty {{[-w|--permit-write]}} {{shell}}
```

- Share with credential (Basic Auth):

```
gotty {{[-w|--permit-write]}} {{[-c|--credential]}}
{{username}}:{{password}} {{shell}}
```

gouldtoppm

Convert a Gould scanner file to a PPM image.

More information: https://netpbm.sourceforge.net/doc/gouldtoppm.html.

- Convert a Gould scanner file to a PPM image:

```
gouldtoppm {{path/to/file.gould}} > {{path/to/output.ppm}}
```

gource

Renders an animated tree diagram of Git, SVN, Mercurial and Bazaar repositories.

It shows files and directories being created, modified or removed over time.

More information: https://gource.io.

- Run gource in a directory (if it isn't the repository's root directory, the root is sought up from there):

```
gource {{path/to/repository}}
```

- Run gource in the current directory, with a custom output resolution:

```
gource -{{width}}x{{height}}
```

- Specify how long each day should be in the animation (this combines with -c, if provided):

```
gource {{[-s|--seconds-per-day]}} {{seconds}}
```

- Use fullscreen mode and a custom background color:

```
gource {{[-f|--fullscreen]}} {{[-b|--background-colour]}} {{hex_color_code}}
```

- Specify the animation title:

```
gource --title {{title}}
```

- Pause the animation:

```
<Space>
```

- Adjust simulation speed:

```
<{{+|-}}>
```

- Display help:

```
gource {{[-h|--help]}}
```

gow

Watches Go files and restarts the app on changes.

More information: https://github.com/mitranim/gow.

- Start and watch the current directory:

```
gow run .
```

- Start the application with the specified arguments:

```
gow run . {{argument1 argument2 ...}}
```

- Watch subdirectories in verbose mode:

```
gow -v -w={{path/to/directory1,path/to/directory2,...}} run .
```

- Watch the specified file extensions:

```
gow -e={{go,html}} run .
```

- Display help:

```
gow -h
```

gox

Cross-compile Go programs.

More information: https://github.com/mitchellh/gox.

- Compile Go program in the current directory for all operating systems and architecture combinations:

```
gox
```

- Download and compile a Go program from a remote URL:

```
gox {{url_1}} {{url_2}}
```

- Compile current directory for a particular operating system:

```
gox -os="{{os}}"
```

- Compile current directory for a single operating system and architecture combination:

```
gox -osarch="{{os}}/{{arch}}"
```

gpg-card

Administrate OpenPGP and PIV smart cards.

Similar to **`gpg --card-edit`**.

More information: https://manned.org/gpg-card.

- Start in interactive mode:

```
gpg-card
```

- Invoke one or more commands non-interactively:

```
gpg-card {{command1}} -- {{command2}} -- {{command3}}
```

- Show information about a smart card:

```
gpg-card list
```

- Retrieve the public key using the URL stored on an OpenPGP card:

```
gpg-card fetch
```

- Set the URL used by the `fetch` command:

```
gpg-card url
```

- Change or unblock PINs (uses the default action for the card in non-interactive mode):

```
gpg-card passwd
```

- Toggle the forcesig flag of an OpenPGP card (i.e. require entering the user PIN for signing):

```
gpg-card forcesig
```

- Factory reset a smart card (i.e. delete all data and reset PINs):

```
gpg-card factory-reset
```

gpg-tui

Terminal user interface for GNU Privacy Guard.

More information: https://github.com/orhun/gpg-tui.

- Start `gpg-tui`:

  `gpg-tui`

- Start `gpg-tui` with color and ASCII armored output:

  `gpg-tui --style {{colored}} --armor`

- Quit `gpg-tui`:

  `<q>`

- Interactively generate a new key:

  `<g>`

- Export the selected key:

  `<x>`

- Set the detail level for the selected key:

  `<1>|<2>|<3>`

- Refresh `gpg-tui`:

  `<r>`

- Display help in `gpg-tui`:

  `<?>`

gpg-zip

Encrypt files and directories in an archive using GPG.

More information: https://www.gnupg.org/documentation/manuals/gnupg/gpg_002dzip.html.

- Encrypt a directory into `archive.gpg` using a passphrase:

```
gpg-zip {{[-c|--symmetric]}} {{[-o|--output]}} {{archive.gpg}} {{path/to/directory}}
```

- Decrypt `archive.gpg` into a directory of the same name:

```
gpg-zip {{[-d|--decrypt]}} {{path/to/archive.gpg}}
```

- List the contents of the encrypted `archive.gpg`:

```
gpg-zip --list-archive {{path/to/archive.gpg}}
```

gpg

GNU Privacy Guard, an OpenPGP encryption and signing tool.

More information: https://gnupg.org/documentation/manuals/gnupg/Invoking-GPG.html.

- Create a GPG public and private key interactively:

```
gpg {{[--full-gen-key|--full-generate-key]}}
```

- List all keys from the public keyring:

```
gpg {{[-k|--list-keys]}}
```

- Sign `doc.txt` without encryption (writes output to `doc.txt.asc`):

```
gpg --clearsign {{doc.txt}}
```

- Encrypt and sign `doc.txt` for `alice@example.com` and `bob@example.com` (output to `doc.txt.gpg`):

```
gpg {{[-es|--encrypt --sign]}} {{[-r|--recipient]}}
{{alice@example.com}} {{[-r|--recipient]}}
{{bob@example.com}} {{doc.txt}}
```

- Encrypt `doc.txt` with only a passphrase (output to `doc.txt.gpg`):

```
gpg {{[-c|--symmetric]}} {{doc.txt}}
```

- Decrypt `doc.txt.gpg` (output to `stdout`):

```
gpg {{[-d|--decrypt]}} {{doc.txt.gpg}}
```

- Import a public key:

```
gpg --import {{public.gpg}}
```

- Export the public/private key for `alice@example.com` (output to `stdout`):

```
gpg {{--export|--export-secret-keys}} {{[-a|--armor]}}
{{alice@example.com}}
```

gpg2

This command is an alias of **gpg**.

- View documentation for the original command:

```
tldr gpg
```

gpgconf

Modify .gnupg home directories.

More information: https://www.gnupg.org/documentation/manuals/gnupg/gpgconf.html.

- List all components:

```
gpgconf --list-components
```

- List the directories used by gpgconf:

```
gpgconf {{[-L|--list-dirs]}}
```

- List all options of a component:

```
gpgconf --list-options {{component}}
```

- List programs and test whether they are runnable:

```
gpgconf --check-programs
```

- Reload a component:

```
gpgconf --reload {{component}}
```

gpgv

Verify OpenPGP signatures.

Note: **`gpgv`** reads trusted pubkeys from **`~/.gnupg/trustedkeys.kbx`** in absence of the **`--keyring`** option.

See also: **`gpg`**.

More information: https://www.gnupg.org/documentation/manuals/gnupg/gpgv.html.

- Verify a clearsigned or inline-signed file (the signature is embedded in the file itself):

```
gpgv {{path/to/file.asc}}
```

- Verify a detached signature (`.asc` or `.sig`) against its corresponding data file:

```
gpgv {{path/to/signature.asc}} {{path/to/data_file}}
```

- Verify a detached signature using a specific public keyring or exported public key file (`.gpg` or `.kbx`):

```
gpgv --keyring {{path/to/pubkey_or_keyring.gpg}} {{path/to/signature.asc}} {{path/to/data_file}}
```

- Verify a detached signature using a specific public key file in plain text format (`.txt`):

```
gpg --dearmor {{[-o|--output]}} {{path/to/pubkey.gpg}} {{path/to/pubkey.txt}} && gpgv --keyring {{path/to/pubkey.gpg}} {{path/to/signature.asc}} {{path/to/data_file}}
```

gprbuild

A high-level build tool for projects written in Ada and other languages (C/C++/Fortran).

More information: https://docs.adacore.com/gprbuild-docs/html/gprbuild_ug.html.

- Build a project (assuming only one `*.gpr` file exists in the current directory):

```
gprbuild
```

- Build a specific [P]roject file:

```
gprbuild -P {{project_name}}
```

- Clean up the build workspace:

```
gprclean
```

- Install compiled binaries:

```
gprinstall --prefix {{path/to/installation_directory}}
```

gprof

Performance analysis tool for many programming languages.

It profiles the function executions of a program.

More information: https://ftp.gnu.org/old-gnu/Manuals/gprof/html_mono/gprof.html.

- Compile binary to default `a.out` with gprof information and run it to get `gmon.out`:

```
gcc {{[-p|-pg]}} {{program.c}} && ./a.out
```

- Run gprof on default `a.out` and `gmon.out` to obtain profile output:

```
gprof
```

- Run gprof on a named binary:

```
gprof {{path/to/binary}} {{path/to/gmon.out}}
```

- Suppress profile field's description:

```
gprof {{[-b|--brief]}}
```

- Display routines that have zero usage:

```
gprof {{[-bz|--brief --display-unused-functions]}}
```

gradle build

Build a project using Gradle.

More information: https://docs.gradle.org/current/userguide/gradle_basics.html.

- Build the project:

```
gradle build
```

- Perform a clean build:

```
gradle clean build
```

- Build the project while skipping tests:

```
gradle build {{[-x|--exclude-task]}} test
```

- Build with more detailed logging:

```
gradle build {{[-i|--info]}}
```

gradle

An open source build automation system.

More information: https://gradle.org.

- Compile a package:

```
gradle build
```

- Exclude test task:

```
gradle build {{[-x|--exclude-task]}} {{test}}
```

- Run in offline mode to prevent Gradle from accessing the network during builds:

```
gradle build --offline
```

- Clear the build directory:

```
gradle clean
```

- Build an Android Package (APK) in release mode:

```
gradle assembleRelease
```

- List the main tasks:

```
gradle tasks
```

- List all the tasks:

```
gradle tasks --all
```

grafana-cli

A small executable that is bundled with Grafana serve.

More information: https://grafana.com/docs/grafana/latest/cli/.

- Install, update, or remove specific plugins:

```
grafana-cli plugins {{install|update|remove}} {{plugin_id1 plugin_id2 ...}}
```

- List all installed plugins:

```
grafana-cli plugins ls
```

grap

A charting preprocessor for the groff (GNU Troff) document formatting system.

See also: **`pic`**, **`groff`**.

More information: https://manned.org/grap.

- Process a `grap` file and save the output file for future processing with `pic` and `groff`:

```
grap {{path/to/input.grap}} > {{path/to/output.pic}}
```

- Typeset a `grap` file to PDF using the [me] macro package, saving the output to a file:

```
grap {{path/to/input.grap}} | pic -T {{pdf}} | groff -{{me}} -T {{pdf}} > {{path/to/output.pdf}}
```

graphml2gv

Convert a graph from **`graphml`** to **`gv`** format.

Converters: **`gml2gv`**, **`gv2gml`**, **`gv2gxl`**, **`gxl2gv`**, **`graphml2gv`** & **`mm2gv`**.

More information: https://graphviz.org/pdf/graphml2gv.1.pdf.

- Convert a graph from `gml` to `gv` format:

```
graphml2gv -o {{output.gv}} {{input.gml}}
```

- Convert a graph using `stdin` and `stdout`:

```
cat {{input.gml}} | graphml2gv > {{output.gv}}
```

- Display help:

```
graphml2gv -?
```

>

Redirect output.

More information: https://gnu.org/software/bash/manual/bash.html#Redirecting-Output.

- Redirect `stdout` to a file:

```
{{command}} > {{path/to/file}}
```

- Append to a file:

```
{{command}} >> {{path/to/file}}
```

- Redirect both `stdout` and `stderr` to a file:

```
{{command}} &> {{path/to/file}}
```

- Redirect `stderr` to `/dev/null` to keep the terminal output clean:

```
{{command}} 2> /dev/null
```

- Clear the file contents or create a new empty file:

```
> {{path/to/file}}
```

- Redirect `stderr` to `stdout` for piping them together:

```
{{command1}} 2>&1 | {{command2}}
```

- Open a persistent file descriptor:

```
exec {{3}}>{{path/to/file}}
```

- Write to a custom file descriptor:

```
{{echo text}} >&{{3}}
```

grep

Find patterns in files using **regex**es.

More information: https://www.gnu.org/software/grep/manual/grep.html.

- Search for a pattern within a file:

```
grep "{{search_pattern}}" {{path/to/file1 path/to/file2 ...}}
```

- Search for an exact string (disables regexes):

```
grep {{[-F|--fixed-strings]}} "{{exact_string}}" {{path/to/file}}
```

- Search for a pattern in all files recursively in a directory, showing line numbers of matches, ignoring binary files:

```
grep {{[-rnI|--recursive --line-number --binary-files=without-match]}} "{{search_pattern}}" {{path/to/directory}}
```

- Use extended regexes (supports ?, +, {}, (), and |), in case-insensitive mode:

```
grep {{[-Ei|--extended-regexp --ignore-case]}} "{{search_pattern}}" {{path/to/file}}
```

- Print 3 lines of [C]ontext around, [B]efore or [A]fter each match:

```
grep {{--context|--before-context|--after-context}} 3 "{{search_pattern}}" {{path/to/file}}
```

- Print file name and line number for each match with color output:

```
grep {{[-Hn|--with-filename --line-number]}} --color=always "{{search_pattern}}" {{path/to/file}}
```

- Search for lines matching a pattern, printing only the matched text:

```
grep {{[-o|--only-matching]}} "{{search_pattern}}" {{path/to/file}}
```

- Search stdin for lines that do not match a pattern:

```
cat {{path/to/file}} | grep {{[-v|--invert-match]}} "{{search_pattern}}"
```

grex

Generate **regex**s.

More information: https://github.com/pemistahl/grex.

- Generate a simple regex:

```
grex {{space_separated_strings}}
```

- Generate a case-insensitive regex:

```
grex {{[-i|--ignore-case]}} {{space_separated_strings}}
```

- Replace digits with '\d':

```
grex {{[-d|--digits]}} {{space_separated_strings}}
```

- Replace Unicode word character with '\w':

```
grex {{[-w|--words]}} {{space_separated_strings}}
```

- Replace spaces with '\s':

```
grex {{[-s|--spaces]}} {{space_separated_strings}}
```

- Add {min, max} quantifier representation for repeating sub-strings:

```
grex {{[-r|--repetitions]}} {{space_separated_strings}}
```

grip

Preview GitHub-flavoured Markdown files locally.

More information: https://manned.org/grip.

- Start the server and serve the rendered `README` file of a current directory:

```
grip
```

- Start the server and serve a specific Markdown file:

```
grip {{path/to/file.md}}
```

- Start the server and open the `README` file of the current directory in the browser:

```
grip {{[-b|--browser]}}
```

- Start the server in the specified port and serve the rendered `README` file of the current directory:

```
grip {{port}}
```

groff

GNU replacement for the **`troff`** and **`nroff`** typesetting utilities.

More information: https://www.gnu.org/software/groff/manual/groff.html.node/Groff-Options.html.

- Format output for a PostScript printer, saving the output to a file:

```
groff {{path/to/input.roff}} > {{path/to/output.ps}}
```

- Render a man page using the ASCII output device, and display it using a pager:

```
groff -man -T ascii {{path/to/manpage.1}} | less --RAW-CONTROL-CHARS
```

- Render a man page into an HTML file:

```
groff -man -T html {{path/to/manpage.1}} > {{path/to/manpage.html}}
```

- Typeset a roff file containing [t]ables and [p]ictures, using the [me] macro set, to PDF, saving the output:

```
groff {{-t}} {{-p}} -{{me}} -T {{pdf}} {{path/to/input.me}} > {{path/to/output.pdf}}
```

- Run a `groff` command with preprocessor and macro options guessed by the `grog` utility:

```
eval "$(grog -T utf8 {{path/to/input.me}})"
```

gron

Transform **JSON** into individual assignments for easier management.

More information: https://manned.org/gron.

- Process JSON file into individual assignments:

```
gron {{path/to/file|url}}
```

- Don't sort output data:

```
gron --no-sort {{path/to/file|url}}
```

- Disable certificate validation:

```
gron {{[-k|--insecure]}} {{url}}
```

- Display values of gron assignments:

```
gron {{[-v|--values]}} {{path/to/file|url}}
```

- Turn assignments converted with gron back into JSON:

```
gron {{[-u|--ungron]}} {{path/to/file|url}}
```

- Process individual lines of input as separate JSON objects:

```
gron {{[-s|--stream]}} {{path/to/file|url}}
```

- Represent processed data as a JSON stream:

```
gron {{[-j|--json]}} {{path/to/file|url}}
```

groups

Print group memberships for a user.

See also: **`groupadd`**, **`groupdel`**, **`groupmod`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/groups-invocation.html.

- Print group memberships for the current user:

```
groups
```

- Print group memberships for a list of users:

```
groups {{username1 username2 ...}}
```

grpcurl

Interact with gRPC servers.

Like **`curl`**, but for gRPC.

More information: https://github.com/fullstorydev/grpcurl.

- Send an empty request:

```
grpcurl {{grpc.server.com:443}} {{my.custom.server.Service/Method}}
```

- Send a request with a header and a body:

```
grpcurl -H "{{Authorization: Bearer $token}}" -d '{{{"foo": "bar"}}}' {{grpc.server.com:443}} {{my.custom.server.Service/Method}}
```

- List all services exposed by a server:

```
grpcurl {{grpc.server.com:443}} list
```

- List all methods in a particular service:

```
grpcurl {{grpc.server.com:443}} list {{my.custom.server.Service}}
```

grumphp

A PHP Composer plugin that enables source code quality checks.

More information: https://github.com/phpro/grumphp#commands.

- Register the Git hooks:

  ```
  grumphp git:init
  ```

- Trigger the pre-commit hook manually:

  ```
  grumphp git:pre-commit
  ```

- Check every versioned file:

  ```
  grumphp run
  ```

grunt

A JavaScript task runner for automating processes.

More information: https://github.com/gruntjs/grunt-cli.

- Run the default task process:

```
grunt
```

- Run one or more tasks:

```
grunt {{task1 task2 ...}}
```

- Specify an alternative configuration file:

```
grunt --gruntfile {{path/to/file}}
```

- Specify an alternative base path for relative files:

```
grunt --base {{path/to/directory}}
```

- Specify an additional directory to scan for tasks in:

```
grunt --tasks {{path/to/directory}}
```

- Perform a dry-run without writing any files:

```
grunt --no-write
```

- Display help:

```
grunt --help
```

gst-device-monitor-1.0

List devices detected by GStreamer.

More information: https://manned.org/gst-device-monitor-1.0.

- List all current devices:

```
gst-device-monitor-1.0
```

- List all current devices and monitor for any changes:

```
gst-device-monitor-1.0 {{[-f|--follow]}}
```

- List devices of a specific type:

```
gst-device-monitor-1.0 {{Audio|Video}}/{{Source|Sink}}
```

gst-discoverer-1.0

Display file metadata and stream information.

More information: https://manned.org/gst-discoverer-1.0.1.

- Print file metadata:

```
gst-discoverer-1.0 {{path/to/file}}
```

gst-inspect-1.0

Print information on GStreamer plugins.

More information: https://gstreamer.freedesktop.org/documentation/tools/gst-inspect.html.

- Print information on a plugin:

```
gst-inspect-1.0 {{plugin}}
```

- List hardware transcoding capabilities of your device:

```
gst-inspect-1.0 {{vaapi|nvcodec|...}}
```

- List available container plugins:

```
gst-inspect-1.0 {{matroska|avi|ogg|isomp4|...}}
```

- List available audio codecs:

```
gst-inspect-1.0 {{opus|vorbis|flac|...}}
```

- List GStreamer core elements:

```
gst-inspect-1.0 coreelements
```

- List plugins that utilize graphics APIs:

```
gst-inspect-1.0 {{vulkan|opengl|...}}
```

- List available image codecs:

```
gst-inspect-1.0 {{png|jpeg|...}}
```

- List all available plugins:

```
gst-inspect-1.0
```

gst-launch-1.0

Build and run a GStreamer pipeline.

See also: **`gst-inspect-1.0`**, **`ffmpeg`**.

More information: https://gstreamer.freedesktop.org/documentation/tools/gst-launch.html.

- Play test video in a window:

```
gst-launch-1.0 videotestsrc ! autovideosink
```

- Play test audio:

```
gst-launch-1.0 audiotestsrc ! autoaudiosink
```

- Play a media file in a window:

```
gst-launch-1.0 playbin uri={{protocol}}://{{host}}/{{path/to/file}}
```

- Re-encode a media file:

```
gst-launch-1.0 filesrc location={{path/to/file}} ! {{file_type}}demux ! {{codec_type}}dec ! {{codec_type}}enc ! {{file_type}}mux ! filesink location={{path/to/file}}
```

- Stream a file to an RTSP server:

```
gst-launch-1.0 filesrc location={{path/to/file}} ! rtspclientsink location=rtsp://{{host_IP}}/{{path/to/file}}
```

- Force an End Of Stream event if the pipeline is shut down with `<Ctrl c>` for containers that require finalization such as `mp4`:

```
gst-launch-1.0 {{[-e|--eos-on-shutdown]}} videotestsrc ! x264enc ! mp4mux ! filesink location={{path/to/file.mp4}}
```

- Multiplex together test video and test audio into a file:

```
gst-launch-1.0 {{[-e|--eos-on-shutdown]}} videotestsrc ! x264enc ! {{element_name}}. audiotestsrc ! opusenc ! {{element_name}}. matroskamux name={{element_name}} ! filesink location={{path/to/file.mkv}}
```

- Dump a pipeline into a `.dot` file which can then be rendered with tools like `dot`:

```
GST_DEBUG_DUMP_DOT_DIR={{path/to/directory}} gst-launch-1.0
{{pipeline}}
```

gsutil

Access Google Cloud Storage.

You can use **`gsutil`** to do a wide range of bucket and object management tasks.

More information: https://cloud.google.com/storage/docs/gsutil.

- List all buckets in a project you are logged into:

```
gsutil ls
```

- List the objects in a bucket:

```
gsutil ls -r 'gs://{{bucket_name}}/{{prefix}}**'
```

- Download an object from a bucket:

```
gsutil cp gs://{{bucket_name}}/{{object_name}} {{path/to/
save_location}}
```

- Upload an object to a bucket:

```
gsutil cp {{object_location}} gs://
{{destination_bucket_name}}/
```

- Rename or move objects in a bucket:

```
gsutil mv gs://{{bucket_name}}/{{old_object_name}} gs://
{{bucket_name}}/{{new_object_name}}
```

- Create a new bucket in the project you are logged into:

```
gsutil mb gs://{{bucket_name}}
```

- Delete a bucket and remove all the objects in it:

```
gsutil rm -r gs://{{bucket_name}}
```

gt

Create and manage sequences of dependent code changes (stacks) for Git and GitHub.

More information: https://graphite.dev/docs/get-started.

- Initialize `gt` for the repository in the current directory:

```
gt init
```

- Create a new branch stacked on top of the current branch and commit staged changes:

```
gt create {{branch_name}}
```

- Create a new commit and fix upstack branches:

```
gt modify -cam {{commit_message}}
```

- Force push all branches in the current stack to GitHub and create or update PRs:

```
gt stack submit
```

- Checkout different branch (prompts interactive mode when branch name is omitted):

```
gt co {{branch_name}}
```

- Sync stack with remote version (also deletes merged branches):

```
gt sync
```

- Log all tracked stacks:

```
gt log short
```

- Display help for a specified subcommand:

```
gt {{subcommand}} --help
```

gtop

System monitoring dashboard for the terminal.

More information: https://github.com/aksakalli/gtop.

- Show the system stats dashboard:

  ```
  gtop
  ```

- Sort by CPU usage:

  ```
  <c>
  ```

- Sort by memory usage:

  ```
  <m>
  ```

guacd

Apache Guacamole proxy daemon.

Support loader for client plugins to interface between the Guacamole protocol and any arbitrary remote desktop protocol (e.g. RDP, VNC, Other).

More information: https://manned.org/guacd.

- Bind to a specific port on localhost:

```
guacd -b {{127.0.0.1}} -l {{4823}}
```

- Start in debug mode, keeping the process in the foreground:

```
guacd -f -L {{debug}}
```

- Start with TLS support:

```
guacd -C {{my-cert.crt}} -K {{my-key.pem}}
```

- Write the PID to a file:

```
guacd -p {{path/to/file.pid}}
```

guetzli

JPEG image compression utility.

More information: https://github.com/google/guetzli.

- Compress a JPEG image:

```
guetzli {{input.jpg}} {{output.jpg}}
```

- Create a compressed JPEG from a PNG:

```
guetzli {{input.png}} {{output.jpg}}
```

- Compress a JPEG with the desired visual quality (84-100):

```
guetzli --quality {{quality_value}} {{input.jpg}}
{{output.jpg}}
```

guile

Guile Scheme interpreter.

More information: https://www.gnu.org/software/guile/manual/guile.html#Invoking-Guile.

- Start a REPL (interactive shell):

```
guile
```

- Execute the script in a given Scheme file:

```
guile {{script.scm}}
```

- Execute a Scheme expression:

```
guile -c "{{expression}}"
```

- Listen on a port or a Unix domain socket (the default is port 37146) for remote REPL connections:

```
guile --listen={{port_or_socket}}
```

gulp

JavaScript task runner and streaming build system.

Tasks are defined within **`gulpfile.js`** at the project root.

More information: https://github.com/gulpjs/gulp-cli.

- Run the default task:

  ```
  gulp
  ```

- Run individual tasks:

  ```
  gulp {{task}} {{othertask}}
  ```

- Print the task dependency tree for the loaded gulpfile:

  ```
  gulp --tasks
  ```

gum

Make glamorous shell scripts.

See also: **`whiptail`**, **`dialog`**.

More information: https://github.com/charmbracelet/gum#tutorial.

- Interactively pick a specific option to print to `stdout`:

```
gum choose {{"option1" "option2" "..."}}
```

- Open an interactive prompt for the user to input a string with a specific placeholder:

```
gum input --placeholder "{{value}}"
```

- Open an interactive confirmation prompt and exit with either `<0>` or `<1>`:

```
gum confirm "{{Continue?}}" --default=false --affirmative
"{{Yes}}" --negative "{{No}}" {{&& echo "Yes selected" ||
echo "No selected"}}
```

- Show a spinner while a command is taking place with text alongside:

```
gum spin {{[-s|--spinner]}} {{dot|line|minidot|jump|pulse|
points|globe|moon|monkey|meter|hamburger}} --title
"{{loading...}}" {{command}}
```

- Format text to include emojis:

```
gum format {{[-t|--type]}} {{emoji}} "{{:smile: :heart:
hello}}"
```

- Interactively prompt for multi-line text and write the input to a file:

```
gum write > {{path/to/file}}
```

gunicorn

Python WSGI HTTP Server.

More information: https://docs.gunicorn.org/en/latest/run.html.

- Run Python web app:

```
gunicorn {{import.path:app_object}}
```

- Listen on port 8080 on localhost:

```
gunicorn {{[-b|--bind]}} {{localhost}}:{{8080}}
{{import.path:app_object}}
```

- Turn on live reload:

```
gunicorn --reload {{import.path:app_object}}
```

- Use 4 worker processes for handling requests:

```
gunicorn {{[-w|--workers]}} {{4}} {{import.path:app_object}}
```

- Use 4 worker threads for handling requests:

```
gunicorn --threads {{4}} {{import.path:app_object}}
```

- Run app over HTTPS:

```
gunicorn --certfile {{cert.pem}} --keyfile {{key.pem}}
{{import.path:app_object}}
```

gunzip

Extract files from a **`gzip`** (**`.gz`**) archive.

More information: https://manned.org/gunzip.

- Extract a file from an archive, replacing the original file if it exists:

```
gunzip {{archive.tar.gz}}
```

- Extract a file to a target destination:

```
gunzip {{[-c|--stdout]}} {{archive.tar.gz}} > {{archive.tar}}
```

- Extract a file and keep the archive file:

```
gunzip {{[-k|--keep]}} {{archive.tar.gz}}
```

- List the contents of a compressed file:

```
gunzip {{[-l|--list]}} {{file.txt.gz}}
```

- Decompress an archive from `stdin`:

```
cat {{path/to/archive.gz}} | gunzip
```

gv2gml

Convert a graph from **`gv`** to **`gml`** format.

Converters: **`gml2gv`**, **`gv2gml`**, **`gv2gxl`**, **`gxl2gv`**, **`graphml2gv`** & **`mm2gv`**.

More information: https://graphviz.org/pdf/gml2gv.1.pdf.

- Convert a graph from `gv` to `gml` format:

```
gv2gml -o {{output.gml}} {{input.gv}}
```

- Convert a graph using `stdin` and `stdout`:

```
cat {{input.gv}} | gv2gml > {{output.gml}}
```

- Display help:

```
gv2gml -?
```

gv2gxl

Convert a graph from **`gv`** to **`gxl`** format.

Converters: **`gml2gv`**, **`gv2gml`**, **`gv2gxl`**, **`gxl2gv`**, **`graphml2gv`** & **`mm2gv`**.

More information: https://graphviz.org/pdf/gxl2gv.1.pdf.

- Convert a graph from `gv` to `gxl` format:

```
gv2gxl -o {{output.gxl}} {{input.gv}}
```

- Convert a graph using `stdin` and `stdout`:

```
cat {{input.gv}} | gv2gxl > {{output.gxl}}
```

- Display help:

```
gv2gxl -?
```

gvcolor

Colorize a ranked digraph with a range of colors.

Graphviz filters: **`acyclic`**, **`bcomps`**, **`comps`**, **`edgepaint`**, **`gvcolor`**, **`gvpack`**, **`mingle`**, **`nop`**, **`sccmap`**, **`tred`**, & **`unflatten`**.

More information: https://graphviz.org/pdf/gvcolor.1.pdf.

- Colorize one or more ranked digraph (that were already processed by `dot`):

```
gvcolor {{path/to/layout1.gv path/to/layout2.gv ...}} > {{path/to/output.gv}}
```

- Lay out a graph and colorize it, then convert to a PNG image:

```
dot {{path/to/input.gv}} | gvcolor | dot -T {{png}} > {{path/to/output.png}}
```

- Display help:

```
gvcolor -?
```

gvedit

Simple Graphviz graph editor and viewer.

More information: https://graphviz.org/pdf/gvedit.1.pdf.

- Open a Graphviz file in the editor:

```
gvedit {{path/to/graph.gv}}
```

- [s]cale coordinates from points to inches (useful for layouts generated by Graphviz tools):

```
gvedit -s {{path/to/graph.gv}}
```

- Enable [v]erbose mode:

```
gvedit -v {{path/to/graph.gv}}
```

- Display help:

```
gvedit -?
```

gvgen

Generate simple, structured abstract graphs.

More information: https://graphviz.org/pdf/gvgen.1.pdf.

- Generate a [c]ycle with 10 vertices and edges and write it to `stdout`:

```
gvgen -c {{10}}
```

- Generate a 4×3 [g]rid:

```
gvgen -g {{4,3}}
```

- Generate a binary [t]ree of height 5:

```
gvgen -t {{5}}
```

- Generate a complete [b]ipartite graph with 3 and 4 vertices:

```
gvgen -b {{3,4}}
```

- Create a [r]andom graph and [o]utput it to a file:

```
gvgen -r {{10,5}} -o {{random.gv}}
```

- Generate a [d]irected graph with [v]erbose output:

```
gvgen -d -v -c {{6}}
```

- Display help:

```
gvgen -?
```

gvim

A Graphical User Interface version of Vim (Vi IMproved), a command-line text editor.

See also: **`vimdiff`**, **`vimtutor`**, **`nvim`**, **`vim`**.

More information: https://www.vim.org.

- Open `gvim`:

```
gvim
```

- Open a specific file:

```
gvim {{path/to/file}}
```

gvmap

Find clusters and creates a geographical map highlighting clusters.

More information: https://graphviz.org/pdf/gvmap.1.pdf.

- Generate a clustered map layout from a DOT format graph:

```
gvmap {{graph.gv}} -o {{output.xdot}}
```

- Generate a map using existing cluster subgraphs from the input:

```
gvmap -D {{graph.gv}}
```

- Include graph edges in the [o]utput map:

```
gvmap -e {{graph.gv}} -o {{output.xdot}}
```

- Specify a [c]olor scheme (1: pastel, 2: blue-yellow, 3: white-red, etc.):

```
gvmap -c {{1|2|3|...}} {{graph.gv}} -o {{output.xdot}}
```

- Set the maximum number of [C]lusters (by default 0, meaning there is no limit):

```
gvmap -C {{10}} {{graph.gv}} -o {{output.xdot}}
```

- Only draw cluster 2 (by default, all clusters are drawn):

```
gvmap -highlight={{2}} {{graph.gv}} -o {{output.xdot}}
```

gvpack

Combine several graph layouts (that already have layout information).

Graphviz filters: **`acyclic`**, **`bcomps`**, **`comps`**, **`edgepaint`**, **`gvcolor`**, **`gvpack`**, **`mingle`**, **`nop`**, **`sccmap`**, **`tred`**, & **`unflatten`**.

More information: https://graphviz.org/pdf/gvpack.1.pdf.

- Combine several graph layouts (that already have layout information):

```
gvpack {{path/to/layout1.gv path/to/layout2.gv ...}} >
{{path/to/output.gv}}
```

- Combine several graph layouts at the graph level, keeping graphs separate:

```
gvpack -g {{path/to/layout1.gv path/to/layout2.gv ...}} >
{{path/to/output.gv}}
```

- Combine several graph layouts at the node level, ignoring clusters:

```
gvpack -n {{path/to/layout1.gv path/to/layout2.gv ...}} >
{{path/to/output.gv}}
```

- Combine several graph layouts without packing:

```
gvpack -u {{path/to/layout1.gv path/to/layout2.gv ...}} >
{{path/to/output.gv}}
```

- Display help:

```
gvpack -?
```

gxl2gv

Convert a graph from **`gxl`** to **`gv`** format.

Converters: **`gml2gv`**, **`gv2gml`**, **`gv2gxl`**, **`gxl2gv`**, **`graphml2gv`** & **`mm2gv`**.

More information: https://graphviz.org/pdf/gxl2gv.1.pdf.

- Convert a graph from `gxl` to `gv` format:

```
gxl2gv -o {{output.gv}} {{input.gxl}}
```

- Convert a graph using `stdin` and `stdout`:

```
cat {{input.gxl}} | gxl2gv > {{output.gv}}
```

- Display help:

```
gxl2gv -?
```

gyb

Locally back up Gmail messages using Gmail's API over HTTPS.

More information: https://github.com/GAM-team/got-your-back.

- Estimate the number and the size of all emails on your Gmail account:

```
gyb --email {{email@gmail.com}} --action estimate
```

- Backup a Gmail account to a specific directory:

```
gyb --email {{email@gmail.com}} --action backup --local-folder {{path/to/directory}}
```

- Backup only important or starred emails from a Gmail account to the default local folder:

```
gyb --email {{email@gmail.com}} --search "{{is:important OR is:starred}}"
```

- Restore from a local folder to a Gmail account:

```
gyb --email {{email@gmail.com}} --action restore --local-folder {{path/to/directory}}
```

gzip

Compress/uncompress files with **`gzip`** compression (LZ77).

More information: https://www.gnu.org/software/gzip/manual/gzip.html.

- Compress a file, replacing it with a `gzip` archive:

```
gzip {{path/to/file}}
```

- Decompress a file, replacing it with the original uncompressed version:

```
gzip {{[-d|--decompress]}} {{path/to/file.gz}}
```

- Compress a file, keeping the original file:

```
gzip {{[-k|--keep]}} {{path/to/file}}
```

- Compress a file, specifying the output filename:

```
gzip {{[-c|--stdout]}} {{path/to/file}} > {{path/to/
compressed_file.gz}}
```

- Decompress a `gzip` archive specifying the output filename:

```
gzip {{[-cd|--stdout --decompress]}} {{path/to/file.gz}} >
{{path/to/uncompressed_file}}
```

- Specify the compression level. 1 is the fastest (low compression), 9 is the slowest (high compression), 6 is the default:

```
gzip -{{1..9}} {{[-c|--stdout]}} {{path/to/file}} > {{path/
to/compressed_file.gz}}
```

- Display the name and reduction percentage for each file compressed or decompressed:

```
gzip {{[-vd|--verbose --decompress]}} {{path/to/file.gz}}
```

hadolint

Dockerfile linter.

More information: https://github.com/hadolint/hadolint.

- Lint a Dockerfile:

```
hadolint {{path/to/Dockerfile}}
```

- Lint a Dockerfile, displaying the output in JSON format:

```
hadolint --format {{json}} {{path/to/Dockerfile}}
```

- Lint a Dockerfile, displaying the output in a specific format:

```
hadolint --format {{tty|json|checkstyle|codeclimate|codacy}}
{{path/to/Dockerfile}}
```

- Lint a Dockerfile ignoring specific rules:

```
hadolint --ignore {{DL3006}} --ignore {{DL3008}} {{path/to/
Dockerfile}}
```

- Lint multiple Dockerfiles using specific trusted registries:

```
hadolint --trusted-registry {{docker.io}} --trusted-registry
{{example.com}}:{{5000}} {{path/to/Dockerfile1 path/to/
Dockerfile2 ...}}
```

hakyll-init

Generate a new Hakyll sample blog.

More information: https://github.com/jaspervdj/hakyll-init.

- Generate a new Hakyll sample blog:

```
hakyll-init {{path/to/directory}}
```

- Display help:

```
hakyll-init --help
```

handbrakecli

Command-line interface to the HandBrake video conversion and DVD ripping tool.

More information: https://handbrake.fr/docs/en/latest/cli/command-line-reference.html.

- Convert a video file to MKV (AAC 160kbit audio and x264 CRF20 video):

```
handbrakecli {{[-i|--input]}} {{input.avi}} {{[-o|--output]}} {{output.mkv}} {{[-e|--encoder]}} x264 {{[-q|--quality]}} 20 {{[-B|--ab]}} 160
```

- Resize a video file to 320x240:

```
handbrakecli {{[-i|--input]}} {{input.mp4}} {{[-o|--output]}} {{output.mp4}} {{[-w|--width]}} 320 {{[-l|--height]}} 240
```

- List available presets:

```
handbrakecli {{[-z|--preset-list]}}
```

- Convert an AVI video to MP4 using the Android preset:

```
handbrakecli {{[-Z|--preset]}} "Android" {{[-i|--input]}} {{input.ext}} {{[-o|--output]}} {{output.mp4}}
```

- Print the content of a DVD, getting the CSS keys in the process:

```
handbrakecli {{[-i|--input]}} {{/dev/sr0}} {{[-t|--title]}} 0
```

- Rip the first track of a DVD in the specified device. Audiotracks and subtitle languages are specified as lists:

```
handbrakecli {{[-i|--input]}} {{/dev/sr0}} {{[-t|--title]}} 1 {{[-o|--output]}} {{out.mkv}} {{[-f|--format]}} av_mkv {{[-e|--encoder]}} x264 {{[-s|--subtitle]}} {{1,4,5}} {{[-a|--audio]}} {{1,2}} {{[-E|--aencoder]}} copy {{[-q|--quality]}} {{23}}
```

hangups

Third party client for Google Hangouts.

More information: https://github.com/tdryer/hangups.

- Start `hangups`:

  ```
  hangups
  ```

- Display troubleshooting information and help:

  ```
  hangups -h
  ```

- Set a refresh token for hangups:

  ```
  hangups --token-path {{path/to/token}}
  ```

hardhat

A development environment for Ethereum software.

More information: https://hardhat.org/hardhat-runner/docs/getting-started#quick-start.

- List available subcommands (or create a new project if no configuration exists):

```
hardhat
```

- Compile the current project and build all artifacts:

```
hardhat compile
```

- Run a user-defined script after compiling the project:

```
hardhat run {{path/to/script.js}}
```

- Run Mocha tests:

```
hardhat test
```

- Run all given test files:

```
hardhat test {{path/to/file1.js path/to/file2.js ...}}
```

- Start a local Ethereum JSON-RPC node for development:

```
hardhat node
```

- Start a local Ethereum JSON-RPC node with a specific hostname and port:

```
hardhat node --hostname {{hostname}} --port {{port}}
```

- Clean the cache and all artifacts:

```
hardhat clean
```

hash

View cached executable locations.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-hash.

- View cached command locations for the current shell:

```
hash
```

- Clear the hash table:

```
hash -r
```

- Delete a specific command from the hash table:

```
hash -d {{command}}
```

- Print the full path of `command`:

```
hash -t {{command}}
```

- Display help:

```
hash --help
```

hashcat

Fast and advanced password recovery tool.

More information: https://hashcat.net/wiki/doku.php?id=hashcat.

- Perform a brute-force attack (mode 3) with the default hashcat mask:

```
hashcat --hash-type {{hash_type_id}} --attack-mode {{3}}
{{hash_value}}
```

- Perform a brute-force attack (mode 3) with a known pattern of 4 digits:

```
hashcat --hash-type {{hash_type_id}} --attack-mode {{3}}
{{hash_value}} "{{?d?d?d?d}}"
```

- Perform a brute-force attack (mode 3) using at most 8 of all printable ASCII characters:

```
hashcat --hash-type {{hash_type_id}} --attack-mode {{3}} --
increment {{hash_value}} "{{?a?a?a?a?a?a?a?a}}"
```

- Perform a dictionary attack (mode 0) using the RockYou wordlist of a Kali Linux box:

```
hashcat --hash-type {{hash_type_id}} --attack-mode {{0}}
{{hash_value}} {{/usr/share/wordlists/rockyou.txt}}
```

- Perform a rule-based dictionary attack (mode 0) using the RockYou wordlist mutated with common password variations:

```
hashcat --hash-type {{hash_type_id}} --attack-mode {{0}} --
rules-file {{/usr/share/hashcat/rules/best64.rule}}
{{hash_value}} {{/usr/share/wordlists/rockyou.txt}}
```

- Perform a combination attack (mode 1) using the concatenation of words from two different custom dictionaries:

```
hashcat --hash-type {{hash_type_id}} --attack-mode {{1}}
{{hash_value}} /{{path/to/dictionary1.txt}} /{{path/to/
dictionary2.txt}}
```

- Show result of an already cracked hash:

```
hashcat --show {{hash_value}}
```

- Show all example hashes:

```
hashcat --example-hashes
```

hashid

Python3 program that identifies data and password hashes.

More information: https://github.com/psypanda/hashID.

- Identify hashes from `stdin` (through typing, copying and pasting, or piping the hash into the program):

```
hashid
```

- Identify one or more hashes:

```
hashid {{hash1 hash2 ...}}
```

- Identify hashes on a file (one hash per line):

```
hashid {{path/to/hashes.txt}}
```

- Show all possible hash types (including salted hashes):

```
hashid --extended {{hash}}
```

- Show `hashcat`'s mode number and `john`'s format string of the hash types:

```
hashid --mode --john {{hash}}
```

- Save output to a file instead of printing to `stdout`:

```
hashid --outfile {{path/to/output.txt}} {{hash}}
```

hatch

Modern, extensible Python project manager.

See also: **`poetry`**.

More information: https://hatch.pypa.io/latest/cli/reference/.

- Create a new Hatch project:

  ```
  hatch new {{project_name}}
  ```

- Initialize Hatch for an existing project:

  ```
  hatch new --init
  ```

- Build a Hatch project:

  ```
  hatch build
  ```

- Remove build artifacts:

  ```
  hatch clean
  ```

- Create a default environment with dependencies defined in the `pyproject.toml` file:

  ```
  hatch env create
  ```

- Show environment dependencies as a table:

  ```
  hatch dep show table
  ```

haxelib

Haxe Library Manager.

More information: https://lib.haxe.org/documentation/using-haxelib/.

- Search for a Haxe library:

  ```
  haxelib search {{keyword}}
  ```

- Install a Haxe library:

  ```
  haxelib install {{libname}}
  ```

- Install a specific version of a Haxe library:

  ```
  haxelib install {{libname}} {{version}}
  ```

- Upgrade all installed Haxe libraries:

  ```
  haxelib upgrade
  ```

- Install the development version of a library from a Git repository:

  ```
  haxelib git {{libname}} {{git_url}}
  ```

- Uninstall a Haxe library:

  ```
  haxelib remove {{libname}}
  ```

- Print a tree of locally installed Haxe libraries:

  ```
  haxelib list
  ```

hcloud

Show how to use the CLI for Hetzner Cloud.

More information: https://github.com/hetznercloud/cli.

- Show available commands and flags:

  `hcloud`

- Display help:

  `hcloud -h`

- Show available commands and flags for `hcloud` contexts:

  `hcloud context`

hd

This command is an alias of **hexdump**.

- View documentation for the original command:

```
tldr hexdump
```

head

Output the first part of files.

More information: https://manned.org/head.1p.

- Output the first few lines of a file:

```
head -n {{count}} {{path/to/file}}
```

helix

Helix, A post-modern text editor, provides several modes for different kinds of text manipulation.

Pressing **<i>** enters insert mode. **<Esc>** enters normal mode, which enables the use of Helix commands.

More information: https://helix-editor.com.

- Open a file:

  ```
  helix {{path/to/file}}
  ```

- Open files and show them one next each other:

  ```
  helix --vsplit {{path/to/file1 path/to/file2}}
  ```

- Show the tutorial to learn Helix (or access it within Helix by pressing `<Esc>` and typing `<:>tutor<Enter>`):

  ```
  helix --tutor
  ```

- Change the Helix theme:

  ```
  <:>theme {{theme_name}}
  ```

- Save and Quit:

  ```
  <:>wq<Enter>
  ```

- Force-quit without saving:

  ```
  <:>q!<Enter>
  ```

- Undo the last operation:

  ```
  <u>
  ```

- Search for a pattern in the file (press `<n>`/`<N>` to go to next/previous match):

  ```
  </>{{search_pattern}}<Enter>
  ```

hello

Print "Hello, world!", "hello, world" or a customizable text.

More information: https://www.gnu.org/software/hello/manual/hello.html#Invoking-hello.

- Print "Hello, world!":

```
hello
```

- Print "hello, world", the traditional type:

```
hello {{[-t|--traditional]}}
```

- Print a text message:

```
hello {{[-g|--greeting]}} "{{greeting_text}}"
```

helm install

Install a helm chart.

More information: https://helm.sh/docs/intro/using_helm/#helm-install-installing-a-package.

- Install a helm chart:

```
helm install {{name}} {{repository_name}}/{{chart_name}}
```

- Install a helm chart from an unpacked chart directory:

```
helm install {{name}} {{path/to/source_directory}}
```

- Install a helm chart from a URL:

```
helm install {{package_name}} {{https://example.com/charts/packagename-1.2.3.tgz}}
```

- Install a helm chart and generate a name:

```
helm install {{repository_name}}/{{chart_name}} {{[-g|--generate-name]}}
```

- Perform a dry run:

```
helm install {{name}} {{repository_name}}/{{chart_name}} --dry-run
```

- Install a helm chart with custom values:

```
helm install {{name}} {{repository_name}}/{{chart_name}} --set {{parameter1}}={{value1}},{{parameter2}}={{value2}}
```

- Install a helm chart passing a custom values file:

```
helm install {{name}} {{repository_name}}/{{chart_name}} {{[-f|--values]}} {{path/to/values.yaml}}
```

helm

A package manager for Kubernetes.

Some subcommands such as **`install`** have their own usage documentation.

More information: https://helm.sh/docs/helm/.

- Create a helm chart:

```
helm create {{chart_name}}
```

- Add a new helm repository:

```
helm repo add {{repository_name}}
```

- List helm repositories:

```
helm repo {{[ls|list]}}
```

- Update helm repositories:

```
helm repo {{[up|update]}}
```

- Delete a helm repository:

```
helm repo {{[rm|remove]}} {{repository_name}}
```

- Install a helm chart:

```
helm install {{name}} {{repository_name}}/{{chart_name}}
```

- Download helm chart as a tar archive:

```
helm get {{chart_release_name}}
```

- Update helm dependencies:

```
helm {{[dep|dependency]}} {{[up|update]}}
```

help

Display information about Bash builtin commands.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-help.

- Display the full list of builtin commands:

```
help
```

- Print instructions on how to use the `while` loop construct:

```
help while
```

- Print instructions on how to use the `for` loop construct:

```
help for
```

- Print instructions on how to use `[[ ]]` for conditional commands:

```
help [[ ]]
```

- Print instruction on how to use `(( ))` to evaluate arithmetic expressions:

```
help \( \)
```

- Print instructions on how to use the `cd` command:

```
help cd
```

help2man

Produce simple man pages from an executable's **`--help`** and **`--version`** output.

More information: https://www.gnu.org/software/help2man/#Invoking-help2man.

- Generate a man page for an executable:

```
help2man {{executable}}
```

- Specify the "name" paragraph in the man page:

```
help2man {{executable}} {{[-n|--name]}} {{name}}
```

- Specify the section for the man page (defaults to 1):

```
help2man {{executable}} {{[-s|--section]}} {{section}}
```

- Output to a file instead of `stdout`:

```
help2man {{executable}} {{[-o|--output]}} {{path/to/file}}
```

- Display help:

```
help2man --help
```

heroku

Create and manage Heroku apps.

More information: https://devcenter.heroku.com/articles/heroku-cli#get-started-with-the-heroku-cli.

- Log in to your Heroku account:

```
heroku login
```

- Create a Heroku app:

```
heroku create
```

- Show logs for an app:

```
heroku logs --app {{app_name}}
```

- Run a one-off process inside a dyno (Heroku virtual machine):

```
heroku run {{process_name}} --app {{app_name}}
```

- List dynos (Heroku virtual machines) for an app:

```
heroku ps --app {{app_name}}
```

- Permanently destroy an app:

```
heroku destroy --app {{app_name}}
```

hexdump

Display file contents in hexadecimal, decimal, octal, or ASCII.

Useful for inspecting dump file, binary data, or debug output.

See also: **`hexyl`**, **`od`**, **`xxd`**.

More information: https://manned.org/man/freebsd/hexdump.1.

- Print the hexadecimal representation of a file, replacing duplicate lines by *:

```
hexdump {{path/to/file}}
```

- Display the input offset in hexadecimal and its ASCII representation in two columns:

```
hexdump -C {{path/to/file}}
```

- Display the hexadecimal representation of a file, but interpret only a specific number of bytes of the input:

```
hexdump -C -n {{number_of_bytes}} {{path/to/file}}
```

- Verbose - no suppression by * on duplicate lines:

```
hexdump -v {{path/to/file}}
```

- Format output using printf-like format string:

```
hexdump -e '{{element_format .. end_format}}' {{path/to/file}}
```

hexedit

File editor for hexadecimal and ASCII content.

More information: https://manned.org/hexedit.

- Open file:

```
hexedit {{path/to/file}}
```

- Display file with sectors:

```
hexedit {{[-s|--sector]}} {{path/to/file}}
```

- Maximize file display:

```
hexedit {{[-m|--maximize]}} {{path/to/file}}
```

- Display help:

```
hexedit {{[-h|--help]}}
```

hexo

A fast, simple & powerful blog framework.

More information: https://hexo.io/docs/commands.

- Initialize a website:

  ```
  hexo init {{path/to/directory}}
  ```

- Create a new article:

  ```
  hexo new {{layout}} {{title}}
  ```

- Generate static files:

  ```
  hexo generate
  ```

- Start a local server:

  ```
  hexo server
  ```

- Deploy the website:

  ```
  hexo deploy
  ```

- Clean the cache file (`db.json`) and generated files (`public/`):

  ```
  hexo clean
  ```

hexyl

A simple hex viewer for the terminal. Uses colored output to distinguish different categories of bytes.

See also: **`od`**, **`xxd`**, **`hexdump`**.

More information: https://github.com/sharkdp/hexyl.

- Print the hexadecimal representation of a file:

```
hexyl {{path/to/file}}
```

- Print the hexadecimal representation of the first n bytes of a file:

```
hexyl {{[-n|--length]}} {{n}} {{path/to/file}}
```

hg add

Adds specified files to the staging area for the next commit in Mercurial.

More information: https://www.mercurial-scm.org/help/commands/add.

- Add files or directories to the staging area:

```
hg add {{path/to/file}}
```

- Add all unstaged files matching a specified pattern:

```
hg add {{[-I|--include]}} {{pattern}}
```

- Add all unstaged files, excluding those that match a specified pattern:

```
hg add {{[-X|--exclude]}} {{pattern}}
```

- Recursively add sub-repositories:

```
hg add {{[-S|--subrepos]}}
```

- Perform a test-run without performing any actions:

```
hg add {{[-n|--dry-run]}}
```

hg branch

Create or show a branch name.

More information: https://www.mercurial-scm.org/help/commands/branch.

- Show the name of the currently active branch:

```
hg branch
```

- Create a new branch for the next commit:

```
hg branch {{branch_name}}
```

hg clone

Create a copy of an existing repository in a new directory.

More information: https://www.mercurial-scm.org/help/commands/clone.

- Clone a repository to a specified directory:

```
hg clone {{remote_repository_source}} {{destination_path}}
```

- Clone a repository to the head of a specific branch, ignoring later commits:

```
hg clone {{[-b|--branch]}} {{branch}}
{{remote_repository_source}}
```

- Clone a repository with only the `.hg` directory, without checking out files:

```
hg clone {{[-U|--noupdate]}} {{remote_repository_source}}
```

- Clone a repository to a specific revision, tag or branch, keeping the entire history:

```
hg clone {{[-u|--updaterev]}} {{revision}}
{{remote_repository_source}}
```

- Clone a repository up to a specific revision without any newer history:

```
hg clone {{[-r|--rev]}} {{revision}}
{{remote_repository_source}}
```

hg commit

Commit all staged or specified files to the repository.

More information: https://www.mercurial-scm.org/help/commands/commit.

- Commit staged files to the repository:

```
hg commit
```

- Commit a specific file or directory:

```
hg commit {{path/to/file_or_directory}}
```

- Commit with a specific message:

```
hg commit {{[-m|--message]}} {{message}}
```

- Commit all files matching a specified pattern:

```
hg commit {{[-I|--include]}} {{pattern}}
```

- Commit all files, excluding those that match a specified pattern:

```
hg commit {{[-X|--exclude]}} {{pattern}}
```

- Commit using the interactive mode:

```
hg commit {{[-i|--interactive]}}
```

hg init

Create a new repository in the specified directory.

More information: https://www.mercurial-scm.org/help/commands/init.

- Initialize a new repository in the current directory:

```
hg init
```

- Initialize a new repository in the specified directory:

```
hg init {{path/to/directory}}
```

hg log

Display the revision history of the repository.

More information: https://www.mercurial-scm.org/help/commands/log.

- Display the entire revision history of the repository:

```
hg log
```

- Display the revision history with an ASCII graph:

```
hg log {{[-G|--graph]}}
```

- Display the revision history with file names matching a specified pattern:

```
hg log {{[-I|--include]}} {{pattern}}
```

- Display the revision history, excluding file names that match a specified pattern:

```
hg log {{[-X|--exclude]}} {{pattern}}
```

- Display the log information for a specific revision:

```
hg log {{[-r|--rev]}} {{revision}}
```

- Display the revision history for a specific branch:

```
hg log {{[-b|--branch]}} {{branch}}
```

- Display the revision history for a specific date:

```
hg log {{[-d|--date]}} {{date}}
```

- Display revisions committed by a specific user:

```
hg log {{[-u|--user]}} {{user}}
```

hg pull

Pull changes from a specified repository to the local repository.

More information: https://www.mercurial-scm.org/help/commands/pull.

- Pull from the "default" source path:

```
hg pull
```

- Pull from a specified source repository:

```
hg pull {{path/to/source_repository}}
```

- Update the local repository to the head of the remote:

```
hg pull {{[-u|--update]}}
```

- Pull changes even when the remote repository is unrelated:

```
hg pull {{[-f|--force]}}
```

- Specify a specific revision changeset to pull up to:

```
hg pull {{[-r|--rev]}} {{revision}}
```

- Specify a specific branch to pull:

```
hg pull {{[-b|--branch]}} {{branch}}
```

- Specify a specific bookmark to pull:

```
hg pull {{[-B|--bookmark]}} {{bookmark}}
```

hg push

Push changes from the local repository to a specified destination.

More information: https://www.mercurial-scm.org/help/commands/push.

- Push changes to the "default" remote path:

```
hg push
```

- Push changes to a specified remote repository:

```
hg push {{path/to/destination_repository}}
```

- Push a new branch if it does not exist (disabled by default):

```
hg push --new-branch
```

- Specify a specific revision changeset to push:

```
hg push {{[-r|--rev]}} {{revision}}
```

- Specify a specific branch to push:

```
hg push {{[-b|--branch]}} {{branch}}
```

- Specify a specific bookmark to push:

```
hg push {{[-B|--bookmark]}} {{bookmark}}
```

hg remove

Remove specified files from the staging area.

More information: https://www.mercurial-scm.org/help/commands/remove.

- Remove files or directories from the staging area:

```
hg remove {{path/to/file}}
```

- Remove all staged files matching a specified pattern:

```
hg remove {{[-I|--include]}} {{pattern}}
```

- Remove all staged files, excluding those that match a specified pattern:

```
hg remove {{[-X|--exclude]}} {{pattern}}
```

- Recursively remove sub-repositories:

```
hg remove {{[-S|--subrepos]}}
```

- Remove files from the repository that have been physically removed:

```
hg remove {{[-A|--after]}}
```

hg root

Display the root location of a Hg repository.

More information: https://www.mercurial-scm.org/help/commands/root.

- Display the root location of the current repository:

```
hg root
```

- Display the root location of the specified repository:

```
hg root --cwd {{path/to/directory}}
```

hg serve

Start a standalone Mercurial web server for browsing repositories.

More information: https://www.mercurial-scm.org/help/commands/serve.

- Start a web server instance:

```
hg serve
```

- Start a web server instance on the specified port:

```
hg serve {{[-p|--port]}} {{port}}
```

- Start a web server instance on the specified listening address:

```
hg serve {{[-a|--address]}} {{address}}
```

- Start a web server instance with a specific identifier:

```
hg serve {{[-n|--name]}} {{name}}
```

- Start a web server instance using the specified theme (see the templates directory):

```
hg serve --style {{style}}
```

- Start a web server instance using the specified SSL certificate bundle:

```
hg serve --certificate {{path/to/certificate}}
```

hg status

Show files that have changed in the working directory.

More information: https://www.mercurial-scm.org/help/commands/status.

- Display the status of changed files:

```
hg status
```

- Display only modified files:

```
hg status {{[-m|--modified]}}
```

- Display only added files:

```
hg status {{[-a|--added]}}
```

- Display only removed files:

```
hg status {{[-r|--removed]}}
```

- Display only deleted (but tracked) files:

```
hg status {{[-d|--deleted]}}
```

- Display changes in the working directory compared to a specified changeset:

```
hg status --rev {{revision}}
```

- Display only files matching a specified glob pattern:

```
hg status {{[-I|--include]}} {{pattern}}
```

- Display files, excluding those that match a specified glob pattern:

```
hg status {{[-X|--exclude]}} {{pattern}}
```

hg update

Update the working directory to a specified changeset.

More information: https://www.mercurial-scm.org/help/commands/update.

- Update to the tip of the current branch:

```
hg update
```

- Update to the specified revision:

```
hg update {{[-r|--rev]}} {{revision}}
```

- Update and discard uncommitted changes:

```
hg update {{[-C|--clean]}}
```

- Update to the last commit matching a specified date:

```
hg update {{[-d|--date]}} {{dd-mm-yyyy}}
```

hg

Mercurial - a distributed source control management system.

Some subcommands such as **`commit`** have their own usage documentation.

More information: https://www.mercurial-scm.org/help/commands.

- Create an empty Mercurial repository:

```
hg init
```

- Clone a remote Mercurial repository from the internet:

```
hg clone {{https://example.com/repo}}
```

- View the status of a local repository:

```
hg status
```

- Add all new files to the next commit:

```
hg add
```

- Commit changes to version history:

```
hg commit {{[-m|--message]}} {{message_text}}
```

- Push local changes to a remote repository:

```
hg push
```

- Pull any changes made to a remote:

```
hg pull
```

- Reset everything the way it was in the latest commit:

```
hg update {{[-C|--clean]}}; hg purge
```

highlight

Outputs syntax-highlighted source code to a variety of formats.

More information: http://andre-simon.de/doku/highlight/en/highlight.php.

- Produce a complete HTML document from a source code file:

```
highlight {{[-o|--out-format]}} {{html}} {{[-s|--style]}}
{{theme_name}} {{[-S|--syntax]}} {{language}} {{path/to/
source_code}}
```

- Produce an HTML fragment, suitable for inclusion in a larger document:

```
highlight {{[-o|--out-format]}} {{html}} {{[-f|--fragment]}}
{{[-S|--syntax]}} {{language}} {{source_file}}
```

- Inline the CSS styling in every tag:

```
highlight {{[-o|--out-format]}} {{html}} --inline-css {{[-
S|--syntax]}} {{language}} {{source_file}}
```

- List all supported languages, themes, or plugins:

```
highlight --list-scripts {{langs|themes|plugins}}
```

- Print a CSS stylesheet for a theme:

```
highlight {{[-o|--out-format]}} {{html}} --print-style {{[-
s|--style]}} {{theme_name}} {{[-S|--syntax]}} {{language}} --
stdout
```

hipstopgm

Read a HIPS file as input and return a PGM image as output.

If the HIPS file contains more than one frame in sequence, **`hipstopgm`** will concatenate all the frames vertically.

More information: https://netpbm.sourceforge.net/doc/hipstopgm.html.

- Convert a HIPS file into a PGM image:

```
hipstopgm {{path/to/file.hips}}
```

- Suppress all informational messages:

```
hipstopgm {{[-q|-quiet]}}
```

- Display version:

```
hipstopgm {{[-v|-version]}}
```

history

Manage command-line history.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-history.

- Display the commands history list with line numbers:

```
history
```

- Display the last 20 commands (in Zsh it displays all commands starting from the 20th):

```
history 20
```

- Display history with timestamps in different formats (only available in Zsh):

```
history -{{d|f|i|E}}
```

- [c]lear the commands history list:

```
history -c
```

- Over[w]rite history file with history of current Bash shell (often combined with `history -c` to purge history):

```
history -w
```

- [d]elete the history entry at the specified offset:

```
history -d {{offset}}
```

- Add a command to history without running it:

```
history -s {{command}}
```

- Run a command without adding it to history by adding a leading space:

```
<Space>{{command}}
```

hive

CLI tool for Apache Hive.

More information: https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Cli.

- Start a Hive interactive shell:

```
hive
```

- Run HiveQL:

```
hive -e "{{hiveql_query}}"
```

- Run a HiveQL file with a variable substitution:

```
hive {{[-d|--define]}} {{key}}={{value}} -f {{path/to/file.sql}}
```

- Run a HiveQL with HiveConfig (e.g. `mapred.reduce.tasks=32`):

```
hive --hiveconf {{conf_name}}={{conf_value}}
```

hledger accounts

List account names.

More information: https://hledger.org/hledger.html#accounts.

- Show all accounts used or declared in the default journal file:

```
hledger {{[acc|accounts]}}
```

- Show accounts used by transactions:

```
hledger {{[acc|accounts]}} {{[-u|--used]}}
```

- Show accounts declared with account directives:

```
hledger {{[acc|accounts]}} {{[-d|--declared]}}
```

- Add new account directives, for accounts used but not declared, to the journal:

```
hledger {{[acc|accounts]}} --undeclared --directives >>
{{2024-accounts.journal}}
```

- Show accounts with `asset` in their name, and their declared/inferred types:

```
hledger {{[acc|accounts]}} asset --types
```

- Show accounts of the `Asset` type:

```
hledger {{[acc|accounts]}} type:A
```

- Show the first two levels of the accounts hierarchy:

```
hledger {{[acc|accounts]}} {{[-t|--tree]}} {{[-2|--depth 2]}}
```

hledger add

Record new transactions with interactive prompting in the console.

More information: https://hledger.org/hledger.html#add.

- Record new transactions, saving to the default journal file:

```
hledger add
```

- Add transactions to `2024.journal`, but also load `2023.journal` for completions:

```
hledger add {{[-f|--file]}} {{path/to/2024.journal}} {{[-f|--file]}} {{path/to/2023.journal}}
```

- Provide answers for the first four prompts:

```
hledger add {{today}} '{{best buy}}' {{expenses:supplies}} '{{$20}}'
```

- Show `add`'s options and documentation with `$PAGER`:

```
hledger add {{[-h|--help]}}
```

- Show `add`'s documentation with `info` or `man` if available:

```
hledger help add
```

hledger aregister

Show the transactions and running balances in one account, with each transaction on one line.

More information: https://hledger.org/hledger.html#aregister.

- Show transactions and running balance in the `assets:bank:checking` account:

```
hledger {{[areg|aregister]}} assets:bank:checking
```

- Show transactions and running balance in the first account named `*savings*`:

```
hledger {{[areg|aregister]}} savings
```

- Show the checking account's cleared transactions, with a specified width:

```
hledger {{[areg|aregister]}} checking {{[-C|--cleared]}} {{[-w|--width]}} {{120}}
```

- Show the checking register, including transactions from forecast rules:

```
hledger {{[areg|aregister]}} checking --forecast
```

hledger balance

A flexible, general purpose "summing" report that shows accounts with some kind of numeric data.

This can be balance changes per period, end balances, budget performance, unrealised capital gains, etc.

More information: https://hledger.org/hledger.html#balance.

- Show the balance change in all accounts from all postings over all time:

```
hledger {{[bal|balance]}}
```

- Show the balance change in accounts named *expenses*, as a tree, summarising the top two levels only:

```
hledger {{[bal|balance]}} {{expenses}} {{[-t|--tree]}}
{{[-2|--depth 2]}}
```

- Show expenses each month, and their totals and averages, sorted by total; and their monthly budget goals:

```
hledger {{[bal|balance]}} {{expenses}} {{[-M|--monthly]}}
{{[-T|--row-total]}} {{[-A|--average]}} {{[-S|--sort-
amount]}} --budget
```

- Similar to the above, matching accounts by Expense type, as a two level tree without squashing boring accounts:

```
hledger {{[bal|balance]}} type:{{X}} {{[-MTAS|--monthly --
row-total --average --sort-amount]}} --budget {{[-t|--tree]}}
{{[-2|--depth 2]}} --no-elide
```

- Show end balances (including from postings before the start date), quarterly in 2024, in accounts named *assets* or *liabilities*:

```
hledger {{[bal|balance]}} {{[-H|--historical]}} {{[-p|--
period]}} '{{quarterly in 2024}}' {{assets}} {{liabilities}}
```

- Similar to the above, also show zero balances, sort by total and summarise to three levels:

```
hledger {{[bal|balance]}} {{[-HQ|--historical --quarterly]}}
date:{{2024}} type:{{AL}} {{[-ES|--empty --sort-amount]}}
{{[-3|--depth 3]}}
```

- Show investment assets' market [V]alue in base currency at the end of each quarter:

```
hledger {{[bal|balance]}} {{[-HVQ|--historical --market --
quarterly]}} {{assets:investments}}
```

- Show unrealised capital gains/losses from market price changes in each quarter, for non-cryptocurrency investment assets:

```
hledger {{[bal|balance]}} --gain {{[-Q|--quarterly]}}
{{assets:investments}} not:{{cryptocurrency}}
```

hledger balancesheet

Show the end balances in asset and liability accounts.

Amounts are shown with normal positive sign, as in conventional financial statements.

More information: https://hledger.org/hledger.html#balancesheet.

- Show the current balances in `Asset` and `Liability` accounts, excluding zeros:

```
hledger {{[bs|balancesheet]}}
```

- Show just the liquid assets (`Cash` account type):

```
hledger {{[bs|balancesheet]}} type:C
```

- Include accounts with zero balances, and show the account hierarchy:

```
hledger {{[bs|balancesheet]}} {{[-E|--empty]}} {{[-t|--tree]}}
```

- Show the balances at the end of each month:

```
hledger {{[bs|balancesheet]}} {{[-M|--monthly]}}
```

- Show the balances' market value in home currency at the end of each month:

```
hledger {{[bs|balancesheet]}} {{[-M|--monthly]}} {{[-V|--market]}}
```

- Show quarterly balances, with just the top two levels of account hierarchy:

```
hledger {{[bs|balancesheet]}} {{[-Qt|--quarterly --tree]}} {{[-2|--depth 2]}}
```

- Same as above above, and generate HTML output in `bs.html`:

```
hledger {{[bs|balancesheet]}} {{[-Qt|--quarterly --tree]}} {{[-2|--depth 2]}} {{[-o|--output-file]}} bs.html
```

hledger import

Import new transactions from one or more data files to the main journal.

More information: https://hledger.org/hledger.html#import.

- Import new transactions from `bank.csv`, using `bank.csv.rules` to convert:

```
hledger import {{path/to/bank.csv}}
```

- Show what would be imported from these two files, without doing anything:

```
hledger import {{path/to/bank1.csv}} {{path/to/bank2.csv}} --dry-run
```

- Import new transactions from all CSV files, using the same rules for all:

```
hledger import --rules-file {{common.rules}} *.csv
```

- Show conversion errors or results while editing `bank.csv.rules`:

```
watchexec -- hledger {{[-f|--file]}} {{path/to/bank.csv}} print
```

- Mark `bank.csv`'s current data as seen, as if already imported:

```
hledger import --catchup {{path/to/bank.csv}}
```

- Mark `bank.csv` as all new, as if not yet imported:

```
rm {{[-f|--force]}} .latest.bank.csv
```

hledger incomestatement

Show revenue inflows and expense outflows during the report period.

Amounts are shown with normal positive sign, as in conventional financial statements.

More information: https://hledger.org/hledger.html#incomestatement.

- Show revenues and expenses (changes in `Revenue` and `Expense` accounts):

```
hledger {{[is|incomestatement]}}
```

- Show revenues and expenses each month:

```
hledger {{[is|incomestatement]}} {{[-M|--monthly]}}
```

- Show monthly revenues/expenses/totals, largest first, summarised to 2 levels:

```
hledger {{[is|incomestatement]}} {{[-MTAS|--monthly --row-
total --average --sort-amount]}} {{[-2|--depth 2]}}
```

- Same as above, and generate HTML output in `is.html`:

```
hledger {{[is|incomestatement]}} {{[-MTAS|--monthly --row-
total --average --sort-amount]}} {{[-2|--depth 2]}} {{[-o|--
output-file]}} is.html
```

hledger print

Show full journal entries, representing transactions.

More information: https://hledger.org/hledger.html#print.

- Show all transactions in the default journal file:

```
hledger print
```

- Show transactions, with any implied amounts or costs made explicit:

```
hledger print {{[-x|--explicit]}} --infer-costs
```

- Show transactions from two specified files, with amounts converted to cost:

```
hledger print {{[-f|--file]}} {{path/to/2023.journal}} {{[-f|--file]}} {{path/to/2024.journal}} {{[-B|--cost]}}
```

- Show `$` transactions in `*food*` but not `*groceries*` accounts this month:

```
hledger print cur:\\$ food not:groceries date:thismonth
```

- Show transactions of amount 50 or more, with `whole foods` in their description:

```
hledger print amt:'>50' desc:'whole foods'
```

- Show cleared transactions, with `EUR` amounts rounded and with decimal commas:

```
hledger print {{[-C|--cleared]}} --commodity '1000, EUR' --round hard
```

- Write transactions from `foo.journal` as a CSV file:

```
hledger print {{[-f|--file]}} {{path/to/foo.journal}} {{[-o|--output-file]}} {{path/to/output_file.csv}}
```

hledger-ui

A terminal interface (TUI) for **`hledger`**, a robust, friendly plain text accounting app.

More information: https://hledger.org/hledger-ui.html.

- Start in the main menu screen, reading from the default journal file:

```
hledger-ui
```

- Start with a different color theme:

```
hledger-ui --theme {{terminal|greenterm|dark}}
```

- Start in the balance sheet accounts screen, showing hierarchy down to level 3:

```
hledger-ui --bs {{[-t|--tree]}} {{[-3|--depth 3]}}
```

- Start in this account's screen, showing cleared transactions, and reload on change:

```
hledger-ui --register {{assets:bank:checking}} {{[-C|--cleared]}} {{[-w|--watch]}}
```

- Read two journal files, and show amounts as current value when known:

```
hledger-ui {{[-f|--file]}} {{path/to/2024.journal}} {{[-f|--file]}} {{path/to/2024-prices.journal}} --value now
```

- Show the manual in Info format, if possible:

```
hledger-ui --info
```

- Display help:

```
hledger-ui {{[-h|--help]}}
```

hledger-web

A web interface and API for **`hledger`**, a robust, friendly plain text accounting app.

More information: https://hledger.org/hledger-web.html.

- Start the web app, and a browser if possible, for local viewing and adding only:

  ```
  hledger-web
  ```

- As above but with a specified file, and allow editing of existing data:

  ```
  hledger-web {{[-f|--file]}} {{path/to/file.journal}} --allow edit
  ```

- Start just the web app, and accept incoming connections to the specified host and port:

  ```
  hledger-web --serve --host {{my.host.name}} --port 8000
  ```

- Start just the web app's JSON API, and allow only read access:

  ```
  hledger-web --serve-api --host {{my.host.name}} --allow view
  ```

- Show amounts converted to current market value in your base currency when known:

  ```
  hledger-web --value now --infer-market-prices
  ```

- Show the manual in Info format if possible:

  ```
  hledger-web --info
  ```

- Display help:

  ```
  hledger-web {{[-h|--help]}}
  ```

hledger

A robust, friendly plain text accounting app.

See also: **`hledger-ui`** for TUI, **`hledger-web`** for web interface.

More information: https://hledger.org/hledger.html.

- Record new transactions interactively, saving to the default journal file:

```
hledger add
```

- Import new transactions from `bank.csv`, using `bank.csv.rules` to convert:

```
hledger import {{path/to/bank.csv}}
```

- Print all transactions, reading from multiple specified journal files:

```
hledger print {{[-f|--file]}} {{path/to/prices-2024.journal}}
{{[-f|--file]}} {{path/to/prices-2023.journal}}
```

- Show all accounts, as a hierarchy, and their types:

```
hledger accounts {{[-t|--tree]}} --types
```

- Show asset and liability account balances, including zeros, hierarchically:

```
hledger {{[bs|balancesheet]}} {{[-E|--empty]}} {{[-t|--
tree]}} --no-elide
```

- Show monthly incomes/expenses/totals, largest first, summarised to 2 levels:

```
hledger {{[is|incomestatement]}} {{[-M|--monthly]}} {{[-T|--
row-total]}} {{[-A|--average]}} --sort {{[-2|--depth 2]}}
```

- Show the `assets:bank:checking` account's transactions and running balance:

```
hledger {{[areg|aregister]}} assets:bank:checking
```

- Show the amount spent on food from the `assets:cash` account:

```
hledger print assets:cash | hledger {{[-f|--file]}} - {{[-
I|--ignore-assertions]}} aregister expenses:food
```

hn

Command-line interface for Hacker News.

More information: https://github.com/rafaelrinaldi/hn-cli#usage.

- View stories on Hacker News:

```
hn
```

- View number of stories on Hacker News:

```
hn {{[-l|--limit]}} {{number}}
```

- View stories on Hacker News, and keep the list open after selecting a link:

```
hn {{[-k|--keep-open]}}
```

- View stories on Hacker News sorted by submission date:

```
hn --latest
```

holehe

Check if an email is attached to an account on sites like Twitter, Instagram, Imgur and over 120 others.

More information: https://github.com/megadose/holehe#-cli-example.

- Show status across all supported websites for the specified email address:

```
holehe {{username@example.org}}
```

- Show status for only sites where the specified email address is in use:

```
holehe {{username@example.org}} --only-used
```

home-manager

Manage a per-user environment using Nix, allowing declarative configuration of the user's home.

More information: https://github.com/nix-community/home-manager.

- Build the configuration defined in `~/.config/nixpkgs/home.nix` without applying it:

  ```
  home-manager build
  ```

- Build and apply (switch to) the new configuration:

  ```
  home-manager switch
  ```

- Roll back to a previous configuration generation:

  ```
  home-manager rollback
  ```

- List all existing configuration generations:

  ```
  home-manager generations
  ```

- When using flakes, run any operation that requires nix to run (build, switch, news) by passing the path to the flake:

  ```
  home-manager {{command}} --flake {{path/to/flake}}
  ```

host

Lookup Domain Name Server.

More information: https://manned.org/host.

- Lookup A, AAAA, and MX records of a domain:

```
host {{domain}}
```

- Lookup a field (CNAME, TXT, ...) of a domain:

```
host -t {{field}} {{domain}}
```

- Reverse lookup an IP:

```
host {{ip_address}}
```

- Specify an alternate DNS server to query:

```
host {{domain}} {{8.8.8.8}}
```

hostapd

Start an access point using a wireless interface.

More information: https://manned.org/hostapd.

- Start an access point:

```
sudo hostapd {{path/to/hostapd.conf}}
```

- Start an access point, forking into the background:

```
sudo hostapd -B {{path/to/hostapd.conf}}
```

hostess

Manage the **`/etc/hosts`** file.

More information: https://github.com/cbednarski/hostess.

- List domains, target IP addresses and on/off status:

  `hostess list`

- Add a domain pointing to your machine to your hosts file:

  `hostess add {{local.example.com}} {{127.0.0.1}}`

- Remove a domain from your hosts file:

  `hostess del {{local.example.com}}`

- Disable a domain (but don't remove it):

  `hostess off {{local.example.com}}`

hostid

Print the numeric identifier for the current host (not necessarily the IP address).

More information: https://www.gnu.org/software/coreutils/manual/html_node/hostid-invocation.html.

- Display the numeric identifier for the current host in hexadecimal:

```
hostid
```

hostname

Show or set the system's host name.

More information: https://manned.org/hostname.

- Show current host name:

```
hostname
```

- Show the network address of the host name:

```
hostname {{[-i|--ip-address]}}
```

- Show the FQDN (Fully Qualified Domain Name):

```
hostname {{[-f|--fqdn]}}
```

- Set current host name:

```
hostname {{new_hostname}}
```

hping

This command is an alias of **`hping3`**.

- View documentation for the original command:

```
tldr hping3
```

hping3

Advanced ping utility which supports protocols such TCP, UDP, and raw IP.

Best run with elevated privileges.

More information: https://github.com/antirez/hping.

- Ping a destination with 4 ICMP ping requests:

```
hping3 --icmp --count {{4}} {{ip_or_hostname}}
```

- Ping an IP address over UDP on port 80:

```
hping3 --udp --destport {{80}} --syn {{ip_or_hostname}}
```

- Scan TCP port 80, scanning from the specific local source port 5090:

```
hping3 --verbose --syn --destport {{80}} --baseport {{5090}}
{{ip_or_hostname}}
```

- Traceroute using a TCP scan to a specific destination port:

```
hping3 --traceroute --verbose --syn --destport {{80}}
{{ip_or_hostname}}
```

- Scan a set of TCP ports on a specific IP address:

```
hping3 --scan {{80,3000,9000}} --syn {{ip_or_hostname}}
```

- Perform a TCP ACK scan to check if a given host is alive:

```
hping3 --count {{2}} --verbose --destport {{80}} --ack
{{ip_or_hostname}}
```

- Perform a charge test on port 80:

```
hping3 --flood --destport {{80}} --syn {{ip_or_hostname}}
```

hr

Print a horizontal rule in the terminal.

More information: https://github.com/LuRsT/hr.

- Print a horizontal rule:

```
hr
```

- Print a horizontal rule with a custom string:

```
hr {{string}}
```

- Print a multiline horizontal rule:

```
hr {{string1 string2 ...}}
```

hsd-cli

The REST tool for the Handshake blockchain.

More information: https://handshake.org.

- Retrieve information about the current server:

  ```
  hsd-cli info
  ```

- Broadcast a local transaction:

  ```
  hsd-cli broadcast {{transaction_hex}}
  ```

- Retrieve a mempool snapshot:

  ```
  hsd-cli mempool
  ```

- View a transaction by address or hash:

  ```
  hsd-cli tx {{address_or_hash}}
  ```

- View a coin by its hash index or address:

  ```
  hsd-cli coin {{hash_index_or_address}}
  ```

- View a block by height or hash:

  ```
  hsd-cli block {{height_or_hash}}
  ```

- Reset the chain to the specified block:

  ```
  hsd-cli reset {{height_or_hash}}
  ```

- Execute an RPC command:

  ```
  hsd-cli rpc {{command}} {{args}}
  ```

hsw-cli

The REST tool for the Handshake wallet.

More information: https://github.com/handshake-org/hs-client.

- Unlock the current wallet (timeout in seconds):

```
hsw-cli unlock {{passphrase}} {{timeout}}
```

- Lock the current wallet:

```
hsw-cli lock
```

- View the current wallet's details:

```
hsw-cli get
```

- View the current wallet's balance:

```
hsw-cli balance
```

- View the current wallet's transaction history:

```
hsw-cli history
```

- Send a transaction with the specified coin amount to an address:

```
hsw-cli send {{address}} {{1.05}}
```

- View the current wallet's pending transactions:

```
hsw-cli pending
```

- View details about a transaction:

```
hsw-cli tx {{transaction_hash}}
```

html5validator

Validate HTML5.

More information: https://github.com/svenkreiss/html5validator.

- Validate a specific file:

```
html5validator {{path/to/file}}
```

- Validate all HTML files in a specific directory:

```
html5validator --root {{path/to/directory}}
```

- Show warnings as well as errors:

```
html5validator --show-warnings {{path/to/file}}
```

- Match multiple files using a glob pattern:

```
html5validator --root {{path/to/directory}} --match "{{*.html
*.php}}"
```

- Ignore specific directory names:

```
html5validator --root {{path/to/directory}} --blacklist
"{{node_modules vendor}}"
```

- Output the results in a specific format:

```
html5validator --format {{gnu|xml|json|text}} {{path/to/
file}}
```

- Output the log at a specific verbosity level:

```
html5validator --root {{path/to/directory}} --log {{debug|
info|warning}}
```

htmlq

Use CSS selectors to extract content from HTML files.

More information: https://github.com/mgdm/htmlq.

- Return all elements of class `card`:

```
cat {{path/to/file.html}} | htmlq '.card'
```

- Get the text content of the first paragraph:

```
cat {{path/to/file.html}} | htmlq --text 'p:first-of-type'
```

- Find all the links in a page:

```
cat {{path/to/file.html}} | htmlq --attribute href 'a'
```

- Remove all images and SVGs from a page:

```
cat {{path/to/file.html}} | htmlq --remove-nodes 'img' --
remove-nodes 'svg'
```

- Pretty print and write the output to a file:

```
htmlq --pretty --filename {{path/to/input.html}} --output
{{path/to/output.html}}
```

htop

Display dynamic real-time information about running processes.

An enhanced version of **`top`**.

See also: **`top`**, **`atop`**, **`glances`**, **`btop`**, **`btm`**.

More information: https://manned.org/htop.

- Start `htop`:

```
htop
```

- Start `htop` displaying processes owned by a specific user:

```
htop {{[-u|--user]}} {{username}}
```

- Display processes hierarchically in a tree view to show the parent-child relationships:

```
htop {{[-t|--tree]}}
```

- Sort processes by a specified `sort_item` (use `htop --sort help` for available options):

```
htop {{[-s|--sort]}} {{sort_item}}
```

- Start `htop` with the specified delay between updates, in tenths of a second (i.e. 50 = 5 seconds):

```
htop {{[-d|--delay]}} {{50}}
```

- Disable all system and process changing features:

```
htop --readonly
```

- See interactive commands while running `htop`:

```
{{<F1>|<?>}}
```

- Switch to a different tab:

```
<Tab>
```

htpasswd

Create and manage htpasswd files to protect web server directories using basic authentication.

More information: https://httpd.apache.org/docs/current/programs/htpasswd.html.

- Create/overwrite htpasswd file:

```
htpasswd -c {{path/to/file}} {{username}}
```

- Add user to htpasswd file or update existing user:

```
htpasswd {{path/to/file}} {{username}}
```

- Add user to htpasswd file in batch mode without an interactive password prompt (for script usage):

```
htpasswd -b {{path/to/file}} {{username}} {{password}}
```

- Delete user from htpasswd file:

```
htpasswd -D {{path/to/file}} {{username}}
```

- Verify user password:

```
htpasswd -v {{path/to/file}} {{username}}
```

- Display a string with username (plain text) and password (md5):

```
htpasswd -nbm {{username}} {{password}}
```

http-server-upload

Zero-configuration HTTP server which provides a lightweight interface to upload files.

More information: https://github.com/crycode-de/http-server-upload.

- Start an HTTP server on the default port to upload files to the current directory:

```
http-server-upload
```

- Start an HTTP server with the specified maximum allowed file size for uploads in MiB (defaults to 200 MiB):

```
MAX_FILE_SIZE={{size_in_megabytes}} http-server-upload
```

- Start an HTTP server on a specific port to upload files to the current directory:

```
PORT={{port}} http-server-upload
```

- Start an HTTP server, storing the uploaded files in a specific directory:

```
UPLOAD_DIR={{path/to/directory}} http-server-upload
```

- Start an HTTP server using a specific directory to temporarily store files during the upload process:

```
UPLOAD_TMP_DIR={{path/to/directory}} http-server-upload
```

- Start an HTTP server accepting uploads with a specific token field in the HTTP post:

```
TOKEN={{secret}} http-server-upload
```

http-server

Simple static HTTP server to serve static files.

More information: https://github.com/http-party/http-server.

- Start an HTTP server listening on the default port to serve the current directory:

```
http-server
```

- Start an HTTP server on a specific port to serve a specific directory:

```
http-server {{path/to/directory}} {{[-p|--port]}} {{port}}
```

- Start an HTTP server using basic authentication:

```
http-server --username {{username}} --password {{password}}
```

- Start an HTTP server with directory listings disabled:

```
http-server -d {{false}}
```

- Start an HTTPS server on the default port using the specified certificate:

```
http-server {{[-S|--ssl]}} {{[-C|--cert]}} {{path/to/
cert.pem}} {{[-K|--key]}} {{path/to/key.pem}}
```

- Start an HTTP server and include the client's IP address in the output logging:

```
http-server --log-ip
```

- Start an HTTP server with CORS enabled by including the `Access-Control-Allow-Origin: *` header in all responses:

```
http-server --cors
```

- Start an HTTP server with logging disabled:

```
http-server {{[-s|--silent]}}
```

http

HTTPie: an HTTP client designed for testing, debugging, and generally interacting with APIs and HTTP servers.

More information: https://httpie.io/docs/cli/usage.

- Make a simple GET request (shows response headers and content):

```
http {{https://example.com}}
```

- Print specific parts of the content (H: request headers, B: request body, h: response headers, b: response body, m: response metadata):

```
http {{[-p|--print]}} {{H|B|h|b|m|Hh|Hhb|...}} {{https://example.com}}
```

- Specify the HTTP method when sending a request and use a proxy to intercept the request:

```
http {{GET|POST|HEAD|PUT|PATCH|DELETE|...}} --proxy {{http|https}}:{{http://localhost:8080|socks5://localhost:9050|...}} {{https://example.com}}
```

- Follow any 3xx redirects and specify additional headers in a request:

```
http {{[-F|--follow]}} {{https://example.com}} {{'User-Agent: Mozilla/5.0' 'Accept-Encoding: gzip'}}
```

- Authenticate to a server using different authentication methods:

```
http {{[-a|--auth]}} {{username:password|token}} {{[-A|--auth-type]}} {{basic|digest|bearer}} {{GET|POST|...}} {{https://example.com/auth}}
```

- Construct a request but do not send it (similar to a dry-run):

```
http --offline {{GET|DELETE|...}} {{https://example.com}}
```

- Use named sessions for persistent custom headers, auth credentials and cookies:

```
http --session {{session_name|path/to/session.json}} {{[-a|--auth]}} {{username}}:{{password}} {{https://example.com/auth}} {{API-KEY:xxx}}
```

- Upload a file to a form (the example below assumes that the form field is `<input type="file" name="cv" />`):

```
http {{[-f|--form]}} {{POST}} {{https://example.com/upload}}
{{cv@path/to/file}}
```

httpflow

A utility to capture and dump HTTP streams.

More information: https://github.com/six-ddc/httpflow.

- Capture traffic on all interfaces:

```
httpflow -i {{any}}
```

- Use a bpf-style capture to filter the results:

```
httpflow {{host httpbin.org or host baidu.com}}
```

- Use a `regex` to filter requests by URLs:

```
httpflow -u '{{regex}}'
```

- Read packets from PCAP format binary file:

```
httpflow -r {{out.cap}}
```

- Write the output to a directory:

```
httpflow -w {{path/to/directory}}
```

httpie

Management interface for HTTPie.

See also: **`http`**, the tool itself.

More information: https://httpie.io/docs/cli/plugin-manager.

- Check updates for `http`:

```
httpie cli check-updates
```

- List installed `http` plugins:

```
httpie cli plugins list
```

- Install/upgrade/uninstall plugins:

```
httpie cli plugins {{install|upgrade|uninstall}}
{{plugin_name}}
```

httping

Measure the latency and throughput of a web server.

More information: https://manned.org/httping.

- Ping the specified URL:

```
httping -g {{url}}
```

- Ping the web server on `host` and `port`:

```
httping -h {{host}} -p {{port}}
```

- Ping the web server on `host` using a TLS connection:

```
httping -l -g https://{{host}}
```

- Ping the web server on `host` using HTTP basic authentication:

```
httping -g http://{{host}} -U {{username}} -P {{password}}
```

httprobe

Take a list of domains and probe for working HTTP and HTTPS servers.

More information: https://github.com/tomnomnom/httprobe.

- Probe a list of domains from a text file:

```
cat {{input_file}} | httprobe
```

- Only check for HTTP if HTTPS is not working:

```
cat {{input_file}} | httprobe --prefer-https
```

- Probe additional ports with a given protocol:

```
cat {{input_file}} | httprobe -p {{https:2222}}
```

- Display help:

```
httprobe --help
```

httpry

A lightweight packet sniffer for displaying and logging HTTP traffic.

It can be run in real-time displaying the traffic as it is parsed, or as a daemon process that logs to an output file.

More information: https://dumpsterventures.com/jason/httpry/.

- Save output to a file:

```
httpry -o {{path/to/file.log}}
```

- Listen on a specific interface and save output to a binary PCAP format file:

```
httpry {{eth0}} -b {{path/to/file.pcap}}
```

- Filter output by a comma-separated list of HTTP verbs:

```
httpry -m {{get|post|put|head|options|delete|trace|connect|patch}}
```

- Read from an input capture file and filter by IP:

```
httpry -r {{path/to/file.log}} '{{host 192.168.5.25}}'
```

- Run as daemon process:

```
httpry -d -o {{path/to/file.log}}
```

https

This command is an alias of **`http`**.

- View documentation for the original command:

```
tldr http
```

httpx

A fast and multi-purpose HTTP toolkit written in Go to run multiple probes at once.

Note: Not to be confused with the unrelated Python's HTTPX which has the same command name.

More information: https://docs.projectdiscovery.io/tools/httpx/running.

- Run a probe against a [u]RL, host, IP Address or subnet (CIDR notation) showing probe status:

```
httpx -probe {{[-u|-target]}} {{url|host|ipaddress|subnet_with_cidr}}
```

- Run a probe against multiple hosts showing status code with input from `subfinder`:

```
subfinder {{[-d|-domain]}} {{example.com}} | httpx {{[-sc|-status-code]}}
```

- Run a rate limited probe against a list of hosts from a file showing technology detected and response time:

```
httpx {{[-rl|-rate-limit]}} {{150}} {{[-l|-list]}} {{path/to/newline_separated_hosts_list}} {{[-td|-tech-detect]}} {{[-rt|-response-time]}}
```

- Run a probe against a [u]RL showing its webpage title, CDN/WAF in use, and page content hash:

```
httpx {{[-u|-target]}} {{url}} -title -cdn -hash {{sha256}}
```

- Run a probe against a list of hosts with custom defined ports and timeout after certain seconds:

```
httpx -probe {{[-u|-target]}} {{host1,host2,...}} {{[-p|-ports]}} http:{{80,8000-8080}},https:{{443,8443}} -timeout {{10}}
```

- Run a probe against a list of hosts filtering out codes of certain responses:

```
httpx {{[-u|-target]}} {{host1,host2,...}} {{[-fc|-filter-code]}} {{400,401,404}}
```

- Run a probe against a list of hosts matching codes of certain responses:

```
httpx {{[-u|-target]}} {{host1,host2,...}} {{[-mc|-match-
code]}} {{200,301,304}}
```

- Run a probe against a URL saving screenshots of certain paths, with screenshot timeouts (assets are saved in `./output`):

```
httpx {{[-u|-target]}} {{https://www.github.com}} -path {{/
tldr-pages/tldr,/projectdiscovery/httpx}} {{[-ss|-
screenshot]}} {{[-st|-screenshot-timeout]}} {{10}}
```

hub branch

Create a branch or show current branch.

See also: **`git branch`**.

- Show the name of the currently active branch:

  ```
  hub branch
  ```

- Create a new branch:

  ```
  hub branch {{branch_name}}
  ```

hub browse

Open a GitHub repository in the browser or print the URL.

More information: https://hub.github.com/hub-browse.1.html.

- Open the homepage of the current repository in the default web browser:

  ```
  hub browse
  ```

- Open the homepage of a specific repository in the default web browser:

  ```
  hub browse {{owner}}/{{repository}}
  ```

- Open the subpage of a specific repository in the default web browser, subpage can be "wiki", "commits", "issues", or other (default: "tree"):

  ```
  hub browse {{owner}}/{{repository}} {{subpage}}
  ```

hub ci-status

Display status of GitHub checks.

More information: https://hub.github.com/hub-ci-status.1.html.

- Check the CI status for this branch:

```
hub ci-status {{[-v|--verbose]}}
```

- Display status of GitHub checks for a commit:

```
hub ci-status {{[-v|--verbose]}} {{commit_SHA}}
```

hub clone

Clone an existing repository.

More information: https://hub.github.com/hub-clone.1.html.

- Clone an existing repository to current directory (If run into authentication problem, try full SSH path):

```
hub clone {{remote_repository_location}}
```

hub create

Create a new repository on GitHub.

More information: https://hub.github.com/hub-create.1.html.

- Upload the current (local-only) repository to your GitHub account as public:

```
hub create
```

- Create a private repository and open the new repository in a web browser:

```
hub create {{[-p|--private]}} {{[-o|--browse]}}
```

hub delete

Delete an existing repository on GitHub.

More information: https://hub.github.com/hub-delete.1.html.

- Delete personal repo on GitHub:

```
hub delete {{repo}}
```

hub fork

Fork a GitHub repo. Like **`git fork`** from **`git-extras`**.

More information: https://hub.github.com/hub-fork.1.html.

- Fork a GitHub repository by its slug:

```
hub fork {{tldr-pages/tldr}}
```

- Fork a GitHub repository by its URL:

```
hub fork {{https://github.com/tldr-pages/tldr}}
```

- Fork current GitHub repository, set remote name to origin:

```
hub fork --remote-name {{origin}}
```

hub init

Initialize a new local Git repository.

More information: https://hub.github.com/hub-init.1.html.

- Initialize a new local repository:

```
hub init
```

hub issue

Manage Github issues.

More information: https://hub.github.com/hub-issue.1.html.

- List the last 10 issues with the bug label:

```
hub issue list {{[-L|--limit]}} {{10}} {{[-l|--labels]}}
"{{bug}}"
```

- Display a specific issue:

```
hub issue show {{issue_number}}
```

- List 10 closed issues assigneed to a specific user:

```
hub issue {{[-s|--state]}} {{closed}} {{[-a|--assignee]}}
{{username}} --limit {{10}}
```

hub

A wrapper for Git that adds commands for working with GitHub-based projects.

If set up as instructed by **`hub alias`**, one can use **`git`** to run **`hub`** commands.

More information: https://hub.github.com/hub.1.html.

- Clone a repository using its slug (owners can omit the username):

  ```
  hub clone {{username}}/{{repo_name}}
  ```

- Create a fork of the current repository (cloned from another user) under your GitHub profile:

  ```
  hub fork
  ```

- Push the current local branch to GitHub and create a PR for it in the original repository:

  ```
  hub push {{remote_name}} && hub pull-request
  ```

- Create a PR of the current (already pushed) branch, reusing the message from the first commit:

  ```
  hub pull-request --no-edit
  ```

- Create a new branch with the contents of a pull request and switch to it:

  ```
  hub pr checkout {{pr_number}}
  ```

- Upload the current (local-only) repository to your GitHub account:

  ```
  hub create
  ```

- Fetch Git objects from upstream and update local branches:

  ```
  hub sync
  ```

huggingface-cli

Interact with Hugging Face Hub.

Login, manage local cache, download or upload files.

More information: https://huggingface.co/docs/huggingface_hub/guides/cli.

- Login to Hugging Face Hub:

```
huggingface-cli login
```

- Display the name of the logged in user:

```
huggingface-cli whoami
```

- Log out:

```
huggingface-cli logout
```

- Print information about the environment:

```
huggingface-cli env
```

- Download files from an repository and print out the path (omit filenames to download entire repository):

```
huggingface-cli download --repo-type {{repo_type}} {{repo_id}} {{filename1 filename2 ...}}
```

- Upload an entire folder or a file to Hugging Face:

```
huggingface-cli upload --repo-type {{repo_type}} {{repo_id}} {{path/to/local_file_or_directory}} {{path/to/repo_file_or_directory}}
```

- Scan cache to see downloaded repositories and their disk usage:

```
huggingface-cli scan-cache
```

- Delete the cache interactively:

```
huggingface-cli delete-cache
```

hugo server

Build and serve a site with Hugo's built-in webserver.

More information: https://gohugo.io/commands/hugo_server/.

- Build and serve a site:

```
hugo server
```

- Build and serve a site on a specified port number:

```
hugo server {{[-p|--port]}} {{port_number}}
```

- Build and serve a site while minifying supported output formats (HTML, XML, etc.):

```
hugo server --minify
```

- Build and serve a site in the production environment with full re-renders while minifying supported formats:

```
hugo server {{[-e|--environment]}} {{production}} --
disableFastRender --minify
```

- Display help:

```
hugo server {{[-h|--help]}}
```

hugo

Template-based static site generator. Uses modules, components, and themes.

Some subcommands such as **`server`** have their own usage documentation.

More information: https://gohugo.io/commands/.

- Create a new Hugo site:

```
hugo new site {{path/to/site}}
```

- Create a new Hugo theme (themes may also be downloaded from https://themes.gohugo.io/):

```
hugo new theme {{theme_name}}
```

- Create a new page:

```
hugo new {{section_name}}/{{page_name}}
```

- Build a site to the `./public/` directory:

```
hugo
```

- Build a site including pages that are marked as a "draft":

```
hugo {{[-D|--buildDrafts]}}
```

- Build a site on your local IP:

```
hugo server --bind {{local_ip}} {{[-b|--baseURL]}} {{http://local_ip}}
```

- Build a site to a given directory:

```
hugo {{[-d|--destination]}} {{path/to/destination}}
```

- Build a site, start up a webserver to serve it, and automatically reload when pages are edited:

```
hugo server
```

hunspell

Check spelling.

More information: https://github.com/hunspell/hunspell.

- Check the spelling of a file:

```
hunspell {{path/to/file}}
```

- Check the spelling of a file with the en_US dictionary:

```
hunspell -d {{en_US}} {{path/to/file}}
```

- List misspelled words in a file:

```
hunspell -l {{path/to/file}}
```

husky

Native Git hooks made easy.

More information: https://typicode.github.io/husky.

- Install Husky in the current directory:

```
husky install
```

- Install Husky into a specific directory:

```
husky install {{path/to/directory}}
```

- Set a specific command as a `pre-push` hook for Git:

```
husky set {{.husky/pre-push}} "{{command}}
{{command_arguments}}"
```

- Add a specific command to the current `pre-commit` hook:

```
husky add {{.husky/pre-commit}} "{{command}}
{{command_arguments}}"
```

- Uninstall Husky hooks from the current directory:

```
husky uninstall
```

- Display help:

```
husky
```

hut

A CLI tool for sourcehut.

More information: https://manned.org/hut.

- Initialize `hut`'s configuration file (this will prompt for an OAuth2 access token, which is required to use `hut`):

```
hut init
```

- List Git/Mercurial repositories:

```
hut {{git|hg}} list
```

- Create a public Git/Mercurial repository:

```
hut {{git|hg}} create {{name}}
```

- List jobs on https://builds.sr.ht:

```
hut builds list
```

- Show the status of a job:

```
hut builds show {{job_id}}
```

- SSH into a job container:

```
hut ssh {{job_id}}
```

hx

This command is an alias of **helix**.

- View documentation for the original command:

```
tldr helix
```

hydra

Online password guessing tool.

Protocols supported include FTP, HTTP(S), SMTP, SNMP, XMPP, SSH, and more.

More information: https://manned.org/hydra.

- Start Hydra's wizard:

```
hydra-wizard
```

- Guess SSH credentials using a given username and a list of passwords:

```
hydra -l {{username}} -P {{path/to/wordlist.txt}} {{host_ip}}
{{ssh}}
```

- Guess HTTPS webform credentials using two specific lists of usernames and passwords ("https_post_request" can be like "username=^USER^&password=^PASS^"):

```
hydra -L {{path/to/usernames.txt}} -P {{path/to/
wordlist.txt}} {{host_ip}} {{https-post-form}}
"{{url_without_host}}:{{https_post_request}}:
{{login_failed_string}}"
```

- Guess FTP credentials using usernames and passwords lists, specifying the number of threads:

```
hydra -L {{path/to/usernames.txt}} -P {{path/to/
wordlist.txt}} -t {{n_tasks}} {{host_ip}} {{ftp}}
```

- Guess MySQL credentials using a username and a passwords list, exiting when a username/password pair is found:

```
hydra -l {{username}} -P {{path/to/wordlist.txt}} -f
{{host_ip}} {{mysql}}
```

- Guess RDP credentials using a username and a passwords list, showing each attempt:

```
hydra -l {{username}} -P {{path/to/wordlist.txt}} -V {{rdp://
host_ip}}
```

- Guess IMAP credentials on a range of hosts using a list of colon-separated username/password pairs:

```
hydra -C {{path/to/username_password_pairs.txt}} {{imap://
[host_range_cidr]}}
```

- Guess POP3 credentials on a list of hosts using usernames and passwords lists, exiting when a username/password pair is found:

```
hydra -L {{path/to/usernames.txt}} -P {{path/to/
wordlist.txt}} -M {{path/to/hosts.txt}} -F {{pop3}}
```

hyperfine

A benchmarking tool.

More information: https://github.com/sharkdp/hyperfine/.

- Run a basic benchmark, performing at least 10 runs:

```
hyperfine '{{make}}'
```

- Run a comparative benchmark:

```
hyperfine '{{make target1}}' '{{make target2}}'
```

- Change minimum number of benchmarking runs:

```
hyperfine {{[-m|--min-runs]}} {{7}} '{{make}}'
```

- Perform benchmark with warmup:

```
hyperfine {{[-w|--warmup]}} {{5}} '{{make}}'
```

- Run a command before each benchmark run (to clear caches, etc.):

```
hyperfine {{[-p|--prepare]}} '{{make clean}}' '{{make}}'
```

- Run a benchmark where a single parameter changes for each run:

```
hyperfine {{[-p|--prepare]}} '{{make clean}}' {{[-P|--
parameter-scan]}} {{num_threads}} {{1}} {{10}} '{{make --jobs
{num_threads}}}'
```

i3-msg

Send messages to a running instance of i3 using IPC.

See https://i3wm.org/docs/userguide.html#list_of_commands for available commands.

More information: https://manned.org/i3-msg.

- Run an i3 command:

```
i3-msg {{command}}
```

- Print a list of workspaces in JSON:

```
i3-msg -t get_workspaces
```

- Print a layout tree of all open windows, containers, outputs, and workspaces in JSON:

```
i3-msg -t get_tree
```

i3-nagbar

Display an error/warning in a bar on top of a screen.

More information: https://manned.org/i3-nagbar.

- Display an error:

```
i3-nagbar {{[-m|--message]}} "{{error message}}"
```

- Display a warning:

```
i3-nagbar {{[-t|--type]}} warning {{[-m|--message]}}
"{{warning message}}"
```

- Use the specified font:

```
i3-nagbar {{[-f|--font]}} "{{pango:monospace bold 9}}" {{[-
m|--message]}} "{{error message}}"
```

- Create a button and run a command in a terminal on click:

```
i3-nagbar {{[-b|--button]}} "{{button text}}" {{command}}
{{[-m|--message]}} "{{error message}}"
```

- Create a button and run a command on click:

```
i3-nagbar {{[-B|--button-no-terminal]}} "{{button
text}}" {{command}} {{[-m|--message]}} "{{error message}}"
```

- Always open `i3-nagbar` on the primary monitor (default: focused monitor):

```
i3-nagbar {{[-pm|--primary --message]}} "{{error message}}"
```

i3

A tiling window manager for X11.

More information: https://i3wm.org/docs/userguide.html.

- Start `i3` (Note that a pre-existing window manager must not be open when this command is run):

  ```
  i3
  ```

- Open a new terminal window:

  ```
  <Super Enter>
  ```

- Move the focused window to a workspace:

  ```
  <Super Shift {{Number}}>
  ```

- Switch to a workspace:

  ```
  <Super {{Number}}>
  ```

- Split the next window horizontally:

  ```
  <Super h>
  ```

- Split the next window vertically:

  ```
  <Super v>
  ```

- Open an application launcher:

  ```
  <Super d>
  ```

i3lock

Simple screen locker built for the i3 window manager.

More information: https://manned.org/i3lock.

- Lock the screen showing a white background:

```
i3lock
```

- Lock the screen with a simple color background (rrggbb format):

```
i3lock {{[-c|--color]}} {{0000ff}}
```

- Lock the screen to a PNG background:

```
i3lock {{[-i|--image]}} {{path/to/file.png}}
```

- Lock the screen and disable the unlock indicator (removes feedback on keypress):

```
i3lock {{[-u|--no-unlock-indicator]}}
```

- Lock the screen and don't hide the mouse pointer:

```
i3lock {{[-p|--pointer]}} {{default}}
```

- Lock the screen to a PNG background tiled over all monitors:

```
i3lock {{[-i|--image]}} {{path/to/file.png}} {{[-t|--
tiling]}}
```

- Lock the screen and show the number of failed login attempts:

```
i3lock {{[-f|--show-failed-attempts]}}
```

ia

Tool to interact with **`archive.org`**.

More information: https://archive.org/services/docs/api/internetarchive/cli.html.

- Configure `ia` with API keys (some functions won't work without this step):

```
ia configure
```

- Upload one or more items to `archive.org`:

```
ia upload {{identifier}} {{path/to/file}} --
metadata="{{mediatype:data}}" --metadata="{{title:example}}"
```

- Download one or more items from `archive.org`:

```
ia download {{item}}
```

- Delete one or more items from `archive.org`:

```
ia delete {{identifier}} {{file}}
```

- Search on `archive.org`, returning results as JSON:

```
ia search '{{subject:"subject" collection:collection}}'
```

ibmcloud api

Set or view the IBM Cloud API endpoint.

More information: https://cloud.ibm.com/docs/cli?topic=cli-ibmcloud_cli#ibmcloud_api.

- View the current API endpoint:

```
ibmcloud api
```

- Set the API endpoint to `cloud.ibm.com`:

```
ibmcloud api cloud.ibm.com
```

- Set a private API endpoint:

```
ibmcloud api private.cloud.ibm.com
```

- Use a VPC connection for a private endpoint:

```
ibmcloud api private.cloud.ibm.com --vpc
```

- Bypass SSL validation of HTTP requests:

```
ibmcloud api https://cloud.ibm.com --skip-ssl-validation
```

- Remove the API endpoint setting:

```
ibmcloud api --unset
```

ibmcloud config

Modify or read out values in the IBM Cloud CLI configuration.

More information: https://cloud.ibm.com/docs/cli?topic=cli-ibmcloud_cli#ibmcloud_config.

- Set HTTP request timeout to 30 seconds:

```
ibmcloud config --http-timeout 30
```

- Enable trace output for HTTP requests:

```
ibmcloud config --trace true
```

- Trace HTTP requests to a specific file:

```
ibmcloud config --trace {{path/to/trace_file}}
```

- Disable color output:

```
ibmcloud config --color false
```

- Set the locale to a specific language:

```
ibmcloud config --locale {{zh_Hans}}
```

- Enable automatic SSO one-time passcode acceptance:

```
ibmcloud config --sso-otp auto
```

ibmcloud cr

Manage IBM Cloud Container Registry content and configuration.

More information: https://cloud.ibm.com/docs/cli?topic=cli-containerregcli.

- Set target region for IBM Cloud Container Registry:

```
ibmcloud cr region-set
```

- List images available:

```
ibmcloud cr {{[images|image-list]}}
```

- Inspect an image data:

```
ibmcloud cr image-inspect {{image}}
```

- Run vulnerability assessment on an image:

```
ibmcloud cr {{[va|vulnerability-assessment]}} {{image}}
```

- Log the local Docker or Podman client in to IBM Cloud Container Registry:

```
ibmcloud cr login
```

- List all actions available under this command:

```
ibmcloud cr help
```

ibmcloud iam

Manage identities and access to resources.

More information: https://cloud.ibm.com/docs/cli?topic=cli-ibmcloud_commands_iam.

- List service IDs in an account:

```
ibmcloud iam service-ids
```

- List all API keys for a service ID:

```
ibmcloud iam service-api-keys {{service_id}}
```

- Create an API key for a service ID with a description and without confirmation:

```
ibmcloud iam service-api-key-create {{api_key_name}} {{service_id}} {{[-d|--description]}} {{description}} {{[-f|--force]}}
```

- List all actions available under this command:

```
ibmcloud iam help
```

ibmcloud ks

Manage Kubernetes and OpenShift clusters in IBM Cloud.

More information: https://cloud.ibm.com/docs/cli?topic=cli-kubernetes-service-cli.

- View the details of a cluster:

```
ibmcloud ks cluster get {{[-c|--cluster]}} {{cluster_id}}
```

- View the rotation status of Cluster Authority certificates for a cluster:

```
ibmcloud ks cluster ca status {{[-c|--cluster]}}
{{cluster_id}}
```

- View worker pools of a cluster:

```
ibmcloud ks worker-pool ls {{[-c|--cluster]}} {{cluster_id}}
```

- Delete a worker node and replace it with a new one in the same worker pool:

```
ibmcloud ks worker replace {{[-c|--cluster]}} {{cluster_id}}
{{[-w|--worker]}} {{worker_id}}
```

- List all actions available under this command:

```
ibmcloud ks help
```

ibmcloud login

Log in to the IBM Cloud.

More information: https://cloud.ibm.com/docs/cli?topic=cli-ibmcloud_cli#ibmcloud_login.

- Log in by using an interactive prompt:

```
ibmcloud login
```

- Log in to a specific API endpoint (default is `cloud.ibm.com`):

```
ibmcloud login -a {{api_endpoint}}
```

- Log in by providing username, password and the targeted region as parameters:

```
ibmcloud login -u {{username}} -p {{password}} -r {{us-south}}
```

- Log in with an API key, passing it as an argument:

```
ibmcloud login --apikey {{api_key_string}}
```

- Log in with an API key, passing it as a file:

```
ibmcloud login --apikey @{{path/to/api_key_file}}
```

- Log in with a federated ID (single sign-on):

```
ibmcloud login --sso
```

ibmcloud logout

Log out of the IBM Cloud CLI.

More information: https://cloud.ibm.com/docs/cli?topic=cli-ibmcloud_cli#ibmcloud_logout.

- Log out of the current session:

```
ibmcloud logout
```

ibmcloud regions

List all available regions on IBM Cloud.

More information: https://cloud.ibm.com/docs/cli?topic=cli-ibmcloud_cli#ibmcloud_regions.

- View information for all regions:

```
ibmcloud regions
```

ibmcloud target

Set or view the target account, region, or resource group.

More information: https://cloud.ibm.com/docs/cli?topic=cli-ibmcloud_cli#ibmcloud_target.

- View the current target account and region:

```
ibmcloud target
```

- Set the target account:

```
ibmcloud target -c {{account_id}}
```

- Switch to a specific region:

```
ibmcloud target -r {{region_name}}
```

- Set the target resource group:

```
ibmcloud target -g {{resource_group_name}}
```

- Clear the targeted region:

```
ibmcloud target --unset-region
```

- Clear the targeted resource group:

```
ibmcloud target --unset-resource-group
```

ibmcloud

Manage IBM Cloud apps and services.

More information: https://cloud.ibm.com/docs/cli?topic=cli-ibmcloud_cli.

- Update `ibmcloud` to the latest version:

  ```
  ibmcloud update
  ```

- List all available IBM Cloud regions:

  ```
  ibmcloud regions
  ```

- Display help:

  ```
  ibmcloud help
  ```

- Display help for a subcommand:

  ```
  ibmcloud help {{subcommand}}
  ```

- Display version:

  ```
  ibmcloud version
  ```

ic

This command is an alias of **ibmcloud**.

- View documentation for the original command:

```
tldr ibmcloud
```

ical

View Hirji/Islamic and Gregorian calendars and convert their dates.

More information: https://manned.org/ical.

- Display the current month's calendar:

```
ical
```

- Convert a Gregorian date to a Hijri date:

```
ical --gregorian {{yyyymmdd}}
```

- Convert a Hirji date to a Gregorian date:

```
ical --hijri {{yyyymmdd}}
```

icontopbm

This command has been superseded by **`sunicontopnm`**.

More information: https://netpbm.sourceforge.net/doc/icontopbm.html.

- View documentation for the current command:

```
tldr sunicontopnm
```

iconv

Convert text from one encoding to another.

More information: https://manned.org/iconv.

- Convert file to a specific encoding, and print to `stdout`:

```
iconv {{[-f|--from-code]}} {{from_encoding}} {{[-t|--to-code]}} {{to_encoding}} {{input_file}}
```

- Convert file to the current locale's encoding, and output to a file:

```
iconv {{[-f|--from-code]}} {{from_encoding}} {{input_file}} > {{output_file}}
```

- List supported encodings:

```
iconv {{[-l|--list]}}
```

id

Display current user and group identity.

More information: https://www.gnu.org/software/coreutils/manual/html_node/id-invocation.html.

- Display current user's ID (UID), group ID (GID) and groups to which they belong:

```
id
```

- Display the current user identity:

```
id {{[-un|--user --name]}}
```

- Display the current user identity as a number:

```
id {{[-u|--user]}}
```

- Display the current primary group identity:

```
id {{[-gn|--group --name]}}
```

- Display the current primary group identity as a number:

```
id {{[-g|--group]}}
```

- Display an arbitrary user's ID (UID), group ID (GID) and groups to which they belong:

```
id {{username}}
```

id3tag

Read, write, and manipulate ID3v1 and ID3v2 tags of MP3 files.

More information: https://manned.org/id3tag.

- Set artist and song title tag of an MP3 file:

```
id3tag --artist {{artist}} --song {{song_title}} {{path/to/file.mp3}}
```

- Set album title of all MP3 files in the current directory:

```
id3tag --album {{album}} {{*.mp3}}
```

- Display help:

```
id3tag --help
```

idea

JetBrains Java and Kotlin IDE.

More information: https://www.jetbrains.com/help/idea/working-with-the-ide-features-from-command-line.html.

- Open the current directory in IntelliJ IDEA:

```
idea {{path/to/directory}}
```

- Open a specific file or directory in IntelliJ IDEA:

```
idea {{path/to/file_or_directory}}
```

- Open the diff viewer to compare up to 3 files:

```
idea diff {{path/to/file1 path/to/file2 path/to/optional_file3}}
```

- Open the merge dialog to perform a two-way file merge:

```
idea merge {{path/to/file1}} {{path/to/file2}} {{path/to/output}}
```

- Run code inspections on a project:

```
idea inspect {{path/to/project_directory}} {{path/to/inspection_profile}} {{path/to/output}}
```

identify

This command is an alias of **`magick identify`**.

- View documentation for the original command:

```
tldr magick identify
```

idevice_id

List attached iOS devices or print the device name of a given device.

More information: https://manned.org/idevice_id.

- List the UDIDs of all attached devices:

```
idevice_id --list
```

- List the UDIDs of all devices available via the network:

```
idevice_id --network
```

idevicebackup

Create or restore backups for iOS devices.

Note: This tool is outdated. Please see **`idevicebackup2`**.

More information: https://manned.org/idevicebackup.

- Create a backup of the device in the specified directory:

```
idevicebackup backup {{path/to/directory}}
```

- Restore a backup from the specified directory:

```
idevicebackup restore {{path/to/directory}}
```

idevicebackup2

Create or restore backups for devices running iOS 4 or later.

More information: https://manned.org/idevicebackup2.

- Create a backup of the device in the specified directory:

```
idevicebackup2 backup {{path/to/directory}}
```

- Restore a backup from the specified directory:

```
idevicebackup2 restore {{path/to/directory}}
```

- Enable encryption for backups:

```
idevicebackup2 encryption on {{password}}
```

- List the files in the last completed backup:

```
idevicebackup2 list
```

idevicecrashreport

Retrieve crash reports from an iOS device.

More information: https://manned.org/idevicecrashreport.

- Retrieve crash reports and move them to a specified directory:

```
idevicecrashreport {{path/to/directory}}
```

- Retrieve crash reports without removing them from the device:

```
idevicecrashreport --keep {{path/to/directory}}
```

- Extract crash reports into separate `.crash` files:

```
idevicecrashreport --extract {{path/to/directory}}
```

idevicedate

Display the current date or set it on an iOS device.

More information: https://manned.org/idevicedate.

- Display the current date and time:

```
idevicedate
```

- Set the date and time on the device to the system time:

```
idevicedate --sync
```

- Set the date and time to a specific timestamp:

```
idevicedate --set {{timestamp}}
```

idevicediagnostics

Interact with the diagnostics interface of an iOS device.

More information: https://manned.org/idevicediagnostics.

- Print diagnostics information:

```
idevicediagnostics diagnostics
```

- Print mobilegestalt key values:

```
idevicediagnostics mobilegestalt {{key1}} {{key2}}
```

- Shutdown, restart or sleep the device:

```
idevicediagnostics {{shutdown|restart|sleep}}
```

ideviceimagemounter

Mount disk images on an iOS device.

More information: https://manned.org/ideviceimagemounter.

- Mount a disk image on the connected device:

```
ideviceimagemounter {{path/to/image_file}} {{path/to/signature_file}}
```

- List currently mounted disk images:

```
ideviceimagemounter --list
```

ideviceinfo

Show information about the first connected iOS device.

More information: https://manned.org/ideviceinfo.

- Display detailed information about the connected device:

```
ideviceinfo
```

- Show information about a specific device by UDID:

```
ideviceinfo --udid {{device_udid}}
```

idevicename

Display the device name or set it to a new name.

More information: https://manned.org/idevicename.

- Display the current device name:

```
idevicename
```

- Set a new device name:

```
idevicename {{new_name}}
```

idevicepair

Manage host pairings with iOS devices.

More information: https://manned.org/idevicepair.

- Pair a device with the host:

```
idevicepair pair
```

- List devices paired with the host:

```
idevicepair list
```

idevicescreenshot

Get a screenshot from the connected iOS device.

More information: https://manned.org/idevicescreenshot.

- Save a screenshot with the default file name as a TIFF image:

```
idevicescreenshot
```

- Save a screenshot with a specific file name:

```
idevicescreenshot {{path/to/file.tiff}}
```

idevicesetlocation

Simulate a location on an iOS device.

More information: https://manned.org/idevicesetlocation.

- Set a specific latitude and longitude:

```
idevicesetlocation {{latitude}} {{longitude}}
```

- Reset the simulated location:

```
idevicesetlocation reset
```

idevicesyslog

Relay syslog messages from a connected iOS device.

More information: https://manned.org/idevicesyslog.

- Relay syslog messages from the connected device:

```
idevicesyslog
```

- Suppress kernel messages and print everything else:

```
idevicesyslog --no-kernel
```

idnits

Check internet-drafts for submission nits.

Looks for violations of Section 2.1 and 2.2 of the requirements listed on https://www.ietf.org/id-info/checklist.

More information: https://github.com/ietf-tools/idnits.

- Check a file for nits:

```
idnits {{path/to/file.txt}}
```

- Count nits without displaying them:

```
idnits --nitcount {{path/to/file.txt}}
```

- Show extra information about offending lines:

```
idnits --verbose {{path/to/file.txt}}
```

- Expect the specified year in the boilerplate instead of the current year:

```
idnits --year {{2021}} {{path/to/file.txt}}
```

- Assume the document is of the specified status:

```
idnits --doctype {{standard|informational|experimental|bcp|ps|ds}} {{path/to/file.txt}}
```

iex

IEx is the interactive shell for Elixir.

More information: https://manned.org/iex.

- Start an interactive session:

```
iex
```

- Start a session that remembers history:

```
iex --erl "-kernel shell_history enabled"
```

- Start and load Mix project files:

```
iex -S mix
```

if

Performs conditional processing in shell scripts.

See also: **`test`**, **`[`**.

More information: https://www.gnu.org/software/bash/manual/bash.html#Conditional-Constructs.

- Execute the specified commands if the condition command's exit status is zero:

```
if {{condition_command}}; then {{echo "Condition is true"}}; fi
```

- Execute the specified commands if the condition command's exit status is not zero:

```
if ! {{condition_command}}; then {{echo "Condition is true"}}; fi
```

- Execute the first specified commands if the condition command's exit status is zero otherwise execute the second specified commands:

```
if {{condition_command}}; then {{echo "Condition is true"}}; else {{echo "Condition is false"}}; fi
```

- Check whether a [f]ile exists:

```
if [[ -f {{path/to/file}} ]]; then {{echo "Condition is true"}}; fi
```

- Check whether a [d]irectory exists:

```
if [[ -d {{path/to/directory}} ]]; then {{echo "Condition is true"}}; fi
```

- Check whether a file or directory [e]xists:

```
if [[ -e {{path/to/file_or_directory}} ]]; then {{echo "Condition is true"}}; fi
```

- Check whether a variable is defined:

```
if [[ -n "${{variable}}" ]]; then {{echo "Condition is
true"}}; fi
```

- List all possible conditions (`test` is an alias to `[`; both are commonly used with `if`):

  ```
  man test
  ```

ifconfig

Network Interface Configurator.

More information: https://net-tools.sourceforge.io/man/ifconfig.8.html.

- View network settings of an interface:

```
ifconfig {{interface_name}}
```

- Display details of all interfaces, including disabled interfaces:

```
ifconfig -a
```

- Disable an interface:

```
ifconfig {{interface_name}} down
```

- Enable an interface:

```
ifconfig {{interface_name}} up
```

- Assign an IP address to an interface:

```
ifconfig {{interface_name}} {{ip_address}}
```

ifdata

Display information about a network interface.

More information: https://manned.org/ifdata.

- [p]rint the whole configuration of the specified interface:

```
ifdata -p {{eth0}}
```

- Indicate the [e]xistence of the specified interface via the exit code:

```
ifdata -e {{eth0}}
```

- [p]rint the IPv4 [a]dress and the [n]etmask of the specified interface:

```
ifdata -pa -pn {{eth0}}
```

- [p]rint the [N]etwork adress, the [b]roadcast adress, and the [m]TU of the specified interface:

```
ifdata -pN -pb -pm {{eth0}}
```

- Display help:

```
ifdata
```

ifne

Run a command depending on the emptyness of **`stdin`**.

More information: https://manned.org/ifne.

- Run the specified command if and only if `stdin` is not empty:

```
ifne {{command options ...}}
```

- Run the specified command if and only if `stdin` is empty, otherwise pass `stdin` to `stdout`:

```
ifne -n {{command options ...}}
```

IFS

IFS (Internal Field Separator) is a special environment variable that defines the delimiter used for word splitting in Unix shells.

The default value of IFS is a space, tab, and newline. The three characters serve as delimiters.

More information: https://www.gnu.org/software/bash/manual/bash.html#Word-Splitting.

- View the current IFS value:

```
echo "$IFS"
```

- Change the IFS value:

```
IFS="{{:}}"
```

- Reset IFS to default:

```
IFS=$' \t\n'
```

- Temporarily change the IFS value in a subshell:

```
(IFS="{{:}}"; echo "{{one:two:three}}")
```

ignite

A CLI for React Native boilerplates, plugins, generators, and more.

More information: https://infinite.red/ignite.

- Create a new React Native project:

  ```
  ignite new {{project_name}}
  ```

- Generate file from a plugin:

  ```
  ignite generate {{plugin_name}} {{path/to/file}}
  ```

- Add an Ignite plugin to the project:

  ```
  ignite add {{plugin_name}}
  ```

- Remove an Ignite plugin from the project:

  ```
  ignite remove {{plugin_name}}
  ```

ilbmtoppm

Convert an ILBM file to a PPM image.

More information: https://netpbm.sourceforge.net/doc/ilbmtoppm.html.

- Convert an ILBM file to a PPM image:

```
ilbmtoppm {{path/to/file.ilbm}} > {{path/to/file.ppm}}
```

- Use the specified color to "show through" where the image is transparent:

```
ilbmtoppm {{[-t|-transparent]}} {{color}} {{path/to/
file.ilbm}} > {{path/to/file.ppm}}
```

- Ignore the chunk with the specified chunk ID:

```
ilbmtoppm {{[-ig|-ignore]}} {{chunkID}} {{path/to/file.ilbm}}
> {{path/to/file.ppm}}
```

- Store the input's transparency information to the specified PBM file:

```
ilbmtoppm {{[-m|-maskfile]}} {{path/to/maskfile.pbm}} {{path/
to/file.ilbm}} > {{path/to/file.ppm}}
```

imapsync

Email IMAP tool for syncing, copying and migrating email mailboxes between two IMAP servers, one way, and without duplicates.

More information: https://imapsync.lamiral.info.

- Synchronize IMAP account between host1 and host2:

```
imapsync --host1 {{host1}} --user1 {{user1}} --password1
{{secret1}} --host2 {{host2}} --user2 {{user2}} --password2
{{secret2}}
```

img2pdf

Losslessly convert raster images to a PDF file.

Some supported image formats are: GIF, JPEG, JPEG2000, PNG, GIF and TIFF.

More information: https://gitlab.mister-muffin.de/josch/img2pdf.

- Convert one or more images to a single PDF, each image being on its own page:

```
img2pdf {{path/to/image1.ext path/to/image2.ext ...}} --output {{path/to/file.pdf}}
```

- Convert only the first frame of a multi-frame image to PDF:

```
img2pdf {{path/to/file.gif}} --first-frame-only --output {{path/to/file.pdf}}
```

- Auto orient the image, use a specific page size in landscape mode, and set a border of specific sizes horizontally and vertically:

```
img2pdf {{path/to/image.ext}} --auto-orient --pagesize {{A4^T}} --border {{2cm}}:{{5.1cm}} --output {{path/to/file.pdf}}
```

- Shrink only larger images to a rectangle of specified dimensions inside a page with a specific size:

```
img2pdf {{path/to/image.ext}} --pagesize {{30cm}}x{{20cm}} --imgsize {{10cm}}x{{15cm}} --fit {{shrink}} --output {{path/to/file.pdf}}
```

- Convert an image to PDF, and specify metadata for the resulting file:

```
img2pdf {{path/to/image.ext}} --title {{title}} --author {{author}} --creationdate {{1970-01-31}} --keywords {{keyword1 keyword2}} --subject {{subject}} --output {{path/to/file.pdf}}
```

img2sixel

Convert images to the Sixel format for displaying in a terminal.

More information: https://manned.org/img2sixel.

- Display an image in the terminal:

```
img2sixel {{path/to/image}}
```

- Resize the image to the specified width and height before displaying it:

```
img2sixel {{[-w|--width]}} {{number}} {{[-h|--height]}} {{number}} {{path/to/image}}
```

imgcat

Display images on the command-line.

Requires a compatible terminal such as iTerm2.

More information: https://github.com/danielgatis/imgcat.

- Display an image on the command-line:

```
imgcat {{path/to/file}}
```

imgtoppm

Convert various image file formats to the PPM (Portable Pixmap) format.

More information: https://netpbm.sourceforge.net/doc/imgtoppm.html.

- Convert an input image to PPM format:

```
imgtoppm {{path/to/input}} > {{path/to/output.ppm}}
```

- Display version:

```
imgtoppm {{[-v|-version]}}
```

immich-go

Immich-Go is an open-source tool designed to streamline uploading large photo collections to your self-hosted Immich server.

See also: **`immich-cli`**.

More information: https://github.com/simulot/immich-go.

- Upload a Google Photos takeout file to Immich server:

```
immich-go -server={{server_url}} -key={{server_key}} upload
{{path/to/takeout_file.zip}}
```

- Import photos captured on June 2019, while auto-generating albums:

```
immich-go -server={{server_url}} -key={{server_key}} upload -
create-albums -google-photos -date={{2019-06}} {{path/to/
takeout_file.zip}}
```

- Upload a takeout file using server and key from a configuration file:

```
immich-go -use-configuration={{~/.immich-go/immich-go.json}}
upload {{path/to/takeout_file.zip}}
```

- Examine Immich server content, remove less quality images, and preserve albums:

```
immich-go -server={{server_url}} -key={{server_key}}
duplicate -yes
```

- Delete all albums created with the pattern "YYYY-MM-DD":

```
immich-go -server={{server_url}} -key={{server_key}} tool
album delete {{\d{4}-\d{2}-\d{2}}}
```

immich

Immich command-line interface (CLI) that allows you to perform certain actions.

See also: **`immich-go`**.

More information: https://immich.app/docs/features/command-line-interface/.

- Authenticate to Immich server:

```
immich login {{server_url/api}} {{server_key}}
```

- Upload some image files:

```
immich upload {{file1.jpg file2.jpg ...}}
```

- Upload a directory including subdirectories:

```
immich upload --recursive {{path/to/directory}}
```

- Create an album based on a directory:

```
immich upload --album-name "{{My summer holiday}}" --
recursive {{path/to/directory}}
```

- Skip assets matching a glob pattern:

```
immich upload --ignore {{**/Raw/** **/*.tif}} --recursive
{{path/to/directory}}
```

- Include hidden files:

```
immich upload --include-hidden --recursive {{path/to/
directory}}
```

impacket-GetADUsers

This command is an alias of **`GetADUsers.py`**.

- View documentation for the original command:

  ```
  tldr GetADUsers.py
  ```

impacket-getArch

This command is an alias of **`getArch.py`**.

- View documentation for the original command:

  ```
  tldr getArch.py
  ```

impacket-GetNPUsers

This command is an alias of **`GetNPUsers.py`**.

- View documentation for the original command:

  ```
  tldr GetNPUsers.py
  ```

impacket-GetUserSPNs

This command is an alias of **`GetUserSPNs.py`**.

- View documentation for the original command:

  ```
  tldr GetUserSPNs.py
  ```

impacket-mqtt_check

This command is an alias of **`mqtt_check.py`**.

- View documentation for the original command:

```
tldr mqtt_check.py
```

impacket-mssqlclient

This command is an alias of **`mssqlclient.py`**.

- View documentation for the original command:

  ```
  tldr mssqlclient.py
  ```

impacket-ntfs-read

This command is an alias of **`ntfs-read.py`**.

- View documentation for the original command:

  ```
  tldr ntfs-read.py
  ```

impacket-ping

This command is an alias of **`ping.py`**.

- View documentation for the original command:

```
tldr ping.py
```

impacket-ping6

This command is an alias of **`ping6.py`**.

- View documentation for the original command:

```
tldr ping6.py
```

impacket-psexec

This command is an alias of **`psexec.py`**.

- View documentation for the original command:

```
tldr psexec.py
```

impacket-rpcdump

This command is an alias of **`rpcdump.py`**.

- View documentation for the original command:

```
tldr rpcdump.py
```

impacket-rpcmap

This command is an alias of **`rpcmap.py`**.

- View documentation for the original command:

```
tldr rpcmap.py
```

impacket-sambaPipe

This command is an alias of **`sambaPipe.py`**.

- View documentation for the original command:

  ```
  tldr sambaPipe.py
  ```

impacket-secretsdump

This command is an alias of **`secretsdump.py`**.

- View documentation for the original command:

  ```
  tldr secretsdump.py
  ```

impacket-smbclient

This command is an alias of **`smbclient.py`**.

- View documentation for the original command:

  ```
  tldr smbclient.py
  ```

impacket-smbserver

This command is an alias of **`smbserver.py`**.

- View documentation for the original command:

  ```
  tldr smbserver.py
  ```

impacket-sniff

This command is an alias of **`sniff.py`**.

- View documentation for the original command:

```
tldr sniff.py
```

impacket-sniffer

This command is an alias of **`sniffer.py`**.

- View documentation for the original command:

  ```
  tldr sniffer.py
  ```

import

This command is an alias of **`magick import`**.

- View documentation for the original command:

  ```
  tldr magick import
  ```

in-toto-record

Create a signed link metadata file to provide evidence for supply chain steps.

More information: https://in-toto.readthedocs.io/en/latest/command-line-tools/in-toto-record.html.

- Start the record (creates a preliminary link file):

```
in-toto-record start {{[-n|--step-name]}} {{path/to/
edit_file1 path/to/edit_file2 ...}} --signing-key {{path/to/
key_file}} {{[-m|--materials]}} {{.}}
```

- Stop the record (expects a preliminary link file):

```
in-toto-record stop {{[-n|--step-name]}} {{path/to/edit_file1
path/to/edit_file2 ...}} --signing-key {{path/to/key_file}}
{{[-p|--products]}} {{.}}
```

in-toto-run

Generating link metadata while carrying out a supply chain step.

More information: https://in-toto.readthedocs.io/en/latest/command-line-tools/in-toto-run.html.

- Tag a Git repo and signing the resulting link file:

```
in-toto-run {{[-n|--step-name]}} {{tag}} {{[-p|--products]}} {{.}} --signing-key {{key_file}} -- {{git tag v1.0}}
```

- Create a tarball, storing files as materials and the tarball as product:

```
in-toto-run {{[-n|--step-name]}} {{package}} {{[-m|--materials]}} {{project}} {{[-p|--products]}} {{project.tar.gz}} -- {{tar czf project.tar.gz project}}
```

- Generate signed attestations for review work:

```
in-toto-run {{[-n|--step-name]}} {{review}} --signing-key {{key_file}} {{[-m|--materials]}} {{document.pdf}} {{[-x|--no-command]}}
```

- Scan the image using Trivy and generate link file:

```
in-toto-run {{[-n|--step-name]}} {{scan}} --signing-key {{key_file}} {{[-p|--products]}} {{report.json}} -- /bin/sh -c "trivy --output report.json --format json {{path/to/image}}"
```

in-toto-sign

Sign in-toto link or layout metadata or verify their signatures.

More information: https://in-toto.readthedocs.io/en/latest/command-line-tools/in-toto-sign.html.

- Sign 'unsigned.layout' with two keys and write it to 'root.layout':

```
in-toto-sign {{[-f|--file]}} {{unsigned.layout}} {{[-k|--keep]}} {{priv_key1}} {{priv_key2}} {{[-o|--output]}} {{root.layout}}
```

- Replace signature in link file and write to default filename:

```
in-toto-sign {{[-f|--file]}} {{package.2f89b927.link}} {{[-k|--keep]}} {{priv_key}}
```

- Verify a layout signed with 3 keys:

```
in-toto-sign {{[-f|--file]}} {{root.layout}} {{[-k|--keep]}} {{pub_key0}} {{pub_key1}} {{pub_key2}} --verify
```

- Sign a layout with the default GPG key in default GPG keyring:

```
in-toto-sign {{[-f|--file]}} {{root.layout}} {{[-g|--gpg]}}
```

- Verify a layout with a GPG key identified by keyid '...439F3C2':

```
in-toto-sign {{[-f|--file]}} {{root.layout}} --verify {{[-g|--gpg]}} {{...439F3C2}}
```

in

This shell keyword is used with **`for`** to separate the iterator and items to iterate.

- View documentation for the `for` keyword:

```
tldr for
```

in2csv

Convert various tabular data formats to CSV.

Included in csvkit.

More information: https://csvkit.readthedocs.io/en/latest/scripts/in2csv.html.

- Convert an XLS file to CSV:

```
in2csv {{data.xls}}
```

- Convert a DBF file to a CSV file:

```
in2csv {{data.dbf}} > {{data.csv}}
```

- Convert a specific sheet from an XLSX file to CSV:

```
in2csv --sheet={{sheet_name}} {{data.xlsx}}
```

- Pipe a JSON file to in2csv:

```
cat {{data.json}} | in2csv {{[-f|--format]}} json >
{{data.csv}}
```

incus

Modern, secure and powerful system container and virtual machine manager.

More information: https://linuxcontainers.org/incus/docs/main.

- List all containers and virtual machines (both running and stopped):

```
incus list
```

- Create a container from an image, with a custom name:

```
incus create {{image}} {{container_name}}
```

- Start or stop an existing container:

```
incus {{start|stop}} {{container_name}}
```

- Open a shell inside an already running container:

```
incus shell {{container_name}}
```

- Remove a stopped container:

```
incus delete {{container_name}}
```

- Pull an image from an image repository (remote) to local:

```
incus copy {{remote}}:{{image}} local:{{custom_image_name}}
```

- List all available images in the official `images:` remote:

```
incus image list images:
```

- List all images already downloaded to the `local:` remote:

```
incus image list local:
```

indent

Change the appearance of a C/C++ program by inserting or deleting whitespace.

More information: https://www.gnu.org/software/indent/manual/indent/Option-Summary.html.

- Format C/C++ source according to the Linux style guide, automatically back up the original files, and replace with the indented versions:

```
indent {{[-linux|--linux-style]}} {{path/to/source1.c path/to/source2.c ...}}
```

- Format C/C++ source according to the GNU style, saving the indented version to a different file:

```
indent {{[-gnu|--gnu-style]}} {{path/to/source.c}} -o {{path/to/indented_source.c}}
```

- Format C/C++ source according to the style of Kernighan & Ritchie (K&R), no tabs, 3 spaces per indent, and wrap lines at 120 characters:

```
indent {{[-kr|--k-and-r-style]}} {{[-il|--indent-level]}}3 {{[-nut|--no-tabs]}} {{[-l|--line-length]}}120 {{path/to/source.c}} -o {{path/to/indented_source.c}}
```

infection

A mutation testing framework for PHP.

More information: https://infection.github.io.

- Analyze code using the configuration file (or create one if it does not exist):

  ```
  infection
  ```

- Use a specific number of threads:

  ```
  infection --threads {{number_of_threads}}
  ```

- Specify a minimum Mutation Score Indicator (MSI):

  ```
  infection --min-msi {{percentage}}
  ```

- Specify a minimum covered code MSI:

  ```
  infection --min-covered-msi {{percentage}}
  ```

- Use a specific test framework (defaults to PHPUnit):

  ```
  infection --test-framework {{phpunit|phpspec}}
  ```

- Only mutate lines of code that are covered by tests:

  ```
  infection --only-covered
  ```

- Display the mutation code that has been applied:

  ```
  infection --show-mutations
  ```

- Specify the log verbosity:

  ```
  infection --log-verbosity {{default|all|none}}
  ```

influx

InfluxDB v1 client.

More information: https://docs.influxdata.com/influxdb/v1/tools/influx-cli/use-influx-cli/.

- Connect to an InfluxDB running on localhost with no credentials:

```
influx
```

- Connect with a specific username (will prompt for a password):

```
influx -username {{username}} -password ""
```

- Connect to a specific host:

```
influx -host {{hostname}}
```

- Use a specific database:

```
influx -database {{database_name}}
```

- Execute a given command:

```
influx -execute "{{influxql_command}}"
```

- Return output in a specific format:

```
influx -execute "{{influxql_command}}" -format {{json|csv|column}}
```

info

Read documentation stored in the info format.

More information: https://manned.org/info.

- Start reading top-level directory menu:

```
info
```

- Start reading at given menu item node from top-level directory:

```
info {{menu_item}}
```

- Start reading at second menu item within first menu item manual:

```
info {{first_menu_item}} {{second_menu_item}}
```

initdb

Create a PostgreSQL database cluster on disk.

More information: https://www.postgresql.org/docs/current/app-initdb.html.

- Create a database cluster at `/usr/local/var/postgres`:

```
initdb {{[-D|--pgdata]}} /usr/local/var/postgres
```

inkmake

GNU Makefile-style SVG exporting using Inkscape's backend.

More information: https://github.com/wader/inkmake.

- Export an SVG file executing the specified Inkfile:

```
inkmake {{path/to/Inkfile}}
```

- Execute an Inkfile and show detailed information:

```
inkmake --verbose {{path/to/Inkfile}}
```

- Execute an Inkfile, specifying SVG input file(s) and an output file:

```
inkmake --svg {{path/to/file.svg}} --out {{path/to/
output_image}} {{path/to/Inkfile}}
```

- Use a custom Inkscape binary as the backend:

```
inkmake --inkscape {{/Applications/Inkscape.app/Contents/
Resources/bin/inkscape}} {{path/to/Inkfile}}
```

- Display help:

```
inkmake --help
```

inkscape

An SVG (Scalable Vector Graphics) editing program.

For Inkscape versions up to 0.92.x, use -e instead of -o.

More information: https://inkscape.org.

- Open an SVG file in the Inkscape GUI:

```
inkscape {{path/to/file.svg}}
```

- Export an SVG file into a bitmap with the default format (PNG) and the default resolution (96 DPI):

```
inkscape {{path/to/file.svg}} {{[-o|--export-filename]}} {{path/to/file.png}}
```

- Export an SVG file into a bitmap of 600x400 pixels (aspect ratio distortion may occur):

```
inkscape {{path/to/file.svg}} {{[-o|--export-filename]}} {{path/to/file.png}} {{[-w|--export-width]}} 600 {{[-h|--export-height]}} 400
```

- Export the drawing (bounding box of all objects) of an SVG file into a bitmap:

```
inkscape {{path/to/file.svg}} {{[-o|--export-filename]}} {{path/to/file.png}} {{[-D|--export-area-drawing]}}
```

- Export a single object, given its ID, into a bitmap:

```
inkscape {{path/to/file.svg}} {{[-i|--export-id]}} {{id}} {{[-o|--export-filename]}} {{object.png}}
```

- Export an SVG document to PDF, converting all texts to paths:

```
inkscape {{path/to/file.svg}} {{[-o|--export-filename]}} {{path/to/file.pdf}} {{[-T|--export-text-to-path]}}
```

- Duplicate the object with id="path123", rotate the duplicate 90 degrees, save the file, and quit Inkscape:

```
inkscape {{path/to/file.svg}} --select=path123 --verb="{{EditDuplicate;ObjectRotate90;FileSave;FileQuit}}"
```

inkview

Inkscape graphical SVG previewer.

Also functions as a slideshow viewer.

More information: https://wiki.inkscape.org/wiki/index.php/Inkview.

- Preview an SVG:

```
inkview {{path/to/file.svg}}
```

- Preview multiple SVGs (use `<ArrowKeys>` to navigate):

```
inkview {{path/to/file1.svg path/to/file2.svg ...}}
```

Install-NodeVersion

Install Node.js runtime versions for **`ps-nvm`**.

This command is part of **`ps-nvm`** and can only be run under PowerShell.

More information: https://github.com/aaronpowell/ps-nvm.

- Install a specific Node.js version:

```
Install-NodeVersion {{node_version}}
```

- Install multiple Node.js versions:

```
Install-NodeVersion {{node_version1 , node_version2 , ...}}
```

- Install latest available version of Node.js 20:

```
Install-NodeVersion ^20
```

- Install the x86 (x86 32-bit) / x64 (x86 64-bit) / arm64 (ARM 64-bit) version of Node.js:

```
Install-NodeVersion {{node_version}} -Architecture {{x86|x64|arm64}}
```

- Use a HTTP proxy to download Node.js:

```
Install-NodeVersion {{node-version}} -Proxy {{http://example.com}}
```

install-tl

TeX Live cross-platform installer.

More information: https://tug.org/texlive/.

- Start the text-based installer (default on Unix systems):

```
install-tl -no-gui
```

- Start the GUI installer (default on macOS and Windows, requires Tcl/Tk):

```
install-tl -gui
```

- Install TeX Live as defined in a specific profile file:

```
install-tl -profile {{path/to/texlive.profile}}
```

- Start the installer with the settings from a specific profile file:

```
install-tl -init-from-file {{path/to/texlive.profile}}
```

- Start the installer for installation on a portable device, like a USB stick:

```
install-tl -portable
```

- Display help:

```
install-tl -help
```

install

Copy files and set attributes.

Copy files (often executable) to a system location like **`/usr/local/bin`**, give them the appropriate permissions/ownership.

More information: https://www.gnu.org/software/coreutils/manual/html_node/install-invocation.html.

- Copy files to the destination:

```
install {{path/to/source_file1 path/to/source_file2 ...}} {{path/to/destination}}
```

- Copy files to the destination, setting their ownership:

```
install {{[-o|--owner]}} {{user}} {{path/to/source_file1 path/to/source_file2 ...}} {{path/to/destination}}
```

- Copy files to the destination, setting their group ownership:

```
install {{[-g|--group]}} {{user}} {{path/to/source_file1 path/to/source_file2 ...}} {{path/to/destination}}
```

- Copy files to the destination, setting their `mode`:

```
install {{[-m|--mode]}} {{+x}} {{path/to/source_file1 path/to/source_file2 ...}} {{path/to/destination}}
```

- Copy files and apply access/modification times of source to the destination:

```
install {{[-p|--preserve-timestamps]}} {{path/to/source_file1 path/to/source_file2 ...}} {{path/to/destination}}
```

- Copy files and create the directories at the destination if they don't exist:

```
install -D {{path/to/source_file1 path/to/source_file2 ...}} {{path/to/destination}}
```

interdiff

Show differences between two diff files.

More information: https://freshmeat.sourceforge.net/projects/patchutils.

- Compare diff files:

```
interdiff {{old_file}} {{new_file}}
```

- Compare diff files, ignoring whitespace:

```
interdiff {{[-w|--ignore-all-space]}} {{old_file}}
{{new_file}}
```

ionic

A framework to build hybrid mobile apps.

More information: https://ionicframework.com/docs/cli.

- Create a new project:

```
ionic start
```

- Start a local dev server for app dev/testing:

```
ionic serve
```

- Generate new app component, directive, page, pipe, provider or tabs:

```
ionic g {{page}}
```

- Run app on an Android/iOS device:

```
ionic cordova run {{android|ios}} --device
```

- Check the health of an Ionic app:

```
ionic doctor {{check}}
```

- Display versions of Ionic, Cordova, environment, etc.:

```
ionic info
```

ioping

Monitor I/O latency in real time.

More information: https://github.com/koct9i/ioping.

- Show disk I/O latency using the default values and the current directory:

```
ioping .
```

- Measure latency on /tmp using 10 requests of 1 megabyte each:

```
ioping {{[-c|-count]}} 10 {{[-s|-size]}} 1M /tmp
```

- Measure disk seek rate on `/dev/sdX`:

```
ioping {{[-R|-rapid]}} {{/dev/sdX}}
```

- Measure disk sequential speed on `/dev/sdX`:

```
ioping {{[-RL|-rapid -linear]}} {{/dev/sdX}}
```

iotop

Display a table of current I/O usage by processes or threads.

More information: https://manned.org/iotop.

- Start top-like I/O monitor:

```
sudo iotop
```

- Show only processes or threads actually doing I/O:

```
sudo iotop {{[-o|--only]}}
```

- Show I/O usage in non-interactive mode:

```
sudo iotop {{[-b|--batch]}}
```

- Show only I/O usage of processes (default is to show all threads):

```
sudo iotop {{[-P|--processes]}}
```

- Show I/O usage of given PID(s):

```
sudo iotop {{[-p|--pid]}} {{PID}}
```

- Show I/O usage of a given user:

```
sudo iotop {{[-u|--user]}} {{user}}
```

- Show accumulated I/O instead of bandwidth:

```
sudo iotop {{[-a|--accumulated]}}
```

ipaggcreate

Produce aggregate statistics of TCP/IP dumps.

More information: https://manned.org/ipaggcreate.

- Count the number of packets sent from each source address appearing in a PCAP file:

```
ipaggcreate --src {{path/to/file.pcap}}
```

- Group and count packets read from a network interface by IP packet length:

```
ipaggcreate --interface {{eth0}} --length
```

- Count the number of bytes sent between each address pair appearing in a PCAP file:

```
ipaggcreate --address-pairs --bytes {{path/to/file.pcap}}
```

ipaggmanip

Manipulate aggregate statistics produced by **`ipaggcreate`**.

More information: https://manned.org/ipaggmanip.

- Combine labels equal in their high-order bits:

```
ipaggmanip {{[-p|--prefix]}} {{16}} {{path/to/file}}
```

- Remove labels with a count smaller than a given number of bytes and output a random sample of such labels:

```
ipaggmanip --cut-smaller {{100}} --cull-labels {{5}} {{path/to/file}}
```

- Replace each label's count with 1 if it is non-zero:

```
ipaggmanip {{[-P|--posterize]}} {{path/to/file}}
```

ipcalc

Calculate IP information (subnet, broadcast, host range) from an IP address and netmask.

More information: https://manned.org/ipcalc.

- Display network info for an IP address:

```
ipcalc {{192.168.0.1}}
```

- Display network info using CIDR notation:

```
ipcalc {{192.168.0.1}}/{{24}}
```

- Display network info using a dotted decimal netmask:

```
ipcalc {{192.168.0.1}} {{255.255.255.0}}
```

- Suppress bitwise output:

```
ipcalc {{[-b|--nobinary]}} {{192.168.0.1}}
```

- Split a network into specified sized blocks:

```
ipcalc {{[-s|--split]}} {{size1 size2 size3 ...}}
{{192.168.0.1}}
```

- Display version:

```
ipcalc {{[-v|--version]}}
```

ipcs

Show information about the usage of XSI IPC facilities: shared memory segments, message queues, and semaphore arrays.

More information: https://manned.org/ipcs.1p.

- Show information about [a]ll the IPC:

```
ipcs -a
```

- Show information about active shared [m]emory segments, message [q]ueues or [s]empahore sets:

```
ipcs {{-m|-q|-s}}
```

- Show information on maximum allowable size in [b]ytes:

```
ipcs -b
```

- Show [c]reator's user name and group name for all IPC facilities:

```
ipcs -c
```

- Show the [p]ID of the last operators for all IPC facilities:

```
ipcs -p
```

- Show access [t]imes for all IPC facilities:

```
ipcs -t
```

- Show [o]utstanding usage for active message queues, and shared memory segments:

```
ipcs -o
```

iperf

Measure network bandwidth between computers.

More information: https://iperf.fr.

- Run on server:

```
iperf {{[-s|--server]}}
```

- Run on server using UDP mode and set server port to listen on 5001:

```
iperf {{[-u|--udp]}} {{[-s|--server]}} {{[-p|--port]}}
{{5001}}
```

- Run on client:

```
iperf {{[-c|--client]}} {{server_address}}
```

- Run on client every 2 seconds:

```
iperf {{[-c|--client]}} {{server_address}} {{[-i|--
interval]}} {{2}}
```

- Run on client with 5 parallel threads:

```
iperf {{[-c|--client]}} {{server_address}} {{[-P|--
parallel]}} {{5}}
```

- Run on client using UDP mode:

```
iperf {{[-u|--udp]}} {{[-c|--client]}} {{server_address}}
{{[-p|--port]}} {{5001}}
```

iperf3

Traffic generator for testing network bandwidth.

More information: https://iperf.fr.

- Run iperf3 as a server:

```
iperf3 {{[-s|--server]}}
```

- Run an iperf3 server on a specific port:

```
iperf3 {{[-s|--server]}} {{[-p|--port]}} {{port}}
```

- Start bandwidth test:

```
iperf3 {{[-c|--client]}} {{server}}
```

- Run iperf3 in multiple parallel streams:

```
iperf3 {{[-c|--client]}} {{server}} {{[-P|--parallel]}}
{{streams}}
```

- Reverse direction of the test. Server sends data to the client:

```
iperf3 {{[-c|--client]}} {{server}} {{[-R|--reverse]}}
```

ipfs

Inter Planetary File System.

A peer-to-peer hypermedia protocol. Aims to make the web more open.

More information: https://ipfs.io.

- Add a file from local to the filesystem, pin it and print the relative hash:

```
ipfs add {{path/to/file}}
```

- Add a directory and its files recursively from local to the filesystem and print the relative hash:

```
ipfs add -r {{path/to/directory}}
```

- Save a remote file and give it a name but not pin it:

```
ipfs get {{hash}} -o {{path/to/file}}
```

- Pin a remote file locally:

```
ipfs pin add {{hash}}
```

- Display pinned files:

```
ipfs pin ls
```

- Unpin a file from the local storage:

```
ipfs pin rm {{hash}}
```

- Remove unpinned files from local storage:

```
ipfs repo gc
```

ipinfo

Official CLI for the IPinfo.io IP geolocation and network intelligence API.

Note: Some commands will require a token from IPinfo.io.

More information: https://github.com/ipinfo/cli#quick-start.

- Display details for your current IP address:

```
ipinfo myip
```

- Display details for a specific IP address:

```
ipinfo {{ip_address}}
```

- Display details for multiple IP addresses in bulk from a file:

```
ipinfo bulk {{path/to/ips.txt}}
```

- Display details for a CIDR or IP range:

```
ipinfo bulk {{cidr_range}}
```

- Display only specific fields from IP lookup results:

```
ipinfo {{ip_address}} {{[-f|--field]}}
{{hostname,country,org}}
```

- Summarize details for a group of IP addresses:

```
ipinfo summarize {{path/to/ips.txt}}
```

- Extract IP addresses from text and highlight them:

```
ipinfo grepip {{path/to/file.txt}}
```

- Display help:

```
ipinfo {{[-h|--help]}}
```

ipmitool

Interface with the Intelligent Platform Management Interface (IPMI).

More information: https://manned.org/ipmitool.

- Open IPMI shell on the local hardware:

```
sudo ipmitool shell
```

- Open IPMI shell on a remote host:

```
ipmitool -H {{ip_address}} -U {{user_name}} shell
```

ippeveprinter

A simple IPP Everywhere printer server.

See also: **`ippeveps`**, **`ippevepcl`**.

More information: https://openprinting.github.io/cups/doc/man-ippeveprinter.html.

- Run the server with a specific service name:

```
ippeveprinter "{{service_name}}"
```

- Load printer attributes from a PPD file:

```
ippeveprinter -P {{path/to/file.ppd}} "{{service_name}}"
```

- Run the `file` command whenever a job is sent to the server:

```
ippeveprinter -c {{/usr/bin/file}} "{{service_name}}"
```

- Specify the directory that will hold the print files (by default, a directory under the user's temporary directory):

```
ippeveprinter -d {{spool_directory}} "{{service_name}}"
```

- Keep the print documents in the spool directory rather than deleting them:

```
ippeveprinter -k "{{service_name}}"
```

- Specify the printer speed in pages/minute unit (10 by default):

```
ippeveprinter -s {{speed}} "{{service_name}}"
```

ippeveps

Print to Adobe PostScript printers.

Supports PDF, PostScript, JPEG, PWG Raster or Apple Raster files.

See also: **`ippevepcl`**, **`ippeveprinter`**.

More information: https://openprinting.github.io/cups/doc/man-ippevepcl.html.

- Print a file to `stdout` (status and progress messages are sent to `stderr`):

```
ippeveps {{path/to/file}}
```

- Print a file from `stdin` to `stdout`:

```
{{wget --output-document - https://example.com/file}} |
ippeveps
```

ippfind

Find services registered with a DNS server or available through local devices.

See also: **`ipptool`**, **`ippeveprinter`**.

More information: https://openprinting.github.io/cups/doc/man-ippfind.html.

- List IPP printers registered on the network with their status:

```
ippfind {{[-l|--ls]}}
```

- Send a specific PostScript document to every PostScript printer on the network:

```
ippfind --txt-pdl application/postscript {{[-x|--exec]}}
ipptool -f {{path/to/document.ps}} '{}' print-job.test \;
```

- Send a PostScript test document to every PostScript printer on the network:

```
ippfind --txt-pdl application/postscript {{[-x|--exec]}}
ipptool -f onepage-letter.ps '{}' print-job.test \;
```

- Send a PostScript test document to every PostScript printer on the network, whose name matches a `regex`:

```
ippfind --txt-pdl application/postscript {{[-h|--host]}}
{{regex}} {{[-x|--exec]}} ipptool -f onepage-letter.ps '{}'
print-job.test \;
```

ipptool

Issue IPP requests and receive printer's/server's responses.

See also: **`ippfind`**, **`ippeveprinter`**.

More information: https://openprinting.github.io/cups/doc/man-ipptool.html.

- Get all attributes and their values supported by a printer:

```
ipptool ipp://{{printer_uri}} get-completed-jobs.test
```

- Get the list of completed jobs of a printer:

```
ipptool ipp://{{printer_uri}} get-completed-jobs.test
```

- Send an email notification when a printer changes:

```
ipptool -d recipient=mailto:{{email}} ipp://{{printer_uri}}
create-printer-subscription.test
```

iproxy

A proxy that binds local TCP ports to be forwarded to the specified ports on a usbmux device.

More information: https://manned.org/iproxy.

- Bind a local TCP port and forward it to a port on the connected USB device:

```
iproxy {{local_port}}:{{device_port}}
```

- Bind multiple local TCP ports and forward them to the respective ports on the connected USB device:

```
iproxy {{local_port1}}:{{device_port1}} {{local_port2}}:
{{device_port2}}
```

- Bind a local port and forward it to a specific device by UDID:

```
iproxy --udid {{device_udid}} {{local_port}}:{{device_port}}
```

- Bind a local port and forward it to a network-connected device with Wi-Fi sync enabled:

```
iproxy --network {{local_port}}:{{device_port}}
```

ipscan

A fast network scanner designed to be simple to use.

Also known as Angry IP Scanner.

More information: https://angryip.org/.

- Scan a specific IP address:

```
ipscan {{192.168.0.1}}
```

- Scan a range of IP addresses:

```
ipscan {{192.168.0.1-254}}
```

- Scan a range of IP addresses and save the results to a file:

```
ipscan {{192.168.0.1-254}} -o {{path/to/output.txt}}
```

- Scan IPs with a specific set of ports:

```
ipscan {{192.168.0.1-254}} -p {{80,443,22}}
```

- Scan with a delay between requests to avoid network congestion:

```
ipscan {{192.168.0.1-254}} -d {{200}}
```

- Display help:

```
ipscan --help
```

ipsumdump

Summarise TCP/IP dumps into a human and machine readable ASCII format.

More information: https://manned.org/ipsumdump.

- Print the source and destination IP addresses of all packets in a PCAP file:

```
ipsumdump --src --dst {{path/to/file.pcap}}
```

- Print the timestamps, source address, source port, destination address, destination port and protocol of all packets read from a given network interface:

```
ipsumdump --interface {{eth0}} -tsSdDp
```

- Print the anonymised source address, anonymised destination address, and IP packet length of all packets in a PCAP file:

```
ipsumdump --src --dst --length --anonymize {{path/to/
file.pcap}}
```

ipython

A Python shell with automatic history, dynamic object introspection, easier configuration, command completion, access to the system shell and more.

More information: https://ipython.readthedocs.io.

- Start a REPL (interactive shell):

```
ipython
```

- Enter an interactive IPython session after running a Python script:

```
ipython -i {{script.py}}
```

- Create default IPython profile:

```
ipython profile create
```

- Print the path to the directory for the default IPython profile:

```
ipython locate profile
```

- Clear the IPython history database, deleting all entries:

```
ipython history clear
```

irb

Interactive Ruby shell.

Evaluate Ruby code read from **`stdin`**.

More information: https://ruby.github.io/irb/.

- Start the interactive shell:

```
irb
```

irssi

Text based IRC client.

More information: https://irssi.org.

- Open Irssi and connect to a server with a nickname:

```
irssi {{[-n|--nick]}} {{nickname}} {{[-c|--connect]}} {{irc.example.com}}
```

- Open Irssi and connect with a specific server on a given port:

```
irssi {{[-c|--connect]}} {{irc.example.com}} {{[-p|--port]}} {{port}}
```

- Display help:

```
irssi {{[-?|--help]}}
```

- Join a channel:

```
/join {{#channelname}}
```

- Change active window (starts at 1):

```
/win {{window_number}}
```

- Exit the application cleanly and quitting any server(s):

```
/quit
```

is-up

Check whether a website is up or down.

More information: https://github.com/sindresorhus/is-up-cli.

- Check the status of the specified website:

```
is-up {{example.com}}
```

isisdl

A downloading utility for ISIS of TU-Berlin. Download all your files and videos from ISIS.

More information: https://github.com/Emily3403/isisdl.

- Start the synchronization process:

```
isisdl
```

- Limit the download rate to 20 MiB/s and download with 5 threads:

```
isisdl --download-rate {{20}} --max-num-threads {{5}}
```

- Run the initialization configuration wizard:

```
isisdl --init
```

- Run the additional configuration wizard:

```
isisdl --config
```

- Initiate a full synchronization of the database and compute the checksum of every file:

```
isisdl --sync
```

- Start ffmpeg to compress downloaded videos:

```
isisdl --compress
```

ispell

Interactive spell checking.

More information: https://www.cs.hmc.edu/~geoff/ispell-man.html.

- Start an interactive session:

```
ispell
```

- Check for typos in the specified file and interactively apply suggestions:

```
ispell {{path/to/file}}
```

- Display version:

```
ispell -v
```

isutf8

Check whether text files contain valid UTF-8.

More information: https://manned.org/isutf8.

- Check whether the specified files contain valid UTF-8:

```
isutf8 {{path/to/file1 path/to/file2 ...}}
```

- Print errors using multiple lines:

```
isutf8 {{[-v|--verbose]}} {{path/to/file1 path/to/file2 ...}}
```

- Do not print anything to `stdout`, indicate the result merely with the exit code:

```
isutf8 {{[-q|--quiet]}} {{path/to/file1 path/to/file2 ...}}
```

- Only print the names of the files containing invalid UTF-8:

```
isutf8 {{[-l|--list]}} {{path/to/file1 path/to/file2 ...}}
```

- Same as `--list` but inverted, i.e., only print the names of the files containing valid UTF-8:

```
isutf8 {{[-i|--invert]}} {{path/to/file1 path/to/file2 ...}}
```

iverilog

Preprocesses and compiles Verilog HDL (IEEE-1364) code into executable programs for simulation.

More information: https://github.com/steveicarus/iverilog.

- Compile a source file into an executable:

```
iverilog {{path/to/source.v}} -o {{path/to/executable}}
```

- Compile a source file into an executable while displaying all warnings:

```
iverilog {{path/to/source.v}} -Wall -o {{path/to/executable}}
```

- Compile and run explicitly using the VVP runtime:

```
iverilog -o {{path/to/executable}} -tvvp {{path/to/source.v}}
```

- Compile using Verilog library files from a different path:

```
iverilog {{path/to/source.v}} -o {{path/to/executable}} -
I{{path/to/library_directory}}
```

- Preprocess Verilog code without compiling:

```
iverilog -E {{path/to/source.v}}
```

ivpn

Interface for the IVPN client.

See also: **`fastd`**, **`mozillavpn`**, **`mullvad`**, **`warp-cli`**.

More information: https://www.ivpn.net.

- Print the current state of IVPN, including the connection and firewall status:

```
ivpn status
```

- List available servers to connect to:

```
ivpn servers
```

j

This command is an alias of **autojump**.

- View documentation for the original command:

```
tldr autojump
```

jadx

Dex to Java decompiler.

Produces Java source code from Android Dex and APK files.

More information: https://github.com/skylot/jadx#usage.

- Decompile a Dex file into a directory:

```
jadx {{path/to/file}}
```

- Decompile a Dex file into a specific directory:

```
jadx {{[-d|--output-dir]}} {{path/to/directory}} {{path/to/file}}
```

jar

Java applications/libraries packager.

More information: https://docs.oracle.com/javase/tutorial/deployment/jar/basicsindex.html.

- Recursively archive all files in the current directory into a .jar file:

```
jar cf {{file.jar}} *
```

- Unzip .jar/.war file to the current directory:

```
jar -xvf {{file.jar}}
```

- List a .jar/.war file content:

```
jar tf {{path/to/file.jar}}
```

- List a .jar/.war file content with verbose output:

```
jar tvf {{path/to/file.jar}}
```

jarsigner

Sign and verify Java archive (JAR) files.

More information: https://docs.oracle.com/en/java/javase/20/docs/specs/man/jarsigner.html.

- Sign a JAR file:

```
jarsigner {{path/to/file.jar}} {{keystore_alias}}
```

- Sign a JAR file with a specific algorithm:

```
jarsigner -sigalg {{algorithm}} {{path/to/file.jar}} {{keystore_alias}}
```

- Verify the signature of a JAR file:

```
jarsigner -verify {{path/to/file.jar}}
```

java

Java application launcher.

More information: https://docs.oracle.com/en/java/javase/20/docs/specs/man/java.html.

- Execute a Java `.class` file that contains a main method by using just the class name:

```
java {{classname}}
```

- Execute a Java program and use additional third-party or user-defined classes:

```
java -classpath {{path/to/classes1}}:{{path/to/classes2}}:.
{{classname}}
```

- Execute a `.jar` program:

```
java -jar {{filename.jar}}
```

- Execute a `.jar` program with debug waiting to connect on port 5005:

```
java -
agentlib:jdwp=transport=dt_socket,server=y,suspend=y,address=*:
5005 -jar {{filename.jar}}
```

- Display JDK, JRE and HotSpot versions:

```
java -version
```

- Display help:

```
java -help
```

javac

Java application compiler.

More information: https://docs.oracle.com/en/java/javase/20/docs/specs/man/javac.html.

- Compile a `.java` file:

```
javac {{path/to/file.java}}
```

- Compile several `.java` files:

```
javac {{path/to/file1.java path/to/file2.java ...}}
```

- Compile all `.java` files in current directory:

```
javac {{*.java}}
```

- Compile a `.java` file and place the resulting class file in a specific directory:

```
javac -d {{path/to/directory}} {{path/to/file.java}}
```

javadoc

Generate Java API documentation in HTML format from source code.

More information: https://docs.oracle.com/en/java/javase/20/docs/specs/man/javadoc.html.

- Generate documentation for Java source code and save the result in a directory:

```
javadoc -d {{path/to/directory}}/ {{path/to/java_source_code}}
```

- Generate documentation with a specific encoding:

```
javadoc -docencoding {{UTF-8}} {{path/to/java_source_code}}
```

- Generate documentation excluding some packages:

```
javadoc -exclude {{package_list}} {{path/to/java_source_code}}
```

javap

Disassemble class files and list them.

More information: https://docs.oracle.com/en/java/javase/20/docs/specs/man/javap.html.

- Disassemble and list one or more `.class` files:

```
javap {{path/to/file1.class path/to/file2.class ...}}
```

- Disassemble and list a built-in class file:

```
javap java.{{package}}.{{class}}
```

- Display help:

```
javap -help
```

- Display version:

```
javap -version
```

jbang

Easily create, edit and run self-contained source-only Java programs.

See also: **`java`**.

More information: https://www.jbang.dev/documentation/jbang/latest/cli/jbang.html.

- Initialize a simple Java class:

```
jbang init {{path/to/file.java}}
```

- Initialize a Java class (useful for scripting):

```
jbang init --template={{cli}} {{path/to/file.java}}
```

- Use `jshell` to explore and use a script and any dependencies in a REPL editor:

```
jbang run --interactive
```

- Setup a temporary project to edit a script in an IDE:

```
jbang edit --open={{codium|code|eclipse|idea|netbeans|gitpod}} {{path/to/script.java}}
```

- Run a Java code snippet (Java 9 and later):

```
{{echo 'Files.list(Paths.get("/etc")).forEach(System.out::println);'}} | jbang -
```

- Run command-line application:

```
jbang {{path/to/file.java}} {{command}} {{arg1 arg2 ...}}
```

- Install a script on the user's `$PATH`:

```
jbang app install --name {{command_name}} {{path/to/script.java}}
```

- Install a specific version of JDK to be used with `jbang`:

```
jbang jdk install {{version}}
```

jc

Convert the output of multiple commands to JSON.

More information: https://github.com/kellyjonbrazil/jc.

- Convert command output to JSON via pipe:

```
{{ifconfig}} | jc {{--ifconfig}}
```

- Convert command output to JSON via magic syntax:

```
jc {{ifconfig}}
```

- Output pretty JSON via pipe:

```
{{ifconfig}} | jc {{--ifconfig}} {{[-p|--pretty]}}
```

- Output pretty JSON via magic syntax:

```
jc {{[-p|--pretty]}} {{ifconfig}}
```

jc

jc can refer to multiple commands with the same name.

- View documentation for the JSON serializer:

  ```
  tldr jc.json
  ```

- View documentation for the autojump alias:

  ```
  tldr autojump
  ```

jcal

Display calendar information in the Jalali format, with the current day highlighted.

More information: https://www.nongnu.org/jcal/jcal.html.

- Display a calendar for the current month:

```
jcal
```

- Display the previous, current, and next months:

```
jcal -3
```

- Display a calendar for a specific year (4 digits):

```
jcal {{year}}
```

- Display a calendar for a specific month and year:

```
jcal {{year}} {{month}}
```

jco

This command is an alias of **autojump**.

- View documentation for the original command:

```
tldr autojump
```

jdeps

Java class dependency analyzer.

More information: https://docs.oracle.com/en/java/javase/20/docs/specs/man/jdeps.html.

- Analyze the dependencies of a `.jar` or `.class` file:

```
jdeps {{path/to/file.class}}
```

- Print a summary of all dependencies of a specific `.jar` file:

```
jdeps {{path/to/file.jar}} -summary
```

- Print all class-level dependencies of a `.jar` file:

```
jdeps {{path/to/file.jar}} -verbose
```

- Output the results of the analysis in a DOT file into a specific directory:

```
jdeps {{path/to/file.jar}} -dotoutput {{path/to/directory}}
```

- Display help:

```
jdeps --help
```

jdupes

A powerful duplicate file finder and an enhanced fork of **fdupes**.

More information: https://codeberg.org/jbruchon/jdupes#usage.

- Search a single directory:

```
jdupes {{path/to/directory}}
```

- Search multiple directories:

```
jdupes {{directory1 directory2 ...}}
```

- Search all directories recursively:

```
jdupes {{[-r|--recurse]}} {{path/to/directory}}
```

- Search directory recursively and let user choose files to preserve:

```
jdupes {{[-d|--delete]}} {{[-r|--recurse]}} {{path/to/
directory}}
```

- Search multiple directories and follow subdirectores under directory2, not directory1:

```
jdupes {{directory1}} {{[-R|--recurse:]}} {{directory2}}
```

- Search multiple directories and keep the directory order in result:

```
jdupes {{[-O|--param-order]}} {{directory1 directory2
directory3 ...}}
```

jekyll

A simple, blog-aware, static site generator.

More information: https://jekyllrb.com/docs/usage/.

- Generate a development server that will run at http://localhost:4000/:

```
jekyll serve
```

- Enable incremental regeneration:

```
jekyll serve --incremental
```

- Enable verbose output:

```
jekyll serve --verbose
```

- Generate the current directory into `./_site`:

```
jekyll build
```

- Clean the site (removes site output and `cache` directory) without building:

```
jekyll clean
```

jello

A JSON processor using Python syntax.

More information: https://github.com/kellyjonbrazil/jello.

- Pretty-print JSON or JSON-Lines data from `stdin` to `stdout`:

```
cat {{file.json}} | jello
```

- Output a schema of JSON or JSON Lines data from `stdin` to `stdout` (useful for grep):

```
cat {{file.json}} | jello -s
```

- Output all elements from arrays (or all the values from objects) in JSON or JSON-Lines data from `stdin` to `stdout`:

```
cat {{file.json}} | jello -l
```

- Output the first element in JSON or JSON-Lines data from `stdin` to `stdout`:

```
cat {{file.json}} | jello _[0]
```

- Output the value of a given key of each element in JSON or JSON-Lines data from `stdin` to `stdout`:

```
cat {{file.json}} | jello '[i.{{key_name}} for i in _]'
```

- Output the value of multiple keys as a new JSON object (assuming the input JSON has the keys `key_name1` and `key_name2`):

```
cat {{file.json}} | jello '{{{"key1": _.key_name1, "key2":
_.key_name2, ...}}}'
```

- Output the value of a given key to a string (and disable JSON output):

```
cat {{file.json}} | jello -r '"{{some text}}: " + _.
{{key_name}}'
```

jenkins-cli

An open-source automation server that facilitates the automation of software development lifecycle.

More information: https://www.jenkins.io/doc/.

- Connect to jenkins CLI:

```
java -jar jenkins-cli.jar -s {{jenkins_server_url}} -auth {{username}}:{{api_token}}
```

- Restart jenkins:

```
java -jar jenkins-cli.jar -s {{jenkins_server_url}} restart
```

- Shutdown jenkins:

```
java -jar jenkins-cli.jar -s {{jenkins_server_url}} shutdown
```

- Display help:

```
java -jar jenkins-cli.jar -s {{jenkins_server_url}} help
```

- Display version:

```
java -jar jenkins-cli.jar -s {{jenkins_server_url}} version
```

jenv

Manage the "JAVA_HOME" environment variable.

More information: https://github.com/jenv/jenv/.

- Add a Java version to jEnv:

```
jenv add {{path/to/jdk_home}}
```

- Display the current JDK version used:

```
jenv version
```

- Display all managed JDKs:

```
jenv versions
```

- Set the global JDK version:

```
jenv global {{java_version}}
```

- Set the JDK version for the current shell session:

```
jenv shell {{java_version}}
```

- Enable a jEnv plugin:

```
jenv enable-plugin {{plugin_name}}
```

jest

A zero-configuration JavaScript testing platform.

More information: https://jestjs.io.

- Run all available tests:

  ```
  jest
  ```

- Run the test suites from the given files:

  ```
  jest {{path/to/file1 path/to/file2 ...}}
  ```

- Run the test suites from files within the current and subdirectories, whose paths match the given `regex`:

  ```
  jest {{regex1}} {{regex2}}
  ```

- Run the tests whose names match the given `regex`:

  ```
  jest --testNamePattern {{regex}}
  ```

- Run test suites related to a given source file:

  ```
  jest --findRelatedTests {{path/to/source_file.js}}
  ```

- Run test suites related to all uncommitted files:

  ```
  jest --onlyChanged
  ```

- Watch files for changes and automatically re-run related tests:

  ```
  jest --watch
  ```

- Display help:

  ```
  jest --help
  ```

jetifier

Jetifier AndroidX transition tool in npm format, with a react-native compatible style.

More information: https://github.com/mikehardy/jetifier.

- Migrate project dependencies to the AndroidX format:

  ```
  jetifier
  ```

- Migrate project dependencies from the AndroidX format:

  ```
  jetifier reverse
  ```

jf

Interact with JFrog products like Artifactory, Xray, Distribution, Pipelines and Mission Control.

More information: https://jfrog.com/help/r/jfrog-cli/usage.

- Add a new configuration:

```
jf config add
```

- Show the current configuration:

```
jf config show
```

- Search for artifacts within the given repository and directory:

```
jf rt search --recursive {{repostiory_name}}/{{path}}/
```

jfrog

This command is an alias of **`jf`**.

- View documentation for the original command:

```
tldr jf
```

jhat

Java heap analysis tool.

More information: https://docs.oracle.com/javase/8/docs/technotes/tools/unix/jhat.html.

- Analyze a heap dump (from `jmap`), view via HTTP on port 7000:

```
jhat {{dump_file.bin}}
```

- Analyze a heap dump, specifying an alternate port for the HTTP server:

```
jhat {{[-p|-port]}} {{port}} {{dump_file.bin}}
```

- Analyze a dump letting `jhat` use up to 8 GB RAM (2-4x dump size recommended):

```
jhat -J-mx8G {{dump_file.bin}}
```

jhipster

Web application generator using either monolithic or microservices architecture.

More information: https://www.jhipster.tech/creating-an-app/#command-line-options.

- Generate a simple full-stack project (monolithic or microservices):

```
jhipster
```

- Generate a simple frontend project:

```
jhipster --skip-server
```

- Generate a simple backend project:

```
jhipster --skip-client
```

- Apply latest JHipster updates to the project:

```
jhipster upgrade
```

- Add a new entity to a generated project:

```
jhipster entity {{entity_name}}
```

- Import a JDL file to configure your application (see: https://start.jhipster.tech/jdl-studio/):

```
jhipster import-jdl {{file1.jh file2.jh ...}}
```

- Generate a CI/CD pipeline for your application:

```
jhipster ci-cd
```

- Generate a Kubernetes configuration for your application:

```
jhipster kubernetes
```

jhsdb

Attach to a Java process or launch a postmortem debugger to analyze the core dump from a crashed Java Virtual Machine.

More information: https://manned.org/jhsdb.

- Print stack and locks information of a Java process:

```
jhsdb jstack --pid {{pid}}
```

- Open a core dump in interactive debug mode:

```
jhsdb clhsdb --core {{path/to/core_dump}} --exe {{path/to/jdk_or_bin_or_java}}
```

- Start a remote debug server:

```
jhsdb debugd --pid {{pid}} --serverid {{optional_unique_id}}
```

- Connect to a process in interactive debug mode:

```
jhsdb clhsdb --pid {{pid}}
```

jigsaw

A Laravel-based static site builder for PHP.

More information: https://jigsaw.tighten.co.

- Initialize a project:

  ```
  jigsaw init
  ```

- Initialize a project using a starter template:

  ```
  jigsaw init {{template_name}}
  ```

- Build the site for development:

  ```
  jigsaw build
  ```

- Preview the site from the "build_local" directory:

  ```
  jigsaw serve
  ```

- Build the site for production:

  ```
  jigsaw build production
  ```

- Preview the site from the "build_production" directory:

  ```
  jigsaw serve {{build_production}}
  ```

jira

Interface (third party) for interacting with Jira.

More information: https://github.com/ankitpokhrel/jira-cli.

- List recent issues:

```
jira issue list
```

- List issues from the current sprint, assigned to me:

```
jira sprint list --current {{[-a|--assignee]}} $(jira me)
```

- Create a new issue, optionally set a parent issue:

```
jira issue create {{[-P|--parent]}} {{parent}}
```

jj abandon

Abandon a revision, rebasing descendants onto its parent(s).

Abandoning a revision removes its associated change ID.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-abandon.

- Abandon revisions specified by given revsets (e.g. `B::D`, `A..D`, `B|C|D`, etc.):

  ```
  jj abandon {{revsets}}
  ```

- Abandon revisions, without deleting their bookmarks and moving them to the parent revisions instead:

  ```
  jj abandon --retain-bookmarks {{revsets}}
  ```

- Abandon revisions, without modifying the contents of their children:

  ```
  jj abandon --restore-descendants {{revsets}}
  ```

jj absorb

Split changes in the source revision and move each change to the closest mutable ancestor where the corresponding lines were modified last.

Changes that have zero or multiple matching regions in ancestral revisions won't be moved.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-absorb.

- Move all eligible and unambiguous changes from a revision to other revisions automatically:

```
jj absorb {{[-f|--from]}} {{revset}} {{[-t|--into]}} {{revsets}}
```

- Move only changes in given files from a revision to other revisions:

```
jj absorb {{[-f|--from]}} {{revset}} {{[-t|--into]}} {{revsets}} {{filesets}}
```

jj bookmark

Manage bookmarks in a **`jj`** repository.

When using a Git backend, bookmarks correspond to Git branches.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-bookmark.

- Create a new bookmark at the given revision:

```
jj {{[b|bookmark]}} {{[c|create]}} {{[-r|--revision]}} {{revision}} {{name}}
```

- List all existing bookmarks and their targets:

```
jj {{[b|bookmark]}} {{[l|list]}}
```

- Move an existing bookmark to another revision:

```
jj {{[b|bookmark]}} {{[m|move]}} {{[-t|--to]}} {{revision}} {{name}}
```

- Track given remote bookmarks:

```
jj {{[b|bookmark]}} {{[t|track]}} {{name}}@{{remote}}
```

- Delete a bookmark, and propagate the deletion to remotes on the next push:

```
jj {{[b|bookmark]}} {{[d|delete]}} {{name}}
```

- Forget a bookmark locally, without marking its deletion to be pushed:

```
jj {{[b|bookmark]}} {{[f|forget]}} {{name}}
```

jj commit

Update the description and create a new change on top.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-commit.

- Open editor to write the commit message and then create a new empty commit on top:

  ```
  jj commit
  ```

- Commit with the given message:

  ```
  jj commit {{[-m|--message]}} "{{message}}"
  ```

- Interactively choose which changes to include:

  ```
  jj commit {{[-i|--interactive]}}
  ```

jj describe

Update the change description or other metadata.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-describe.

- Update the description of the current change:

```
jj {{[desc|describe]}}
```

- Update the description of given revsets:

```
jj {{[desc|describe]}} {{revsets}}
```

- Update the description to the given message:

```
jj {{[desc|describe]}} {{[-m|--message]}} {{message}}
```

jj diff

Compare file contents between two revisions.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-diff.

- Show changes of current revision:

```
jj diff
```

- Show changes of given revsets (e.g. `B::D`, `A..D`, `B|C|D`, etc.):

```
jj diff {{[-r|--revisions]}} {{revsets}}
```

- Show changes from given revision to given revision:

```
jj diff {{[-f|--from]}} {{from_revset}} {{[-t|--to]}}
{{to_revset}}
```

- Show diff statistics:

```
jj diff --stat
```

- Show a Git-format diff:

```
jj diff --git
```

jj duplicate

Create new changes with the same content as existing ones.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-duplicate.

- Duplicate the current revision onto its existing parent:

```
jj duplicate
```

- Duplicate a specific revision onto its existing parent:

```
jj duplicate {{revset}}
```

- Duplicate a revision onto a different parent:

```
jj duplicate --destination {{dest_revset}} {{revset}}
```

- Duplicate a revision and insert it after other revision(s):

```
jj duplicate --insert-after {{after_revset}} {{revset}}
```

- Duplicate a revision and insert it before other revision(s):

```
jj duplicate --insert-before {{before_revset}} {{revset}}
```

- Duplicate onto multiple parents (creates a merge commit):

```
jj duplicate --destination {{destination1}} --destination
{{destination2}} {{revset}}
```

- Duplicate multiple revisions:

```
jj duplicate {{revset1 revset2 ...}}
```

jj edit

Set the specified revision as the working-copy revision.

Note: It is generally recommended to instead use **`jj new`** and **`jj squash`**.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-edit.

- Set the given revision as the working copy:

```
jj edit {{revset}}
```

jj evolog

Show how a change has evolved over time, listing the previous commits it has pointed to.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-evolog.

- Show how a revision has evolved over time:

```
jj evolog {{[-r|--revisions]}} {{revsets}}
```

- Show diff statistics in the evolution log:

```
jj evolog {{[-r|--revisions]}} {{revsets}} --stat
```

- Show summary of each change in the evolution log:

```
jj evolog {{[-r|--revisions]}} {{revsets}} {{[-s|--summary]}}
```

jj git clone

Create a new repo backed by a clone of a Git repo.

Note: Unless **`--colocate`** is used, it is not a valid Git repository and **`git`** commands can't be used on it.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-git-clone.

- Create a new repo backed by a clone of a Git repo into a new directory (the default directory is the repository name):

```
jj git clone {{source}} {{path/to/directory}}
```

- Create a clone and use the given name for newly created remote:

```
jj git clone --remote {{remote_name}} {{source}}
```

- Clone a Git repo, only fetching the 10 most recent commits:

```
jj git clone --depth {{10}} {{source}}
```

- Clone colocating the Jujutsu repo with the Git repo (allowing the use of both `jj` and `git` commands in the same directory):

```
jj git clone --colocate {{source}}
```

jj git fetch

Fetch from a Git remote, downloading objects and refs from the remote repository.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-git-fetch.

- Fetch the latest changes from the default remote repository:

```
jj git fetch
```

- Fetch the latest changes from a given remote repository:

```
jj git fetch --remote {{remote}}
```

- Fetch the latest changes only from given branches:

```
jj git fetch {{[-b|--branch]}} {{branch}}
```

- Fetch the latest changes from all remotes:

```
jj git fetch --all-remote
```

jj git init

Create a new Git backed Jujutsu repo.

Note: Unless **`--colocate`** is used, it is not a valid Git repository and **`git`** commands can't be used on it.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-git-init.

- Create a new Git backed repo in the current directory:

```
jj git init
```

- Create a new Git backed repo in the given directory:

```
jj git init {{path/to/directory}}
```

- Initialize the Jujutsu repository as a valid Git repository (allowing the use of both `jj` and `git` commands in the same directory):

```
jj git init --colocate
```

- Initialize the Jujutsu repository backed by an existing Git repository:

```
jj git init --git-repo {{git_repo}}
```

jj git push

Push to a Git remote.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-git-push.

- Push a bookmark to the given remote (defaults to `git.push` setting):

```
jj git push {{[-b|--bookmark]}} {{bookmark}} --remote {{remote}}
```

- Push a new bookmark:

```
jj git push {{[-b|--bookmark]}} {{bookmark}} {{[-N|--allow-new]}}
```

- Push all tracked bookmarks:

```
jj git push --tracked
```

- Push all bookmarks (including new bookmarks):

```
jj git push --all
```

- Push all bookmarks pointing to given revisions:

```
jj git push {{[-r|--revisions]}} {{revset}}
```

- Push changes/commits by creating new bookmarks (Name format is as per `templates.git_push_bookmark` setting, defaults to `"push-" ++ change_id.short()`):

```
jj git push {{[-c|--change]}} {{revset}}
```

- Push a revision with the given name:

```
jj git push --named {{name}}={{revision}}
```

jj git remote

Manage Git remotes.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-git-remote.

- List all Git remotes:

```
jj git remote list
```

- Add a Git remote:

```
jj git remote add {{remote}} {{url}}
```

- Change the URL of a Git remote:

```
jj git remote set-url {{remote}} {{url}}
```

- Remove a Git remote:

```
jj git remote remove {{remote}}
```

- Rename a Git remote:

```
jj git remote rename {{old_name}} {{new_name}}
```

jj git

Run Git-related commands for a `jj` repository.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-git.

- Create a new Git backed repository:

```
jj git init
```

- Create a new repository backed by a clone of a Git repository:

```
jj git clone {{source}}
```

- Fetch from a Git remote:

```
jj git fetch
```

- Push all tracked bookmarks to Git remote:

```
jj git push
```

- Push given bookmark to Git remote:

```
jj git push {{[-b|--bookmark]}} {{bookmark}}
```

jj log

Show revision history as a graph.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-log.

- Show revision history as a graph:

```
jj log
```

- Show only given revsets (e.g. `B::D`, `A..D`, `B|C|D`, etc.):

```
jj log {{[-r|--revisions]}} {{revsets}}
```

- Show log with a particular template for each line (e.g. 5 characters of commit hash and author):

```
jj log {{[-T|--template]}} 'commit_id.shortest(5) ++ " " ++ author'
```

jj new

Create a new empty change.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-new.

- Create a new empty change on top of current revision:

```
jj new
```

- Create a new empty change on top of specific revision:

```
jj new {{revision}}
```

- Create a new merge change on top of multiple revisions:

```
jj new {{revset1 revset2 ...}}
```

- Create a new empty change before and after specified revisions:

```
jj new {{[-B|--insert-before]}} {{revsets}} {{[-A|--insert-after]}} {{revsets}}
```

jj operation

Work with the operation log of a `jj` repository.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-operation.

- Show operation log:

```
jj {{[op|operation]}} log
```

- Undo the last operation:

```
jj {{[op|operation]}} undo
```

- Undo a given operation:

```
jj {{[op|operation]}} undo {{operation}}
```

- Restore the repository to its state at a given operation:

```
jj {{[op|operation]}} restore {{operation}}
```

- Show changes to the repository in an operation:

```
jj {{[op|operation]}} show {{operation}}
```

- Show stat, summary and patch of modifications of an operation:

```
jj {{[op|operation]}} show {{--stat}} {{[-s|--summary]}} {{[-p|--patch]}} {{operation}}
```

jj parallelize

Parallelize revisions by making them siblings.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-parallelize.

- Parallelize given revisions:

```
jj parallelize {{revsets}}
```

jj rebase

Move revisions to different parent(s).

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-rebase.

- Move given revisions to a different parent(s):

```
jj rebase {{[-r|--revisions]}} {{revset}} {{[-d|--
destination]}} {{revset}}
```

- Move given revisions and all their descendants:

```
jj rebase {{[-s|--source]}} {{revset}} {{[-d|--destination]}}
{{revset}}
```

- Move all revisions in the branch containing given revisions:

```
jj rebase {{[-b|--branch]}} {{revset}} {{[-d|--destination]}}
{{revset}}
```

- Move revisions to before and/or after other revisions:

```
jj rebase {{[-r|--revisions]}} {{revset}} {{[-B|--insert-
before]}} {{revset}} {{[-A|--insert-after]}} {{revset}}
```

jj resolve

Resolve conflicted files with an external merge tool.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-resolve.

- Resolve all conflicts:

```
jj resolve
```

- List all conflicts:

```
jj resolve {{[-l|--list]}}
```

- Resolve conflicts in a given revision:

```
jj resolve {{[-r|--revision]}} {{revset}}
```

- Resolve conflicts in given files:

```
jj resolve {{file1 file2 ...}}
```

- Resolve accepting the incoming versions:

```
jj resolve --tool :theirs
```

- Resolve accepting the outgoing versions:

```
jj resolve --tool :ours
```

jj restore

Restore files from another revision.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-restore.

- Restore files from a revision into another revision:

```
jj restore {{[-f|--from]}} {{revset}} {{[-t|--into]}} {{revset}} {{filesets}}
```

- Undo the changes in a revision as compared to the merge of its parents:

```
jj restore {{[-c|--changes-in]}} {{revset}} {{filesets}}
```

- Interactively choose what parts to restore:

```
jj restore {{[-f|--from]}} {{revset}} {{[-t|--into]}} {{revset}} {{[-i|--interactive]}}
```

jj revert

Apply the reverse of the given revision(s).

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-revert.

- Apply the reverse of the revisions specified by the given revsets (e.g. `B::D`, `A..D`, `B|C|D`, etc.):

```
jj revert {{[-r|--revisions]}} {{revsets}}
```

- Apply the reverse on top of specified revisions:

```
jj revert {{[-r|--revisions]}} {{revsets}} {{[-d|--
destination]}} {{revsets}}
```

- Apply the reverse before and/or after specified revisions:

```
jj revert {{[-r|--revisions]}} {{revsets}} {{[-B|--insert-
before]}} {{revsets}} {{[-A|--insert-after]}} {{revsets}}
```

jj show

Show commit description and changes in a revision.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-show.

- Show commit description and changes in a revision:

```
jj show {{revset}}
```

- Show a summary of changes in a revision:

```
jj show {{[-s|--summary]}} {{revset}}
```

- Show a histogram of changes in a revision:

```
jj show --stat {{revset}}
```

jj simplify-parents

Simplify parent edges for the specified revision(s).

For example, "A -> B -> C | A -> C" gets simplified to "A -> B -> C".

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-simplify-parents.

- Simplify parent edges of given revisions:

```
jj simplify-parents {{[-r|--revisions]}} {{revsets}}
```

- Simplify parent edges of given revisions and trees of their descendants:

```
jj simplify-parents {{[-s|--source]}} {{revsets}}
```

jj split

Split a revision in two.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-split.

- Split the given revision into two interactively, putting the second revision on top of it:

```
jj split {{[-r|--revision]}} {{revision}}
```

- Split out matching files from the given revision:

```
jj split {{[-r|--revision]}} {{revision}} {{fileset}}
```

- Split the given revision, putting the second revision on top of given destination(s):

```
jj split {{[-r|--revision]}} {{revision}} {{[-d|--destination]}} {{revset}}
```

- Split the given revision, putting the second revision before and/or after other revision(s):

```
jj split {{[-r|--revision]}} {{revision}} {{[-B|--insert-before]}} {{revset}} {{[-A|--insert-after]}} {{revset}}
```

- Split the given revision into two parallel revisions:

```
jj split {{[-r|--revision]}} {{revision}} {{[-p|--parallel]}}
```

jj squash

Move changes from a revision into another revision.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-squash.

- Move all changes from current revision to its parent:

```
jj squash
```

- Move all changes from given revision to its parent:

```
jj squash {{[-r|--revision]}} {{revset}}
```

- Move all changes from given revision(s) to given other revision:

```
jj squash {{[-f|--from]}} {{revsets}} {{[-t|--into]}}
{{revset}}
```

- Interactively choose which parts to squash:

```
jj squash {{[-i|--interactive]}}
```

jj status

Show high-level repository status.

This includes the working copy commit and its parents, and a summary of the changes in the working copy and any existing conflicts in the working copy.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-status.

- Show high-level status of the repository:

```
jj {{[st|status]}}
```

jj undo

Undo the most recent recorded operation in a `jj` repository.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/#jj-undo.

- Undo the last operation:

```
jj undo
```

jj

Jujutsu, a version control system.

Some subcommands such as **`log`**, **`desc`**, **`new`**, **`git`**, etc. have their own usage documentation.

More information: https://jj-vcs.github.io/jj/latest/cli-reference/.

- Update description of the revisions specified by given revsets (e.g. `B::D`, `A..D`, `B|C|D`, etc.):

```
jj {{[desc|describe]}} {{[-m|--message]}} "{{message}}" {{[-r|--revision]}} {{revsets}}
```

- Create a new commit/revision on top of a given revision:

```
jj new {{revset}}
```

- Create a new merge commit on top of multiple revisions:

```
jj new {{revset1 revset2 ...}}
```

- Update the working copy to point to a revision:

```
jj edit {{revset}}
```

- Undo the previous command (which may itself have been `undo`):

```
jj undo
```

- Execute a jj subcommand without snapshotting the working copy:

```
jj --ignore-working-copy {{subcommand}}
```

- Execute a jj subcommand at an operation:

```
jj {{[--at-op|--at-operation]}} {{operation}} {{subcommand}}
```

- Display help for a specific subcommand (like `new`, `commit`, `desc`, etc.):

```
jj help {{subcommand}}
```

jmap

Java memory map tool.

More information: https://docs.oracle.com/en/java/javase/20/docs/specs/man/jmap.html.

- Print shared object mappings for a Java process (output like pmap):

```
jmap {{java_pid}}
```

- Print heap summary information:

```
jmap -heap {{filename.jar}} {{java_pid}}
```

- Print histogram of heap usage by type:

```
jmap -histo {{java_pid}}
```

- Dump contents of the heap into a binary file for analysis with jhat:

```
jmap -dump:format=b,file={{path/to/file}} {{java_pid}}
```

- Dump live objects of the heap into a binary file for analysis with jhat:

```
jmap -dump:live,format=b,file={{path/to/file}} {{java_pid}}
```

jmeter

Open source Java application designed for load testing functional behavior and measure performance.

More information: https://jmeter.apache.org.

- Run a specific test plan in nongui mode:

```
jmeter --nongui --testfile {{path/to/file.jmx}}
```

- Run a test plan in nongui mode using a specific log file:

```
jmeter --nogui --testfile {{path/to/file.jmx}} --logfile
{{path/to/logfile.jtl}}
```

- Run a test plan in nongui mode using a specific proxy:

```
jmeter --nongui --testfile {{path/to/file.jmx}} --proxyHost
{{127.0.0.1}} --proxyPort {{8888}}
```

- Run a test plan in nongui mode using a specific JMeter property:

```
jmeter --jmeterproperty {{key}}='{{value}}' --nongui --
testfile {{path/to/file.jmx}}
```

jmtpfs

FUSE-based filesystem for accessing MTP devices.

More information: https://manned.org/jmtpfs.

- Mount an MTP device to a directory:

```
jmtpfs {{path/to/directory}}
```

- Set mount options:

```
jmtpfs -o {{allow_other,auto_unmount}} {{path/to/directory}}
```

- List available MTP devices:

```
jmtpfs {{[-l|--listDevices]}}
```

- If multiple devices are present, mount a specific device:

```
jmtpfs -device={{bus_id}},{{device_id}} {{path/to/directory}}
```

- Unmount MTP device:

```
fusermount -u {{path/to/directory}}
```

jo

This command is an alias of **autojump**.

- View documentation for the original command:

```
tldr autojump
```

jobs

Display status of jobs in the current session.

More information: https://manned.org/jobs.

- Show status of all jobs:

```
jobs
```

- Show status of a particular job:

```
jobs %{{job_id}}
```

- Show status and process IDs of all jobs:

```
jobs -l
```

- Show process IDs of all jobs:

```
jobs -p
```

joe

Joe's own text editor.

More information: https://joe-editor.sourceforge.io.

- Open a new file in JOE:

```
joe
```

- Open a specific file:

```
joe {{path/to/file}}
```

- Open a specific file, positioning the cursor at the specified line:

```
joe +{{line}} {{path/to/file}}
```

- Open a specific file in read-only mode:

```
joe -rdonly {{path/to/file}}
```

john

Password cracker.

More information: https://www.openwall.com/john/.

- Crack password hashes:

```
john {{path/to/hashes.txt}}
```

- Show passwords cracked:

```
john --show {{path/to/hashes.txt}}
```

- Display users' cracked passwords by user identifier from multiple files:

```
john --show --users={{user_ids}} {{path/to/hashes1.txt path/to/hashes2.txt ...}}
```

- Crack password hashes, using a custom wordlist:

```
john --wordlist={{path/to/wordlist.txt}} {{path/to/hashes.txt}}
```

- List available hash formats:

```
john --list=formats
```

- Crack password hashes, using a specific hash format:

```
john --format={{md5crypt}} {{path/to/hashes.txt}}
```

- Crack password hashes, enabling word mangling rules:

```
john --rules {{path/to/hashes.txt}}
```

- Restore an interrupted cracking session from a state file, e.g. `mycrack.rec`:

```
john --restore={{path/to/mycrack.rec}}
```

join

Join lines of two sorted files on a common field.

More information: https://www.gnu.org/software/coreutils/manual/html_node/join-invocation.html.

- Join two files on the first (default) field:

```
join {{path/to/file1}} {{path/to/file2}}
```

- Join two files using a comma (instead of a space) as the field separator:

```
join -t ',' {{path/to/file1}} {{path/to/file2}}
```

- Join field3 of file1 with field1 of file2:

```
join -1 {{3}} -2 {{1}} {{path/to/file1}} {{path/to/file2}}
```

- Produce a line for each unpairable line for file1:

```
join -a {{1}} {{path/to/file1}} {{path/to/file2}}
```

- Join a file from `stdin`:

```
cat {{path/to/file1}} | join - {{path/to/file2}}
```

josm

Extensible OpenStreetMap editor for Java 8+.

More information: https://manned.org/josm.

- Launch JOSM:

  ```
  josm
  ```

- Launch JOSM in maximized mode:

  ```
  josm --maximize
  ```

- Launch JOSM and set a specific language:

  ```
  josm --language {{de}}
  ```

- Launch JOSM and reset all preferences to their default values:

  ```
  josm --reset-preferences
  ```

- Launch JOSM and download a specific bounding box:

  ```
  josm --download {{minlat,minlon,maxlat,maxlon}}
  ```

- Launch JOSM and download a specific bounding box as raw GPS:

  ```
  josm --downloadgps {{minlat,minlon,maxlat,maxlon}}
  ```

- Launch JOSM without plugins:

  ```
  josm --skip-plugins
  ```

jp2a

Convert JPEG images to ASCII.

More information: https://manned.org/jp2a.

- Read JPEG image from a file and print in ASCII:

```
jp2a {{path/to/image.jpeg}}
```

- Read JPEG image from a URL and print in ASCII:

```
jp2a {{www.example.com/image.jpeg}}
```

- Colorize the ASCII output:

```
jp2a --colors {{path/to/image.jpeg}}
```

- Specify characters to be used for the ASCII output:

```
jp2a --chars='{{..-ooxx@@}}' {{path/to/image.jpeg}}
```

- Write the ASCII output into a file:

```
jp2a --output={{path/to/output_file.txt}} {{path/to/image.jpeg}}
```

- Write the ASCII output in HTML file format, suitable for viewing in web browsers:

```
jp2a --html --output={{path/to/output_file.html}} {{path/to/image.jpeg}}
```

jpegoptim

Optimise JPEG images.

More information: https://manned.org/jpegoptim.

- Optimise a set of JPEG images, retaining all associated data:

```
jpegoptim {{image1.jpeg image2.jpeg image3.jpeg ...}}
```

- Optimise JPEG images, stripping all non-essential data:

```
jpegoptim {{[-s|--strip-all]}} {{image1.jpeg image2.jpeg image3.jpeg ...}}
```

- Force the output images to be progressive:

```
jpegoptim --all-progressive {{image1.jpeg image2.jpeg image3.jpeg ...}}
```

- Force the output images to have a fixed maximum filesize:

```
jpegoptim {{[-S|--size]}} {{250k}} {{image1.jpeg image2.jpeg image3.jpeg ...}}
```

jpegtopnm

Convert a JPEG/JFIF file to the PPM or PGM format.

More information: https://netpbm.sourceforge.net/doc/jpegtopnm.html.

- Convert JPEG/JFIF image to a PPM or PGM image:

```
jpegtopnm {{path/to/file.jpg}} > {{path/to/file.pnm}}
```

- Display version:

```
jpegtopnm {{[-v|-version]}}
```

jps

Show JVM process status of current user.

More information: https://docs.oracle.com/en/java/javase/20/docs/specs/man/jps.html.

- List all JVM processes:

```
jps
```

- List all JVM processes with only PID:

```
jps -q
```

- Display the arguments passed to the processes:

```
jps -m
```

- Display the full package name of all processes:

```
jps -l
```

- Display the arguments passed to the JVM:

```
jps -v
```

jq

A JSON processor that uses a domain-specific language (DSL).

More information: https://jqlang.github.io/jq/manual/.

- Execute a specific expression only using the `jq` binary (print a colored and formatted JSON output):

```
jq '.' {{path/to/file.json}}
```

- Execute a specific script:

```
{{cat path/to/file.json}} | jq {{[-f|--from-file]}} {{path/
to/script.jq}}
```

- Pass specific arguments:

```
{{cat path/to/file.json}} | jq {{--arg "name1" "value1" --arg
"name2" "value2" ...}} '{{. + $ARGS.named}}'
```

- Create new JSON object via old JSON objects from multiple files:

```
{{cat path/to/multiple_json_file_*.json}} | jq
'{{{newKey1: .key1, newKey2: .key2.nestedKey, ...}}}'
```

- Print specific array items:

```
{{cat path/to/file.json}} | jq '{{.[index1], .
[index2], ...}}'
```

- Print all array/object values:

```
{{cat path/to/file.json}} | jq '.[]'
```

- Print objects with 2-condition filter in array:

```
{{cat path/to/file.json}} | jq '.[] |
select((.key1=="value1") and .key2=="value2")'
```

- Add/remove specific keys:

```
{{cat path/to/file.json}} | jq '. {{+|-}} {{{"key1":
"value1", "key2": "value2", ...}}}'
```

jrnl

A simple journal application.

More information: https://jrnl.sh.

- Insert a new entry with your editor:

```
jrnl
```

- Quickly insert a new entry:

```
jrnl {{today at 3am}}: {{title}}. {{content}}
```

- View the last ten entries:

```
jrnl -n {{10}}
```

- View everything that happened from the start of last year to the start of last march:

```
jrnl -from "{{last year}}" -until {{march}}
```

- Edit all entries tagged with "texas" and "history":

```
jrnl {{@texas}} -and {{@history}} --edit
```

json5

Convert JSON5 files to JSON.

More information: https://json5.org.

- Convert JSON5 `stdin` to JSON `stdout`:

```
echo {{input}} | json5
```

- Convert a JSON5 file to JSON and output to `stdout`:

```
json5 {{path/to/input_file.json5}}
```

- Convert a JSON5 file to the specified JSON file:

```
json5 {{path/to/input_file.json5}} --out-file {{path/to/output_file.json}}
```

- Validate a JSON5 file:

```
json5 {{path/to/input_file.json5}} --validate
```

- Specify the number of spaces to indent by (or "t" for tabs):

```
json5 --space {{indent_amount}}
```

- Display help:

```
json5 --help
```

jstack

Java stack trace tool.

More information: https://manned.org/jstack.

- Print Java stack traces for all threads in a Java process:

```
jstack {{java_pid}}
```

- Print mixed mode (Java/C++) stack traces for all threads in a Java process:

```
jstack -m {{java_pid}}
```

- Print stack traces from Java core dump:

```
jstack {{/usr/bin/java}} {{file.core}}
```

jtbl

Utility to print JSON and JSON Lines data as a table in the terminal.

More information: https://github.com/kellyjonbrazil/jtbl.

- Print a table from JSON or JSON Lines input:

```
cat {{file.json}} | jtbl
```

- Print a table and specify the column width for wrapping:

```
cat {{file.json}} | jtbl --cols={{width}}
```

- Print a table and truncate rows instead of wrapping:

```
cat {{file.json}} | jtbl {{[-t|--truncate]}}
```

- Print a table and don't wrap or truncate rows:

```
cat {{file.json}} | jtbl {{[-n|--no-wrap]}}
```

julia

A high-level, high-performance dynamic programming language for technical computing.

More information: https://docs.julialang.org/en/v1/manual/getting-started/.

- Start a REPL (interactive shell):

```
julia
```

- Execute a Julia program and exit:

```
julia {{program.jl}}
```

- Execute a Julia program that takes arguments:

```
julia {{program.jl}} {{arguments}}
```

- Evaluate a string containing Julia code:

```
julia {{[-e|--eval]}} '{{julia_code}}'
```

- Evaluate a string of Julia code, passing arguments to it:

```
julia {{[-e|--eval]}} '{{for x in ARGS; println(x);
end}}' {{arguments}}
```

- Evaluate an expression and print the result:

```
julia {{[-E|--print]}} '{{(1 - cos(pi/4))/2}}'
```

- Start Julia in multithreaded mode, using `n` threads:

```
julia {{[-t|--threads]}} {{n}}
```

jupyter lab

Interactive development environment for Jupyter notebooks.

More information: https://jupyterlab.readthedocs.io/en/stable/.

- Start JupyterLab:

  ```
  jupyter lab
  ```

- Open a specific notebook:

  ```
  jupyter lab {{path/to/notebook.ipynb}}
  ```

- Start JupyterLab in a specific directory:

  ```
  jupyter lab --notebook-dir {{path/to/directory}}
  ```

- Start JupyterLab in debug mode:

  ```
  jupyter lab --debug
  ```

jupyter

Web application to create and share documents that contain code, visualizations and notes.

Primarily used for data analysis, scientific computing and machine learning.

More information: https://docs.jupyter.org/en/latest/.

- Start a Jupyter notebook server in the current directory:

```
jupyter notebook
```

- Open a specific Jupyter notebook:

```
jupyter notebook {{example.ipynb}}
```

- Export a specific Jupyter notebook into another format:

```
jupyter nbconvert --to {{html|markdown|pdf|script}}
{{example.ipynb}}
```

- Start a server on a specific port:

```
jupyter notebook --port {{port}}
```

- List currently running notebook servers:

```
jupyter notebook list
```

- Stop the currently running server:

```
jupyter notebook stop
```

- Start JupyterLab, if installed, in the current directory:

```
jupyter lab
```

jupyterlab

This command is an alias of **`jupyter lab`**.

- View documentation for the original command:

```
tldr jupyter lab
```

jupytext

Convert Jupyter notebooks to plain text documents, and back again.

More information: https://jupytext.readthedocs.io.

- Turn a notebook into a paired `.ipynb`/`.py` notebook:

```
jupytext --set-formats ipynb,py {{notebook.ipynb}}
```

- Convert a notebook to a `.py` file:

```
jupytext --to py {{notebook.ipynb}}
```

- Convert a `.py` file to a notebook with no outputs:

```
jupytext --to notebook {{notebook.py}}
```

- Convert a `.md` file to a notebook and run it:

```
jupytext --to notebook --execute {{notebook.md}}
```

- Update the input cells in a notebook and preserve outputs and metadata:

```
jupytext --update --to notebook {{notebook.py}}
```

- Update all paired representations of a notebook:

```
jupytext --sync {{notebook.ipynb}}
```

just

Save and run project-specific commands.

More information: https://github.com/casey/just.

- Run a recipe specified in the justfile:

```
just {{recipe}}
```

- Initialize new justfile in project root:

```
just --init
```

- Edit justfile in the default editor:

```
just {{[-e|--edit]}}
```

- List available recipes in the justfile:

```
just {{[-l|--list]}}
```

- Print justfile:

```
just --dump
```

just

just can refer to multiple commands with the same name.

- View documentation for the command runner:

  `tldr just.1`

- View documentation for the V8 JavaScript runtime:

  `tldr just.js`

jwt

Work with JSON Web Tokens (JWTs).

Encryption algorithms available are HS256, HS384, HS512, RS256, RS384, RS512, ES256, ES384.

More information: https://github.com/mike-engel/jwt-cli.

- Decode a JWT:

```
jwt decode {{jwt_string}}
```

- Decode a JWT as a JSON string:

```
jwt decode {{[-j|--json]}} {{jwt_string}}
```

- Encode a JSON string to a JWT:

```
jwt encode {{[-A|--alg]}} {{HS256}} {{[-S|--secret]}}
{{1234567890}} '{{json_string}}'
```

- Encode key pair payload to JWT:

```
jwt encode {{[-A|--alg]}} {{HS256}} {{[-S|--secret]}}
{{1234567890}} {{[-P|--payload]}} {{key=value}}
```

k3d

A wrapper to easily create k3s clusters inside Docker.

More information: https://k3d.io.

- Create a cluster:

```
k3d cluster create {{cluster_name}}
```

- Delete a cluster:

```
k3d cluster delete {{cluster_name}}
```

- Create a new containerized k3s node:

```
k3d node create {{node_name}}
```

- Import an image from Docker into a k3d cluster:

```
k3d image import {{image_name}} --cluster {{cluster_name}}
```

- Create a new registry:

```
k3d registry create {{registry_name}}
```

k3s

Install and manage lightweight Kubernetes clusters.

More information: https://docs.k3s.io/cli.

- Run the embedded `kubectl` command:

  ```
  k3s kubectl get nodes
  ```

- Take an etcd snapshot of the cluster:

  ```
  k3s etcd-snapshot save
  ```

- Rotate the CA certificate:

  ```
  k3s certificate rotate-ca
  ```

- Manage bootstrap tokens:

  ```
  k3s token list
  ```

- Uninstall K3s and remove all components:

  ```
  k3s-uninstall.sh
  ```

k6

Open source load testing tool and SaaS for engineering teams.

More information: https://k6.io.

- Run load test locally:

```
k6 run {{script.js}}
```

- Run load test locally with a given number of virtual users and duration:

```
k6 run {{[-u|--vus]}} {{10}} {{[-d|--duration]}} {{30s}} {{script.js}}
```

- Run load test locally with a given environment variable:

```
k6 run {{[-e|--env]}} {{HOSTNAME=example.com}} {{script.js}}
```

- Run load test locally using InfluxDB to store results:

```
k6 run {{[-o|--out]}} influxdb={{http://localhost:8086/k6db}} {{script.js}}
```

- Run load test locally and discard response bodies (significantly faster):

```
k6 run --discard-response-bodies {{script.js}}
```

- Run load test locally using the base JavaScript compatibility mode (significantly faster):

```
k6 run --compatibility-mode=base {{script.js}}
```

- Log in to cloud service using secret token:

```
k6 login cloud --token {{secret}}
```

- Run load test on cloud infrastructure:

```
k6 cloud {{script.js}}
```

k8s-unused-secret-detector

Detect unused Kubernetes secrets.

More information: https://github.com/dtan4/k8s-unused-secret-detector.

- Detect unused secrets:

```
k8s-unused-secret-detector
```

- Detect unused secrets in a specific namespace:

```
k8s-unused-secret-detector {{[-n|--namespace]}} {{namespace}}
```

- Delete unused secrets in a specific namespace:

```
k8s-unused-secret-detector {{[-n|--namespace]}} {{namespace}}
| kubectl delete secrets {{[-n|--namespace]}} {{namespace}}
```

k8sec

Manage Kubernetes secrets.

More information: https://github.com/dtan4/k8sec.

- List all secrets:

```
k8sec list
```

- List a specific secret as a base64-encoded string:

```
k8sec list {{secret_name}} --base64
```

- Set a secret's value:

```
k8sec set {{secret_name}} {{key=value}}
```

- Set a base64-encoded value:

```
k8sec set --base64 {{secret_name}} {{key=encoded_value}}
```

- Unset a secret:

```
k8sec unset {{secret_name}}
```

- Load secrets from a file:

```
k8sec load {{[-f|--filename]}} {{path/to/file}}
{{secret_name}}
```

- Dump secrets to a file:

```
k8sec dump {{[-f|--filename]}} {{path/to/file}}
{{secret_name}}
```

k9s

View and manage Kubernetes clusters.

More information: https://k9scli.io/topics/commands/.

- Manage a cluster using a kubeconfig context:

```
k9s --context {{kubeconfig_context_name}}
```

- Manage a cluster in read-only mode (disabling all commands that may cause modifications):

```
k9s --readonly --cluster {{cluster_name}}
```

- Manage a cluster using a given kubernetes namespace:

```
k9s {{[-n|--namespace]}} {{kubernetes_namespace}} --cluster {{cluster_name}}
```

- Manage a cluster launching k9s in the pod view and enable debug logging:

```
k9s {{[-c|--command]}} {{pod}} {{[-l|--logLevel]}} debug --cluster {{cluster_name}}
```

kafkacat

This command is an alias of **kcat**.

- View documentation for the original command:

```
tldr kcat
```

kaggle competitions

Manage Kaggle competitions from the command-line.

More information: https://github.com/Kaggle/kaggle-api/blob/main/docs/README.md#competitions.

- List all competitions:

```
kaggle {{[c|competitions]}} list
```

- Download competition data:

```
kaggle {{[c|competitions]}} download {{competition_name}}
```

- Download specific file:

```
kaggle {{[c|competitions]}} download {{competition_name}} {{[-f|--file]}} {{file}}
```

- Submit files to a competition:

```
kaggle {{[c|competitions]}} submit {{competition_name}} {{[-f|--file]}} {{path/to/file}} {{[-m|--message]}} "{{message}}"
```

- Show or download leaderboard:

```
kaggle {{[c|competitions]}} leaderboard {{competition_name}} {{--show|--download}}
```

- View submissions:

```
kaggle {{[c|competitions]}} submissions {{competition_name}}
```

kaggle datasets

Manage Kaggle datasets from command line.

More information: https://github.com/Kaggle/kaggle-api/blob/main/docs/README.md#datasets.

- List all datasets owned by a user or organization:

```
kaggle {{[d|datasets]}} list --user {{username}}
```

- Search dataset by name:

```
kaggle {{[d|datasets]}} list {{[-s|--search]}} "{{dataset_name}}"
```

- Download a dataset:

```
kaggle {{[d|datasets]}} download "{{dataset_name}}"
```

- Create a public dataset:

```
kaggle {{[d|datasets]}} create {{[-p|--path]}} {{path/to/dataset}} {{[-u|--public]}}
```

- Download metadata of dataset:

```
kaggle {{[d|datasets]}} metadata {{dataset_name}}
```

- Initialize metadata for dataset:

```
kaggle {{[d|datasets]}} init {{[-p|--path]}} {{path/to/dataset}}
```

- Delete a dataset:

```
kaggle {{[d|datasets]}} delete {{dataset_name}}
```

kaggle

Official CLI for Kaggle implemented in Python 3.

More information: https://github.com/Kaggle/kaggle-api/blob/main/docs/README.md.

- View current configuration values:

```
kaggle config view
```

- Download a specific file from a competition dataset:

```
kaggle competitions download {{competition}} {{[-f|--file]}} {{path/to/file}}
```

- List competitions matching a search term:

```
kaggle competitions list {{[-s|--search]}} {{search_term}}
```

- List files available for a specific competition:

```
kaggle competitions files {{competition}}
```

- Submit a file to a competition with a message:

```
kaggle competitions submit {{competition}} {{[-f|--file]}} {{path/to/submission.csv}} {{[-m|--message]}} "{{message}}"
```

- List datasets matching a search term:

```
kaggle datasets list {{[-s|--search]}} {{search_term}}
```

- Download all files from a dataset:

```
kaggle datasets download {{owner}}/{{dataset_name}}
```

- List kernels (notebooks) matching a search term:

```
kaggle kernels list {{[-s|--search]}} {{search_term}}
```

kahlan

A unit and Behaviour Driven Development test framework for PHP.

More information: https://kahlan.github.io/docs/cli-options.html.

- Run all specifications in the "spec" directory:

  ```
  kahlan
  ```

- Run specifications using a specific configuration file:

  ```
  kahlan --config={{path/to/configuration_file}}
  ```

- Run specifications and output using a reporter:

  ```
  kahlan --reporter={{dot|bar|json|tap|verbose}}
  ```

- Run specifications with code coverage (detail can be between 0 and 4):

  ```
  kahlan --coverage={{detail_level}}
  ```

kak

Kakoune is a mode-based code editor implementing the "multiple selections" paradigm.

Data can be selected and simultaneously edited in different locations, using multiple selections; users can also connect to the same session for collaborative editing.

More information: https://kakoune.org.

- Open a file and enter normal mode, to execute commands:

  ```
  kak {{path/to/file}}
  ```

- Enter insert mode from normal mode, to write text into the file:

  ```
  <i>
  ```

- Escape insert mode, to go back to normal mode:

  ```
  <Esc>
  ```

- Replace all instances of "foo" in the current file with "bar":

  ```
  %s{{foo}}<Enter>c{{bar}}<Esc>
  ```

- Unselect all secondary selections, and keep only the main one:

  ```
  <Space>
  ```

- Search for numbers and select the first two:

  ```
  /\d+<Enter>N
  ```

- Insert the contents of a file:

  ```
  <!>cat {{path/to/file}}<Enter>
  ```

- Save the current file:

  ```
  <:>w<Enter>
  ```

katana

A fast crawler focused on execution in automation pipelines offering both headless and non-headless crawling.

See also: **`gau`**, **`scrapy`**, **`waymore`**.

More information: https://docs.projectdiscovery.io/opensource/katana/usage.

- Crawl a list of URLs:

```
katana -list {{https://example.com,https://google.com,...}}
```

- Crawl a [u]RL using headless mode using Chromium:

```
katana -u {{https://example.com}} {{[-hl|-headless]}}
```

- Pass requests through a proxy (http/socks5) and use custom headers from a file:

```
katana -proxy {{http://127.0.0.1:8080}} {{[-H|-headers]}} {{path/to/headers.txt}} -u {{https://example.com}}
```

- Specify the crawling strategy, depth of subdirectories to crawl, and rate limiting (requests per second):

```
katana {{[-s|-strategy]}} {{depth-first|breadth-first}} {{[-d|-depth]}} {{value}} {{[-rl|-rate-limit]}} {{value}} -u {{https://example.com}}
```

- Find subdomains using `subfinder`, crawl each for a maximum number of seconds, and write results to an output file:

```
subfinder {{[-dL|-list]}} {{path/to/domains.txt}} | katana {{[-ct|-crawl-duration]}} {{value}} {{[-o|-output]}} {{path/to/output.txt}}
```

kate

KDE's advanced text editor.

More information: https://docs.kde.org/stable/en/kate/kate/fundamentals.html#starting-from-the-command-line.

- Open specific files:

```
kate {{path/to/file1 path/to/file2 ...}}
```

- Open specific remote files:

```
kate {{https://example.com/path/to/file1 https://example.com/path/to/file2 ...}}
```

- Create a new editor instance even if one is already open:

```
kate {{[-n|--new]}}
```

- Open a file with the cursor at the specific line:

```
kate {{[-l|--line]}} {{line_number}} {{path/to/file}}
```

- Open a file with the cursor at the specific line and column:

```
kate {{[-l|--line]}} {{line_number}} {{[-c|--column]}} {{column_number}} {{path/to/file}}
```

- Create a file from `stdin`:

```
cat {{path/to/file}} | kate {{[-i|--stdin]}}
```

- Display help:

```
kate {{[-h|--help]}}
```

kcadm.sh

Perform administration tasks.

More information: https://www.keycloak.org/docs/latest/server_admin/#admin-cli.

- Start an authenticated session:

```
kcadm.sh config credentials --server {{host}} --realm
{{realm_name}} --user {{username}} --password {{password}}
```

- Create a user:

```
kcadm.sh create users -s username={{username}} -r
{{realm_name}}
```

- List all realms:

```
kcadm.sh get realms
```

- Update a realm with JSON config:

```
kcadm.sh update realms/{{realm_name}} -f {{path/to/
file.json}}
```

kcat

Apache Kafka producer and consumer tool.

More information: https://github.com/edenhill/kcat.

- Consume messages starting with the newest offset:

```
kcat -C -t {{topic}} -b {{brokers}}
```

- Consume messages starting with the oldest offset and exit after the last message is received:

```
kcat -C -t {{topic}} -b {{brokers}} -o beginning -e
```

- Consume messages as a Kafka consumer group:

```
kcat -G {{group_id}} {{topic}} -b {{brokers}}
```

- Publish message by reading from `stdin`:

```
echo {{message}} | kcat -P -t {{topic}} -b {{brokers}}
```

- Publish messages by reading from a file:

```
kcat -P -t {{topic}} -b {{brokers}} {{path/to/file}}
```

- List metadata for all topics and brokers:

```
kcat -L -b {{brokers}}
```

- List metadata for a specific topic:

```
kcat -L -t {{topic}} -b {{brokers}}
```

- Get offset for a topic/partition for a specific point in time:

```
kcat -Q -t {{topic}}:{{partition}}:{{unix_timestamp}} -b
{{brokers}}
```

kdash

A simple dashboard for Kubernetes.

More information: https://github.com/kdash-rs/kdash/#usage.

- Show dashboard:

```
kdash
```

- Show dashboard in debug mode and write logs to a file in the current directory:

```
kdash {{[-d|--debug]}}
```

- Set the tick rate:

```
kdash {{[-t|--tick-rate]}} {{100}}
```

- Set the polling rate (must be a multiple of the tick rate):

```
kdash {{[-t|--tick-rate]}} {{200}} {{[-p|--poll-rate]}} {{400}}
```

kdeconnect-cli

Use KDE Connect for sharing files or text to a device, ringing it, unlocking it, and much more.

More information: https://kdeconnect.kde.org.

- List all devices:

```
kdeconnect-cli --list-devices
```

- List available (paired and reachable) devices:

```
kdeconnect-cli --list-available
```

- Request pairing with a specific device, specifying its ID:

```
kdeconnect-cli --pair --device {{device_id}}
```

- Ring a device, specifying its name:

```
kdeconnect-cli --ring --name "{{device_name}}"
```

- Share an URL or file with a paired device, specifying its ID:

```
kdeconnect-cli --share {{url|path/to/file}} --device
{{device_id}}
```

- Send an SMS with an optional attachment to a specific number:

```
kdeconnect-cli --name "{{device_name}}" --send-sms
"{{message}}" --destination {{phone_number}} --attachment
{{path/to/file}}
```

- Unlock a specific device:

```
kdeconnect-cli --name "{{device_name}}" --unlock
```

- Simulate a key press on a specific device:

```
kdeconnect-cli --name "{{device_name}}" --send-keys {{key}}
```

kdenlive

KDE's non-linear video editor.

More information: https://manned.org/kdenlive.

- Start the video editor:

```
kdenlive
```

- Open a specific file:

```
kdenlive {{path/to/file.kdenlive}}
```

- Use a specific path for an MLT environment:

```
kdenlive --mlt-path {{path/to/directory}}
```

- Use a specific log level for an MLT environment:

```
kdenlive --mlt-log {{verbose|debug}}
```

- Display help:

```
kdenlive --help
```

- Display version:

```
kdenlive --version
```

kdig

Advanced DNS lookup utility.

More information: https://www.knot-dns.cz/docs/latest/html/man_kdig.html.

- Lookup the IP(s) associated with a hostname (A records):

```
kdig {{example.com}}
```

- Specify a specific DNS server to query (e.g. Google DNS):

```
kdig {{example.com}} @{{8.8.8.8}}
```

- Query a specific DNS record type associated with a given domain name:

```
kdig {{example.com}} {{A|AAAA|NS|SOA|DNSKEY|ANY}}
```

- Lookup the IP(s) associated with a hostname (A records) using DNS over TLS (DoT):

```
kdig -d @{{8.8.8.8}} +tls-ca +tls-host={{dns.google}} {{example.com}}
```

- Lookup the IP(s) associated with a hostname (A records) using DNS over HTTPS (DoH):

```
kdig -d @{{1.1.1.1}} +https +tls-hostname={{1dot1dot1dot1.cloudflare-dns.com}} {{example.com}}
```

keep-header

Keep first line untouched by a command, passing it directly to **`stdout`**.

More information: https://github.com/eBay/tsv-utils#keep-header.

- Sort a file and keep the first line at the top:

```
keep-header {{path/to/file}} -- sort
```

- Output first line directly to `stdout`, passing the remainder of the file through the specified command:

```
keep-header {{path/to/file}} -- {{command}}
```

- Read from `stdin`, sorting all except the first line:

```
cat {{path/to/file}} | keep-header -- {{command}}
```

- Grep a file, keeping the first line regardless of the search pattern:

```
keep-header {{path/to/file}} -- grep {{pattern}}
```

keepass2

A light-weight password manager.

More information: https://keepass.info.

- Start KeePass 2, opening the most recently opened password database:

```
keepass2
```

- Start KeePass 2, opening a specific password database:

```
keepass2 {{path/to/database.kbdx}}
```

- Use a specific key file to open a password database:

```
keepass2 {{path/to/database.kbdx}} -keyfile:{{path/to/
key_file.key}}
```

keepassxc-cli

Interface for KeePassXC.

More information: https://manned.org/keepassxc-cli.

- Search entries:

```
keepassxc-cli search {{path/to/database_file}} {{name}}
```

- List the contents of a folder:

```
keepassxc-cli ls {{path/to/database_file}} {{path/to/
directory}}
```

- Add an entry with an auto-generated password:

```
keepassxc-cli add {{[-g|--generate]}} {{path/to/
database_file}} {{entry_name}}
```

- Delete an entry:

```
keepassxc-cli rm {{path/to/database_file}} {{entry_name}}
```

- Copy an entry's password to the clipboard:

```
keepassxc-cli clip {{path/to/database_file}} {{entry_name}}
```

- Copy a TOTP code to the clipboard:

```
keepassxc-cli clip {{[-t|--totp]}} {{path/to/database_file}}
{{entry_name}}
```

- Generate a passphrase with 7 words:

```
keepassxc-cli diceware {{[-W|--words]}} 7
```

- Generate a password with 16 printable ASCII characters:

```
keepassxc-cli generate {{[-lUns|--lower --upper --numeric --
special]}} {{[-L|--length]}} 16
```

kerl

Build and install Erlang/OTP instances with ease.

More information: https://github.com/kerl/kerl.

- Build and install an Erlang/OTP version in a directory:

```
kerl build-install {{28.0}} {{28.0}} {{path/to/
installation_directory}}/{{28.0}}
```

- Activate an Erlang/OTP installation:

```
. {{path/to/installation}}/activate
```

- Deactivate the current Erlang/OTP installation:

```
kerl_deactivate
```

- List all available Erlang/OTP releases:

```
kerl list releases
```

- List installed Erlang/OTP builds:

```
kerl list installations
```

keybase

Key directory that maps social media identities to encryption keys in a publicly auditable manner.

More information: https://book.keybase.io/docs/cli.

- Follow another user:

```
keybase follow {{username}}
```

- Add a new proof:

```
keybase prove {{service}} {{service_username}}
```

- Sign a file:

```
keybase sign {{[-i|--infile]}} {{input_file}} {{[-o|--outfile]}} {{output_file}}
```

- Verify a signed file:

```
keybase verify {{[-i|--infile]}} {{input_file}} {{[-o|--outfile]}} {{output_file}}
```

- Encrypt a file:

```
keybase encrypt {{[-i|--infile]}} {{input_file}} {{[-o|--outfile]}} {{output_file}} {{receiver}}
```

- Decrypt a file:

```
keybase decrypt {{[-i|--infile]}} {{input_file}} {{[-o|--outfile]}} {{output_file}}
```

- Revoke current device, log out, and delete local data:

```
keybase deprovision
```

keychain

Re-use ssh-agent and/or gpg-agent between logins.

More information: https://funtoo.org/Keychain.

- Check for a running ssh-agent, and start one if needed:

```
keychain
```

- Also check for gpg-agent:

```
keychain --agents "{{gpg,ssh}}"
```

- List signatures of all active keys:

```
keychain --list
```

- List fingerprints of all active keys:

```
keychain --list-fp
```

- Add a timeout for identities added to the agent, in minutes:

```
keychain --timeout {{minutes}}
```

keytool

A certificate management utility included with Java.

More information: https://docs.oracle.com/en/java/javase/20/docs/specs/man/keytool.html.

- Create a keystore:

```
keytool -genkeypair -v -keystore {{path/to/file.keystore}} -alias {{key_name}}
```

- Change a keystore password:

```
keytool -storepasswd -keystore {{path/to/file.keystore}}
```

- Change a key's password inside a specific keystore:

```
keytool -keypasswd -alias {{key_name}} -keystore {{path/to/file.keystore}}
```

khal

A text-based calendar and scheduling application.

More information: https://lostpackets.de/khal.

- Start Khal on interactive mode:

```
ikhal
```

- Print all events scheduled in personal calendar for the next seven days:

```
khal list -a {{personal}} {{today}} {{7d}}
```

- Print all events scheduled not in personal calendar for tomorrow at 10:00:

```
khal at -d {{personal}} {{tomorrow}} {{10:00}}
```

- Print a calendar with a list of events for the next three months:

```
khal calendar
```

- Add new event to personal calendar:

```
khal new -a {{personal}} {{2020-09-08}} {{18:00}} {{18:30}}
"{{Dentist appointment}}"
```

kill

Sends a signal to a process, usually related to stopping the process.

All signals except for SIGKILL and SIGSTOP can be intercepted by the process to perform a clean exit.

More information: https://manned.org/kill.1posix.

- Terminate a program using the default SIGTERM (terminate) signal:

```
kill {{process_id}}
```

- List available signal names (to be used without the `SIG` prefix):

```
kill -l
```

- Terminate a program using the SIGHUP (hang up) signal. Many daemons will reload instead of terminating:

```
kill -{{1|HUP}} {{process_id}}
```

- Terminate a program using the SIGINT (interrupt) signal. This is typically initiated by the user pressing `<Ctrl c>`:

```
kill -{{2|INT}} {{process_id}}
```

- Signal the operating system to immediately terminate a program (which gets no chance to capture the signal):

```
kill -{{9|KILL}} {{process_id}}
```

- Signal the operating system to pause a program until a SIGCONT ("continue") signal is received:

```
kill -{{17|STOP}} {{process_id}}
```

- Send a `SIGUSR1` signal to all processes with the given GID (group id):

```
kill -{{SIGUSR1}} -{{group_id}}
```

killall

Send kill signal to all instances of a process by name (must be exact name).

All signals except SIGKILL and SIGSTOP can be intercepted by the process, allowing a clean exit.

More information: https://manned.org/killall.

- Terminate a process using the default SIGTERM (terminate) signal:

```
killall {{process_name}}
```

- List available signal names (to be used without the `SIG` prefix):

```
killall {{[-l|--list]}}
```

- Interactively ask for confirmation before termination:

```
killall {{[-i|--interactive]}} {{process_name}}
```

- Terminate a process using the SIGINT (interrupt) signal, which is the same signal sent by pressing `<Ctrl c>`:

```
killall -INT {{process_name}}
```

- Force kill a process:

```
killall -KILL {{process_name}}
```

kind

Run local Kubernetes clusters using Docker container "nodes".

Designed for testing Kubernetes itself, but may be used for local development or continuous integration.

More information: https://github.com/kubernetes-sigs/kind.

- Create a local Kubernetes cluster:

  ```
  kind create cluster --name {{cluster_name}}
  ```

- Delete one or more clusters:

  ```
  kind delete clusters {{cluster_name}}
  ```

- Get details about clusters, nodes, or the kubeconfig:

  ```
  kind get {{clusters|nodes|kubeconfig}}
  ```

- Export the kubeconfig or the logs:

  ```
  kind export {{kubeconfig|logs}}
  ```

kinit

Authenticate a principal with a Kerberos server to gain and cache a ticket.

Note: A Kerberos principal can be either a user, service, or application.

More information: https://web.mit.edu/kerberos/krb5-latest/doc/user/user_commands/kinit.html.

- Authenticate a user and obtain a ticket-granting ticket:

```
kinit {{username}}
```

- Renew a ticket-granting ticket:

```
kinit -R
```

- Specify a lifetime for the ticket:

```
kinit -l {{5h}}
```

- Specify a total renewable lifetime for the ticket:

```
kinit -r {{1w}}
```

- Specify a different principal name to authenticate as:

```
kinit -p {{principal@REALM}}
```

- Specify a different keytab file to authenticate with:

```
kinit -t {{path/to/keytab}}
```

kismet

A wireless network and device detector, sniffer, wardriving tool, and WIDS (wireless intrusion detection) framework.

More information: https://www.kismetwireless.net/docs/readme/starting/commandline/.

- Capture packets from a specific wireless interface:

```
sudo kismet -c {{wlan0}}
```

- Monitor multiple channels on a wireless interface:

```
sudo kismet -c {{wlan0,wlan1}} -m
```

- Capture packets and save them to a specific directory:

```
sudo kismet -c {{wlan0}} -d {{path/to/output}}
```

- Start Kismet with a specific configuration file:

```
sudo kismet -c {{wlan0}} {{[-f|--config-file]}} {{path/to/config.conf}}
```

- Monitor and log data to an SQLite database:

```
sudo kismet -c {{wlan0}} --log-to-db
```

- Monitor using a specific data source:

```
sudo kismet -c {{wlan0}} --data-source={{rtl433}}
```

- Enable alerts for specific events:

```
sudo kismet -c {{wlan0}} --enable-alert={{new_ap}}
```

- Display detailed information about a specific AP's packets:

```
sudo kismet -c {{wlan0}} --info {{BSSID}}
```

kite

This command is an alias of **`kiterunner`**.

- View documentation for the original command:

  ```
  tldr kiterunner
  ```

kiterunner brute

A contextual web scanner for bruteforcing API paths and web endpoints using wordlists.

The **`brute`** subcommand targets one or multiple hosts.

More information: https://github.com/assetnote/kiterunner.

- Bruteforce a target with an Assetnote wordlist (e.g., first 20,000 API routes):

```
kiterunner brute {{https://example.com}} {{[-A|--assetnote-wordlist]}} {{apiroutes-210328:20000}}
```

- Bruteforce a target with a custom wordlist:

```
kiterunner brute {{https://example.com}} {{[-w|--wordlist]}} {{path/to/wordlist.txt}}
```

- Bruteforce using a dirsearch-style wordlist with extension substitution:

```
kiterunner brute {{https://example.com}} {{[-w|--wordlist]}} {{path/to/dirsearch.txt}} {{[-D|--dirsearch-compat]}} {{[-e|--extensions]}} {{json,txt}}
```

- Bruteforce with specific file extensions appended and output in JSON format:

```
kiterunner brute {{https://example.com}} {{[-w|--wordlist]}} {{path/to/wordlist.txt}} {{[-e|--extensions]}} {{aspx,ashx}} {{[-o|--output]}} {{json}}
```

- Bruteforce a list of targets from a file with custom concurrency settings for performance:

```
kiterunner brute {{path/to/targets.txt}} {{[-w|--wordlist]}} {{path/to/wordlist.txt}} {{[-x|--max-connection-per-host]}} {{5}} {{[-j|--max-parallel-hosts]}} {{100}}
```

- Bruteforce and ignore specific content length responses:

```
kiterunner brute {{https://example.com}} {{[-w|--wordlist]}} {{path/to/wordlist.txt}} --ignore-length {{100-105}}
```

- Bruteforce with custom HTTP headers:

```
kiterunner brute {{https://example.com}} {{[-w|--wordlist]}}
{{path/to/wordlist.txt}} {{[-H|--header]}} "{{Authorization:
Bearer token}}"
```

- Bruteforce a list of targets from a file with fail status code filtering:

```
kiterunner brute {{path/to/targets.txt}} {{[-w|--wordlist]}}
{{path/to/wordlist.txt}} --fail-status-codes {{400,401,404}}
```

kiterunner kb

A contextual web scanner for manipulating kitebuilder schemas used in API and web endpoint discovery.

The **kb** subcommand handles schema compilation, conversion, parsing, and request replay.

More information: https://github.com/assetnote/kiterunner.

- Compile a kitebuilder schema from JSON to a kite file:

```
kiterunner kb compile {{path/to/wordlist.json}} {{path/to/wordlist.kite}}
```

- Convert a kite file to a text wordlist:

```
kiterunner kb convert {{path/to/wordlist.kite}} {{path/to/wordlist.txt}}
```

- Convert a text wordlist to a kite file:

```
kiterunner kb convert {{path/to/wordlist.txt}} {{path/to/wordlist.kite}}
```

- Convert a kite file to a JSON schema:

```
kiterunner kb convert {{path/to/wordlist.kite}} {{path/to/wordlist.json}}
```

- Parse a kitebuilder schema and output prettified JSON data:

```
kiterunner kb parse {{path/to/wordlist.json}} {{[-o|--output]}} {{json}}
```

- Parse a kite file and output prettified text data:

```
kiterunner kb parse {{path/to/wordlist.kite}} {{[-o|--output]}} {{text}}
```

- Replay a specific request from a kitebuilder schema output:

```
kiterunner kb replay {{[-w|--kitebuilder-list]}} {{path/to/wordlist.kite}} "{{request_output}}"
```

- Replay a request through a proxy for inspection:

```
kiterunner kb replay {{[-w|--kitebuilder-list]}} {{path/to/
wordlist.kite}} {{[-p|--proxy]}} {{http://localhost:8080}}
"{{request_output}}"
```

kiterunner scan

A contextual web scanner for concurrently scanning API paths and web endpoints using kitebuilder wordlists.

The **scan** subcommand targets one or multiple hosts with structured API requests.

More information: https://github.com/assetnote/kiterunner.

- Scan a target with an Assetnote wordlist (e.g., first 5000 API routes):

```
kiterunner scan {{https://example.com}} {{[-A|--assetnote-wordlist]}} {{apiroutes-210228:5000}}
```

- Scan a target with a kitebuilder wordlist:

```
kiterunner scan {{https://example.com}} {{[-w|--kitebuilder-list]}} {{path/to/wordlist.kite}}
```

- Scan multiple hosts from a file with a kitebuilder wordlist:

```
kiterunner scan {{path/to/hosts.txt}} {{[-w|--kitebuilder-list]}} {{path/to/wordlist.kite}}
```

- Scan with an Assetnote wordlist and JSON output:

```
kiterunner scan {{https://example.com}} {{[-A|--assetnote-wordlist]}} {{apiroutes-210228:5000}} -o {{json}}
```

- Scan with custom concurrency settings for performance:

```
kiterunner scan {{https://example.com}} {{[-w|--kitebuilder-list]}} {{path/to/wordlist.kite}} {{[-x|--max-connection-per-host]}} {{5}} {{[-j|--max-parallel-hosts]}} {{100}}
```

- Scan with a wordlist as a normal wordlist, disabling depth scanning:

```
kiterunner scan {{https://example.com}} {{[-w|--kitebuilder-list]}} {{path/to/rafter.txt}} {{[-d|--preflight-depth]}} {{0}}
```

- Scan with custom headers and ignore specific content length responses:

```
kiterunner scan {{https://example.com}} {{[-w|--kitebuilder-list]}} {{path/to/wordlist.kite}} {{[-H|--header]}} "{{Authorization: Bearer token}}" --ignore-length {{100-105}}
```

- Perform a full kitebuilder scan without phase scanning:

```
kiterunner scan {{https://example.com}} {{[-w|--kitebuilder-
list]}} {{path/to/wordlist.kite}} --kitebuilder-full-scan
```

kiterunner wordlist

A contextual web scanner for managing wordlists used in API and web endpoint discovery.

The **`wordlist`** subcommand handles listing and saving wordlists in **`~/.cache/kiterunner`**.

More information: https://github.com/assetnote/kiterunner.

- List all cached and available Assetnote wordlists:

```
kiterunner wordlist list
```

- List wordlists with JSON output:

```
kiterunner wordlist list {{[-o|--output]}} {{json}}
```

- List wordlists with verbose debug output:

```
kiterunner wordlist list {{[-v|--verbose]}} {{debug}}
```

- Save a specific Assetnote wordlist by alias:

```
kiterunner wordlist save {{apiroutes-210328}}
```

- Save a specific Assetnote wordlist by full filename:

```
kiterunner wordlist save {{path/to/
httparchive_apiroutes_2024_05_28.txt}}
```

- Save multiple wordlists by alias:

```
kiterunner wordlist save {{apiroutes-210328,aspx-210328}}
```

- Save a wordlist with quiet mode to suppress output:

```
kiterunner wordlist save {{apiroutes-210328}} {{[-q|--
quiet]}}
```

kiterunner

A contextual web scanner for discovering API paths and web endpoints using wordlists and kitebuilder schemas.

More information: https://github.com/assetnote/kiterunner.

- View documentation for bruteforcing API paths and web endpoints:

```
tldr kiterunner brute
```

- View documentation for concurrently scanning hosts with kitebuilder wordlists:

```
tldr kiterunner scan
```

- View documentation for manipulating kitebuilder schemas:

```
tldr kiterunner kb
```

- View documentation for managing cached and remote wordlists:

```
tldr kiterunner wordlist
```

kitex

Code generation tool provided by the Go RPC framework Kitex.

Kitex accepts both thrift and protobuf IDLs, and supports generating a skeleton of a server side project.

More information: https://www.cloudwego.io.

- Generate client codes when a project is in `$GOPATH`:

  ```
  kitex {{path/to/IDL_file.thrift}}
  ```

- Generate client codes when a project is not in `$GOPATH`:

  ```
  kitex -module {{github.com/xx-org/xx-name}} {{path/to/IDL_file.thrift}}
  ```

- Generate client codes with protobuf IDL:

  ```
  kitex -type protobuf {{path/to/IDL_file.proto}}
  ```

- Generate server codes:

  ```
  kitex -service {{svc_name}} {{path/to/IDL_file.thrift}}
  ```

kitty

A fast, feature-rich, GPU based terminal emulator.

More information: https://sw.kovidgoyal.net/kitty/.

- Open a new terminal:

```
kitty
```

- Open a terminal with the specified title for the window:

```
kitty --title "{{title}}"
```

- Start the theme-chooser builtin:

```
kitty +kitten themes
```

- Display an image in the terminal:

```
kitty +kitten icat {{path/to/image}}
```

- Copy the contents of `stdin` to the clipboard:

```
echo {{example}} | kitty +kitten clipboard
```

kiwi-ng

An OS image and appliance builder.

More information: https://osinside.github.io/kiwi/.

- Build an appliance:

```
kiwi-ng system build --description={{path/to/directory}} --target-dir={{path/to/directory}}
```

- Show build result of built appliance:

```
kiwi-ng result list --target-dir={{path/to/directory}}
```

- Display help:

```
kiwi-ng help
```

- Display version:

```
kiwi-ng -v
```

knife

Interact with a Chef server from a local Chef repo.

More information: https://docs.chef.io/knife.html.

- Bootstrap a new node:

```
knife bootstrap {{fqdn_or_ip}}
```

- List all registered nodes:

```
knife node list
```

- Show a node:

```
knife node show {{node_name}}
```

- Edit a node:

```
knife node edit {{node_name}}
```

- Edit a role:

```
knife role edit {{role_name}}
```

- View a data bag:

```
knife data bag show {{data_bag_name}} {{data_bag_item}}
```

- Upload a local cookbook to the Chef server:

```
knife cookbook upload {{cookbook_name}}
```

knotc

Control knot DNS server.

More information: https://www.knot-dns.cz/docs/latest/html/man_knotc.html.

- Start editing a zone:

```
knotc zone-begin {{zone}}
```

- Set an A record with TTL of 3600:

```
knotc zone-set {{zone}} {{subdomain}} 3600 A {{ip_address}}
```

- Finish editing the zone:

```
knotc zone-commit {{zone}}
```

- Get the current zone data:

```
knotc zone-read {{zone}}
```

- Get the current server configuration:

```
knotc conf-read server
```

kodi-send

Send actions to Kodi.

More information: https://kodi.wiki/view/List_of_built-in_functions.

- Quit Kodi:

```
kodi-send {{[-a|--action]}} Quit
```

- Reboot the system:

```
kodi-send {{[-a|--action]}} Reboot
```

- Send an action to a remote host:

```
kodi-send --host {{192.168.0.1}} --port {{9777}} {{[-a|--action]}} {{Quit}}
```

komac

Create WinGet manifests for the **`winget-pkgs`** repository.

More information: https://github.com/russellbanks/Komac.

- Create a new package from scratch:

```
komac new {{Package.Identifier}} {{[-v|--version]}} {{1.2.3}}
{{[-u|--urls]}} {{https://example.com/app.exe}}
```

- Update an existing package with a new version:

```
komac update {{Package.Identifier}} {{[-v|--version]}}
{{1.2.3}} {{[-u|--urls]}} {{https://example.com/app.exe}}
```

- Update a package with multiple URLs and automatically submit:

```
komac update {{Package.Identifier}} {{[-v|--version]}}
{{1.2.3}} {{[-u|--urls]}} {{https://example.com/app.exe
https://example.com/app.msi ...}} {{[-s|--submit]}}
```

- Remove a version from winget-pkgs:

```
komac remove {{Package.Identifier}} {{[-v|--version]}}
{{1.2.3}}
```

- List all versions for a package:

```
komac {{[list|list-versions]}} {{Package.Identifier}}
```

- Sync your fork of winget-pkgs with the upstream repository:

```
komac {{[sync|sync-fork]}}
```

- Update the stored GitHub token:

```
komac token {{[add|update]}} {{[-t|--token]}}
{{your_github_token}}
```

- Generate shell autocompletion script:

```
komac {{[complete|autocomplete]}} {{bash|elvish|fish|
powershell|zsh}}
```

kompose

Convert docker-compose applications to Kubernetes.

More information: https://github.com/kubernetes/kompose.

- Deploy a dockerized application to Kubernetes:

  ```
  kompose up {{[-f|--file]}} {{docker-compose.yml}}
  ```

- Delete instantiated services/deployments from Kubernetes:

  ```
  kompose down {{[-f|--file]}} {{docker-compose.yml}}
  ```

- Convert a docker-compose file into Kubernetes resources file:

  ```
  kompose convert {{[-f|--file]}} {{docker-compose.yml}}
  ```

kool

Build software development environments.

More information: https://kool.dev/docs/commands-reference/kool.

- Create a project using a specific preset:

  ```
  kool create {{preset}} {{project_name}}
  ```

- Run a specific script defined in the `kool.yml` file in the current directory:

  ```
  kool run {{script}}
  ```

- Start/stop services in the current directory:

  ```
  kool {{start|stop}}
  ```

- Display status of the services in the current directory:

  ```
  kool status
  ```

- Update to the latest version:

  ```
  kool self-update
  ```

- Print the completion script for the specified shell:

  ```
  kool completion {{bash|fish|powershell|zsh}}
  ```

kopia

Fast, secure open-source backup tool.

Supports encryption, compression, deduplication, and incremental snapshots.

More information: https://kopia.io/docs/reference/command-line/.

- Create a repository in the local filesystem:

```
kopia repository create filesystem --path {{path/to/
local_repository}}
```

- Create a repository on Amazon S3:

```
kopia repository create s3 --bucket {{bucket_name}} --access-
key {{AWS_access_key_id}} --secret-access-key
{{AWS_secret_access_key}}
```

- Connect to a repository:

```
kopia repository connect {{repository_type}} --path {{path/
to/repository}}
```

- Create a snapshot of a directory:

```
kopia snapshot create {{path/to/directory}}
```

- List snapshots:

```
kopia snapshot list
```

- Restore a snapshot to a specific directory:

```
kopia snapshot restore {{snapshot_id}} {{path/to/
target_directory}}
```

- Create a new policy:

```
kopia policy set --global --keep-latest
{{number_of_snapshots_to_keep}} --compression
{{compression_algorithm}}
```

- Ignore a specific file or folder from backups:

```
kopia policy set --global --add-ignore {{path/to/
file_or_folder}}
```

kops

Create, destroy, upgrade and maintain Kubernetes clusters.

More information: https://github.com/kubernetes/kops/.

- Create a cluster from the configuration specification:

```
kops create cluster {{[-f|--filename]}} {{cluster_name.yaml}}
```

- Create a new SSH public key:

```
kops create sshpublickey {{key_name}} {{[-i|--ssh-public-
key]}} {{~/.ssh/id_rsa.pub}}
```

- Export the cluster configuration to the `~/.kube/config` file:

```
kops export kubecfg {{cluster_name}}
```

- Get the cluster configuration as YAML:

```
kops get cluster {{cluster_name}} {{[-o|--output]}} yaml
```

- Delete a cluster:

```
kops delete cluster {{cluster_name}} {{[-y|--yes]}}
```

- Validate a cluster:

```
kops validate cluster {{cluster_name}} --wait
{{wait_time_until_ready}} --count
{{num_required_validations}}
```

kosmorro

Compute the ephemerides and the events for a date at a position on Earth.

More information: https://kosmorro.space.

- Get ephemerides for Paris, France:

```
kosmorro --latitude={{48.7996}} --longitude={{2.3511}}
```

- Get ephemerides for Paris, France, in the UTC+2 timezone:

```
kosmorro --latitude={{48.7996}} --longitude={{2.3511}} --
timezone={{2}}
```

- Get ephemerides for Paris, France, on June 9th, 2020:

```
kosmorro --latitude={{48.7996}} --longitude={{2.3511}} --
date={{2020-06-09}}
```

- Generate a PDF (Note: TeXLive must be installed):

```
kosmorro --format={{pdf}} --output={{path/to/file.pdf}}
```

kotlin

Kotlin application launcher.

More information: https://kotlinlang.org.

- Run a jar file:

```
kotlin {{filename.jar}}
```

- Display Kotlin and JVM version:

```
kotlin -version
```

kotlinc

Kotlin compiler.

More information: https://kotlinlang.org/docs/command-line.html.

- Start a REPL (interactive shell):

```
kotlinc
```

- Compile a Kotlin file:

```
kotlinc {{path/to/file.kt}}
```

- Compile several Kotlin files:

```
kotlinc {{path/to/file1.kt path/to/file2.kt ...}}
```

- Execute a specific Kotlin Script file:

```
kotlinc -script {{path/to/file.kts}}
```

- Compile a Kotlin file into a self contained jar file with the Kotlin runtime library included:

```
kotlinc {{path/to/file.kt}} -include-runtime -d {{path/to/file.jar}}
```

kr

This command is an alias of **`kiterunner`**.

- View documentation for the original command:

  ```
  tldr kiterunner
  ```

krita

A sketching and painting program designed for digital artists.

See also: **gimp**.

More information: https://docs.krita.org/en/reference_manual/linux_command_line.html.

- Start Krita:

```
krita
```

- Open specific files:

```
krita {{path/to/image1 path/to/image2 ...}}
```

- Start without a splash screen:

```
krita --nosplash
```

- Start with a specific workspace:

```
krita --workspace {{Animation}}
```

- Start in fullscreen mode:

```
krita --fullscreen
```

krunvm

Create MicroVMs from OCI images.

More information: https://github.com/containers/krunvm.

- Create MicroVM based on Fedora:

```
krunvm create {{docker.io/fedora}} --cpus {{number_of_vcpus}}
--mem {{memory_in_megabytes}} --name "{{name}}"
```

- Start a specific image:

```
krunvm start "{{image_name}}"
```

- List images:

```
krunvm list
```

- Change a specific image:

```
krunvm changevm --cpus {{number_of_vcpus}} --mem
{{memory_in_megabytes}} --name "{{new_vm_name}}"
"{{current_vm_name}}"
```

- Delete a specific image:

```
krunvm delete "{{image_name}}"
```

ksh

Korn Shell, a Bash-compatible command-line interpreter.

See also: `!`, `^`.

More information: https://manned.org/ksh.

- Start an interactive shell session:

```
ksh
```

- Execute specific [c]ommands:

```
ksh -c "{{echo 'ksh is executed'}}"
```

- Execute a specific script:

```
ksh {{path/to/script.ksh}}
```

- Check a specific script for syntax errors without executing it:

```
ksh -n {{path/to/script.ksh}}
```

- E[x]ecute a specific script, printing each command in the script before executing it:

```
ksh -x {{path/to/script.ksh}}
```

kube-capacity

Provide an overview of resource requests, limits, and utilization in a Kubernetes cluster.

Combine the best parts of **`kubectl top`** and **`kubectl describe`** into a CLI focused on cluster resources.

More information: https://github.com/robscott/kube-capacity.

- List nodes including the total CPU and Memory resource requests and limits:

  ```
  kube-capacity
  ```

- Include pods:

  ```
  kube-capacity {{[-p|--pods]}}
  ```

- Include utilization:

  ```
  kube-capacity {{[-u|--util]}}
  ```

kube-fzf

Shell commands for command-line fuzzy searching of Kubernetes Pods.

See also: **`kubectl`** for related commands.

More information: https://github.com/thecasualcoder/kube-fzf.

- Get pod details (from current namespace):

```
findpod
```

- Get pod details (from all namespaces):

```
findpod -a
```

- Describe a pod:

```
describepod
```

- Tail pod logs:

```
tailpod
```

- Exec into a pod's container:

```
execpod {{shell_command}}
```

- Port-forward a pod:

```
pfpod {{port_number}}
```

kubeadm

Interface for creating and managing Kubernetes clusters.

More information: https://kubernetes.io/docs/reference/setup-tools/kubeadm.

- Create a Kubernetes control plane:

  ```
  kubeadm init
  ```

- Bootstrap a Kubernetes worker node and join it to a cluster:

  ```
  kubeadm join --token {{token}}
  ```

- Create a new bootstrap token with a TTL of 12 hours:

  ```
  kubeadm token create --ttl {{12h0m0s}}
  ```

- Check if the Kubernetes cluster is upgradeable and which versions are available:

  ```
  kubeadm upgrade plan
  ```

- Upgrade Kubernetes cluster to a specified version:

  ```
  kubeadm upgrade apply {{version}}
  ```

- View the kubeadm ConfigMap containing the cluster's configuration:

  ```
  kubeadm config view
  ```

- Revert changes made to the host by 'kubeadm init' or 'kubeadm join':

  ```
  kubeadm reset
  ```

kubectl annotate

Annotates Kubernetes resources.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_annotate.

- Annotate a pod:

```
kubectl annotate {{[po|pods]}} {{pod_name}} {{key}}={{value}}
```

- Update a pod annotation by overwriting the existing value:

```
kubectl annotate {{[po|pods]}} {{pod_name}} {{key}}={{value}} --overwrite
```

- Annotate all pods in a namespace with a specific label selector:

```
kubectl annotate {{[po|pods]}} {{key}}={{value}} {{[-l|--selector]}} {{label_key}}={{label_value}}
```

- List all annotations a pod has:

```
kubectl annotate {{[po|pods]}} {{pod_name}} --list
```

- Remove the annotation from a pod:

```
kubectl annotate {{[po|pods]}} {{pod_name}} {{key}}-
```

kubectl api-resources

Print the supported API resources on the server.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_api-resources.

- Print the supported API resources:

```
kubectl api-resources
```

- Print the supported API resources with more information:

```
kubectl api-resources {{[-o|--output]}} wide
```

- Print the supported API resources sorted by a column:

```
kubectl api-resources --sort-by {{name}}
```

- Print the supported namespaced resources:

```
kubectl api-resources --namespaced
```

- Print the supported non-namespaced resources:

```
kubectl api-resources --namespaced=false
```

- Print the supported API resources with a specific API group:

```
kubectl api-resources --api-group {{api_group}}
```

kubectl api-versions

Print the supported API versions on the server, in the form of "group/version".

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_api-versions.

- Print the supported API versions on the server, in the form of "group/version":

```
kubectl api-versions
```

kubectl apply

Manage applications through files defining Kubernetes resources.

Create and update resources in a cluster.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_apply.

- Apply a configuration to a resource by file name:

  ```
  kubectl apply {{[-f|--filename]}} {{path/to/file}}
  ```

- Apply a configuration to a resource from `kustomization.yaml` in a directory:

  ```
  kubectl apply {{[-k|--kustomize]}} {{path/to/directory}}
  ```

- Apply a configuration to a resource by `stdin`:

  ```
  {{cat pod.json}} | kubectl apply {{[-f|--filename]}} -
  ```

- Edit the latest last-applied-configuration annotations of resources from the default editor:

  ```
  kubectl apply edit-last-applied {{[-f|--filename]}} {{path/to/file}}
  ```

- Set the latest last-applied-configuration annotations by setting it to match the contents of a file:

  ```
  kubectl apply set-last-applied {{[-f|--filename]}} {{path/to/file}}
  ```

- View the latest last-applied-configuration annotations by type/name or file:

  ```
  kubectl apply view-last-applied {{[-f|--filename]}} {{path/to/file}}
  ```

kubectl attach

Attach to a process that is already running inside an existing container.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_attach/.

- Get output from a pod:

```
kubectl attach {{pod_name}}
```

- Get output from a container in the specified pod:

```
kubectl attach {{pod_name}} {{[-c|--container]}} {{container_name}}
```

- Get output from the first pod of a replica set:

```
kubectl attach {{[rs|replicaset]}}/{{replicaset_name}}
```

- Create an interactive session with a pod:

```
kubectl attach {{pod_name}} {{[-it|--stdin --tty]}}
```

- Create an interactive session with a specific container from a pod:

```
kubectl attach {{pod_name}} {{[-c|--container]}} {{container_name}} {{[-it|--stdin --tty]}}
```

kubectl auth

Inspect access permissions in a Kubernetes cluster.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_auth.

- Check if the current user can perform all actions on all resources in a specific namespace:

```
kubectl auth can-i '*' '*' {{[-n|--namespace]}} {{namespace}}
```

- Check if the current user can perform a specific verb on a specific resource:

```
kubectl auth can-i {{verb}} {{resource}} {{[-n|--namespace]}} {{namespace}}
```

- Check if a specific user or service account can perform an action on a resource:

```
kubectl auth can-i {{verb}} {{resource}} {{[-n|--namespace]}} {{namespace}} --as {{user_or_sa}}
```

- List all actions the current user is allowed to perform in a namespace:

```
kubectl auth can-i --list {{[-n|--namespace]}} {{namespace}}
```

kubectl autoscale

Create an autoscaler to intelligently scale pod count based on kubernetes cluster demands.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_autoscale.

- Auto scale a deployment with no target CPU utilization specified:

```
kubectl autoscale {{[deploy|deployment]}} {{deployment_name}} --min {{min_replicas}} --max {{max_replicas}}
```

- Auto scale a deployment with target CPU utilization:

```
kubectl autoscale {{[deploy|deployment]}} {{deployment_name}} --max {{max_replicas}} --cpu-percent {{target_cpu}}
```

kubectl certificate

Manage certificate signing requests.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_certificate.

- Approve a certificate signing request by name:

```
kubectl certificate approve {{csr_name}}
```

- Deny a certificate signing request by name:

```
kubectl certificate deny {{csr_name}}
```

- Approve a certificate signing request defined in a manifest file:

```
kubectl certificate approve --filename {{path/to/csr.yaml}}
```

- Deny a certificate signing request defined in a manifest file:

```
kubectl certificate deny --filename {{path/to/csr.yaml}}
```

- Force reapproval of a certificate signing request that was previously denied:

```
kubectl certificate approve --force {{csr_name}}
```

kubectl cluster-info

Display endpoint information about the Kubernetes master and services in the cluster.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_cluster-info/.

- Show basic cluster information:

```
kubectl cluster-info
```

- Dump current cluster state to `stdout` (for debugging):

```
kubectl cluster-info dump
```

- Dump cluster state to a directory:

```
kubectl cluster-info dump --output-directory {{path/to/directory}}
```

- Use a specific kubeconfig context:

```
kubectl cluster-info --context {{context_name}}
```

kubectl config

Manage Kubernetes configuration (kubeconfig) files for accessing clusters via **`kubectl`** or the Kubernetes API.

By default, the Kubernetes will get its configuration from **`${HOME}/.kube/config`**.

See also: **`kubectx`**, **`kubens`**.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_config.

- Get all contexts in the default kubeconfig file:

```
kubectl config get-contexts
```

- Get all clusters/contexts/users in a custom kubeconfig file:

```
kubectl config {{get-clusters|get-contexts|get-users}} --
kubeconfig {{path/to/kubeconfig.yaml}}
```

- Get the current context:

```
kubectl config current-context
```

- Set the default namespace of the current context:

```
kubectl config set-context --current --namespace
{{namespace}}
```

- Switch to another context:

```
kubectl config {{use|use-context}} {{context_name}}
```

- Delete clusters/contexts/users:

```
kubectl config {{delete-cluster|delete-context|delete-user}}
{{cluster|context|user}}
```

- Permanently add custom kubeconfig files:

```
export KUBECONFIG="{{$HOME.kube/config:path/to/
custom_kubeconfig.yaml}}" kubectl config get-contexts
```

kubectl create

Create a resource from a file or from **`stdin`**.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_create.

- Create a resource using the resource definition file:

```
kubectl create {{[-f|--filename]}} {{path/to/file.yml}}
```

- Create a resource from `stdin`:

```
kubectl create {{[-f|--filename]}} -
```

- Create a deployment:

```
kubectl create {{[deploy|deployment]}} {{deployment_name}} --image {{image}}
```

- Create a deployment with replicas:

```
kubectl create {{[deploy|deployment]}} {{deployment_name}} --image {{image}} --replicas {{number_of_replicas}}
```

- Create a service:

```
kubectl create {{[svc|service]}} {{service_type}} {{service_name}} --tcp {{port}}:{{target_port}}
```

- Create a namespace:

```
kubectl create {{[ns|namespace]}} {{namespace_name}}
```

kubectl debug

Debug cluster resources using interactive debugging containers.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_debug.

- Create an interactive debugging session in a pod and immediately attach to it:

```
kubectl debug {{pod_name}} {{[-it|--stdin --tty]}} --image busybox
```

- Create a debug container with a custom image and name:

```
kubectl debug --image {{image}} {{[-c|--container]}} {{container_name}} {{pod_name}}
```

- Create an interactive debugging session on a node and immediately attach to it (the container will run in the host namespaces and the host's filesystem will be mounted at `/host`):

```
kubectl debug node/{{node_name}} {{[-it|--stdin --tty]}} --image busybox
```

- Create a copy of a pod and add a debug container to it:

```
kubectl debug {{pod_name}} {{[-it|--stdin --tty]}} --image {{image}} --copy-to {{pod_copy_name}}
```

- Create a copy of a pod and change the command of a specific container:

```
kubectl debug {{pod_name}} {{[-it|--stdin --tty]}} --copy-to {{pod_copy_name}} --container {{container_name}} -- {{command}}
```

- Create a copy of a pod and change the image of a specific container:

```
kubectl debug {{pod_name}} --copy-to {{pod_copy_name}} --set-image {{container_name}}={{image}}
```

- Create a copy of a pod and change all container images:

```
kubectl debug {{pod_name}} --copy-to {{pod_copy_name}} --set-image '*={{image}}'
```

- Create an ephemeral debug container and target a specific container (useful for debugging distroless containers):

```
kubectl debug {{pod_name}} {{[-it|--stdin --tty]}} --image
{{image}} --target {{target_container_name}}
```

kubectl delete

Delete Kubernetes resources.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_delete.

- Delete a specific pod:

```
kubectl delete {{[po|pods]}} {{pod_name}}
```

- Delete a specific deployment:

```
kubectl delete {{[deploy|deployments]}} {{deployment_name}}
```

- Delete a specific node:

```
kubectl delete {{[no|nodes]}} {{node_name}}
```

- Delete all pods in a specified namespace:

```
kubectl delete {{[po|pods]}} --all {{[-n|--namespace]}} {{namespace}}
```

- Delete all deployments and services in a specified namespace:

```
kubectl delete {{[deploy|deployments]}},{{[svc|services]}} --all {{[-n|--namespace]}} {{namespace}}
```

- Delete all nodes:

```
kubectl delete {{[no|nodes]}} --all
```

- Delete resources defined in a YAML manifest:

```
kubectl delete {{[-f|--filename]}} {{path/to/manifest.yaml}}
```

kubectl describe

Show details of Kubernetes resources.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_describe.

- Show details of pods in a namespace:

```
kubectl describe {{[po|pods]}} {{[-n|--namespace]}} {{namespace}}
```

- Show details of nodes in a namespace:

```
kubectl describe {{[no|nodes]}} {{[-n|--namespace]}} {{namespace}}
```

- Show the details of a specific pod in a namespace:

```
kubectl describe {{[po|pods]}} {{pod_name}} {{[-n|--namespace]}} {{namespace}}
```

- Show the details of a specific node in a namespace:

```
kubectl describe {{[no|nodes]}} {{node_name}} {{[-n|--namespace]}} {{namespace}}
```

- Show details of Kubernetes objects defined in a YAML manifest file:

```
kubectl describe {{[-f|--filename]}} {{path/to/manifest.yaml}}
```

kubectl diff

Check differences between live resources and files.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_diff.

- View differences between the live resource and file definition:

```
kubectl diff {{[-f|--filename]}} {{path/to/filename}}
```

kubectl edit

Edit Kubernetes resources.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_edit.

- Edit a pod in the default namespace:

```
kubectl edit {{[po|pods]}}/{{pod_name}}
```

- Edit a deployment in the default namespace:

```
kubectl edit {{[deploy|deployment]}}/{{deployment_name}}
```

- Edit a service in the default namespace:

```
kubectl edit {{[svc|service]}}/{{service_name}}
```

- Edit all entries of a given resource in a given namespace:

```
kubectl edit {{resource}} {{[-n|--namespace]}} {{namespace}}
```

- Edit a resource using a specific editor:

```
KUBE_EDITOR={{nano}} kubectl edit {{resource}}/
{{resource_name}}
```

- Edit a resource in JSON format:

```
kubectl edit {{resource}}/{{resource_name}} {{[-o|--output]}}
json
```

kubectl events

Display resource events.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_events/.

- Show events in the default namespace:

```
kubectl events
```

- Show events in all namespaces:

```
kubectl events {{[-A|--all-namespaces]}}
```

- Watch events in a specific namespace:

```
kubectl events {{[-w|--watch]}} {{[-n|--namespace]}} {{namespace}}
```

- Show events for a pod in a specific namespace:

```
kubectl events --for {{pod}}/{{pod_name}} {{[-n|--namespace]}} {{namespace}}
```

- Show events for a resource in a specific namespace:

```
kubectl events --for {{resource}}/{{resource_name}} {{[-n|--namespace]}} {{namespace}}
```

- Show events for type `Warning` or `Normal`:

```
kubectl events --types Warning,Normal
```

- Output events in YAML format:

```
kubectl events {{[-o|--output]}} yaml
```

kubectl exec

Execute a command in a container.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_exec.

- Open Bash in a pod, using the first container by default:

```
kubectl exec {{pod_name}} {{[-it|--stdin --tty]}} -- bash
```

kubectl explain

Display the documentation of a Kubernetes API resource, including available fields and descriptions.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_explain/.

- Show documentation for a resource:

```
kubectl explain {{pods|nodes|deployments|...}}
```

- Show documentation for a sub-resource or field of an object:

```
kubectl explain {{pod.spec.containers}}
```

- Show documentation for a specific versioned resource:

```
kubectl explain {{ingress.v1.networking.k8s.io}}
```

- Show all fields recursively for a resource:

```
kubectl explain {{po|pods}} --recursive
```

- Display help:

```
kubectl explain --help
```

kubectl expose

Expose a resource as a new Kubernetes service.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_expose.

- Create a service for a resource, which will be served from container port to node port:

```
kubectl expose {{resource_type}} {{resource_name}} --port {{node_port}} --target-port {{container_port}}
```

- Create a service for a resource identified by a file:

```
kubectl expose {{[-f|--filename]}} {{path/to/file.yml}} --port {{node_port}} --target-port {{container_port}}
```

- Create a service with a name, to serve to a node port which will be same for container port:

```
kubectl expose {{resource_type}} {{resource_name}} --port {{node_port}} --name {{service_name}}
```

kubectl get

Get Kubernetes objects and resources.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_get.

- Get all namespaces in the current cluster:

```
kubectl get {{[ns|namespaces]}}
```

- Get nodes in a specified namespace:

```
kubectl get {{[no|nodes]}} {{[-n|--namespace]}} {{namespace}}
```

- Get pods in a specified namespace:

```
kubectl get {{[po|pods]}} {{[-n|--namespace]}} {{namespace}}
```

- Get deployments in a specified namespace:

```
kubectl get {{[deploy|deployments]}} {{[-n|--namespace]}} {{namespace}}
```

- Get services in a specified namespace:

```
kubectl get {{[svc|services]}} {{[-n|--namespace]}} {{namespace}}
```

- Get other resources:

```
kubectl get {{persistentvolumeclaims|secret|...}}
```

- Get all resources in all namespaces:

```
kubectl get all {{[-A|--all-namespaces]}}
```

- Get Kubernetes objects defined in a YAML manifest file:

```
kubectl get {{[-f|--filename]}} {{path/to/manifest.yaml}}
```

kubectl kustomize

Build a set of Kubernetes resources using a **`kustomization.yaml`** file.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_kustomize/.

- Build resources from the current directory:

```
kubectl kustomize
```

- Build resources from a specific directory:

```
kubectl kustomize {{path/to/directory}}
```

- Build resources from a remote URL:

```
kubectl kustomize {{https://github.com/user/repo/path}}
```

- Build resources and save to a file:

```
kubectl kustomize {{path/to/directory}} > {{output.yaml}}
```

- Build resources with load restrictor disabled:

```
kubectl kustomize --load-restrictor {{LoadRestrictionsNone}} {{path/to/directory}}
```

kubectl label

Label Kubernetes resources.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_label.

- Label a pod:

```
kubectl label {{[po|pods]}} {{pod_name}} {{key}}={{value}}
```

- Update a pod label by overwriting the existing value:

```
kubectl label {{[po|pods]}} {{pod_name}} {{key}}={{value}} --overwrite
```

- Label all pods in the namespace:

```
kubectl label {{[po|pods]}} {{key}}={{value}} --all
```

- Label a pod identified by the pod definition file:

```
kubectl label {{[-f|--filename]}} {{pod_definition_file}} {{key}}={{value}}
```

- Remove the label from a pod:

```
kubectl label {{[po|pods]}} {{pod_name}} {{key}}-
```

kubectl logs

Show logs for containers in a pod.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_logs.

- Show logs for a single-container pod:

```
kubectl logs {{pod_name}}
```

- Show logs for a specified container in a pod:

```
kubectl logs {{[-c|--container]}} {{container_name}} {{pod_name}}
```

- Show logs for all containers in a pod:

```
kubectl logs --all-containers={{true}} {{pod_name}}
```

- Stream pod logs:

```
kubectl logs {{[-f|--follow]}} {{pod_name}}
```

- Show pod logs newer than a relative time like `10s`, `5m`, or `1h`:

```
kubectl logs --since {{relative_time}} {{pod_name}}
```

- Show the 10 most recent logs in a pod:

```
kubectl logs --tail {{10}} {{pod_name}}
```

- Show all pod logs for a given deployment:

```
kubectl logs {{[deploy|deployment]}}/{{deployment_name}}
```

kubectl patch

Patch Kubernetes resources with new values.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_patch.

- Partially patch a secret using a strategic merge JSON patch to remove the finalizer:

```
kubectl patch secrets {{secret_name}} {{[-p|--patch]}} '{"metadata":{"finalizers": []\}\}' --type merge
```

- Partially patch a secret using a strategic merge YAML patch to remove the finalizer:

```
kubectl patch secrets {{secret_name}} {{[-p|--patch]}} $'metadata:\n finalizers: []' --type merge
```

- Partially patch a pod's container using a JSON patch with positional arrays:

```
kubectl patch {{[po|pods]}} {{pod_name}} --type 'json' {{[-p|--patch]}} '[{"op": "replace", "path": "/spec/containers/0/image", "value":"{{new_image_value}}"}]'
```

- Update a deployment's replicas through the scale subresource using a strategic merge JSON patch:

```
kubectl patch {{[deploy|deployments]}} {{deployment_name}} --subresource 'scale' --type 'merge' {{[-p|--patch]}} '{"spec":{"replicas":{{number_of_replicas}}\}\}'
```

kubectl port-forward

Forward one or more local ports to a pod.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_port-forward/.

- Forward local ports 5000 and 6000 to the pod ports 5000 and 6000:

```
kubectl port-forward {{[po|pods]}}/{{pod_name}} 5000 6000
```

- Forward a random local port to the pod port 5000:

```
kubectl port-forward {{[po|pods]}}/{{pod_name}} :5000
```

- Forward local ports 5000 and 6000 to the deployment ports 5000 and 6000:

```
kubectl port-forward {{[deploy|deployment]}}/
{{deployment_name}} 5000 6000
```

- Forward local port 8443 to the service port named https:

```
kubectl port-forward {{[svc|service]}}/{{service_name}}
8443:https
```

- Forward port 8888 on all addresses to the pod port 5000:

```
kubectl port-forward {{[po|pods]}}/{{pod_name}} 8888:5000 --
address 0.0.0.0
```

- Forward port 8888 on localhost and selected IP to the pod port 5000:

```
kubectl port-forward {{[po|pods]}}/{{pod_name}} 8888:5000 --
address localhost,{{10.19.21.23}}
```

kubectl proxy

Create a proxy server or application-level gateway between localhost and the Kubernetes API server.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_proxy/.

- Run a proxy using the default settings on port 8001 and listen on localhost:

```
kubectl proxy
```

- Proxy part of the Kubernetes API while serving static files from a local directory:

```
kubectl proxy {{[-w|--www]}} {{path/to/static_dir}} {{[-P|--www-prefix]}} {{/static_prefix/}} --api-prefix {{/api_subset/}}
```

- Proxy the entire Kubernetes API under a custom prefix:

```
kubectl proxy --api-prefix {{/custom_prefix/}}
```

- Serve the Kubernetes API on a specific port while also serving static content:

```
kubectl proxy {{[-p|--port]}} {{port}} {{[-w|--www]}} {{path/to/static_dir}}
```

- Run a proxy on a random local port, printing the chosen port to `stdout`:

```
kubectl proxy {{[-p|--port]}} 0
```

- Run the proxy on a Unix domain socket instead of a TCP port:

```
kubectl proxy {{[-u|--unix-socket]}} {{path/to/socket}}
```

- Accept connections from remote hosts by listening on all interfaces (use caution when exposing the proxy publicly):

```
kubectl proxy --address 0.0.0.0 --accept-hosts '.*'
```

- Allow only selected API paths while rejecting sensitive endpoints:

```
kubectl proxy --accept-paths '^/api/v1/namespaces/default/.*' --reject-paths '^/api/.*/pods/.*/exec'
```

kubectl replace

Replace a resource by file or **`stdin`**.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_replace.

- Replace the resource using the resource definition file:

```
kubectl replace {{[-f|--filename]}} {{path/to/file.yml}}
```

- Replace the resource using the input passed into `stdin`:

```
kubectl replace {{[-f|--filename]}} -
```

- Force replace, delete and then re-create the resource:

```
kubectl replace {{[-f|--filename]}} {{path/to/file.yml}} --force
```

kubectl rollout

Manage the rollout of a Kubernetes resource (deployments, daemonsets, and statefulsets).

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_rollout.

- Start a rolling restart of a resource:

```
kubectl rollout restart {{resource_type}}/{{resource_name}}
```

- Watch the rolling update status of a resource:

```
kubectl rollout status {{resource_type}}/{{resource_name}}
```

- Roll back a resource to the previous revision:

```
kubectl rollout undo {{resource_type}}/{{resource_name}}
```

- View the rollout history of a resource:

```
kubectl rollout history {{resource_type}}/{{resource_name}}
```

kubectl run

Run pods in Kubernetes. Specifies pod generator to avoid deprecation error in some K8S versions.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_run.

- Run an nginx pod and expose port 80:

```
kubectl run {{nginx-dev}} --image nginx --port 80
```

- Run an nginx pod, setting the TEST_VAR environment variable:

```
kubectl run {{nginx-dev}} --image nginx --env "{{TEST_VAR}}
={{testing}}"
```

- Show API calls that would be made to create an nginx container:

```
kubectl run {{nginx-dev}} --image nginx --dry-run={{none|
server|client}}
```

- Run an Ubuntu pod interactively, never restart it, and remove it when it exits:

```
kubectl run {{temp-ubuntu}} --image ubuntu:22.04 --restart
Never --rm -- /bin/bash
```

- Run an Ubuntu pod, overriding the default command with echo, and specifying custom arguments:

```
kubectl run {{temp-ubuntu}} --image ubuntu:22.04 --command --
echo {{argument1 argument2 ...}}
```

kubectl scale

Set a new size for a deployment, replica set, replication controller, or stateful set.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_scale.

- Scale a replica set:

```
kubectl scale --replicas {{replicas_count}} rs/
{{replica_name}}
```

- Scale a resource identified by a file:

```
kubectl scale --replicas {{replicas_count}} {{[-f|--
filename]}} {{path/to/file.yml}}
```

- Scale a deployment based on current number of replicas:

```
kubectl scale --replicas {{replicas_count}} --current-
replicas {{current_replicas}} {{[deploy|deployment]}}/
{{deployment_name}}
```

kubectl taint

Update the taints on nodes.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_taint.

- Apply taint to a node:

```
kubectl taint {{[no|nodes]}} {{node_name}} {{label_key}}
={{label_value}}:{{effect}}
```

- Remove taint from a node:

```
kubectl taint {{[no|nodes]}} {{node_name}} {{label_key}}:
{{effect}}-
```

- Remove all taints from a node:

```
kubectl taint {{[no|nodes]}} {{node_name}} {{label_key}}-
```

kubectl top

See the resource consumption for nodes or pods.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_top.

- Get the resource consumption of all nodes:

```
kubectl top {{[no|nodes]}}
```

- Get resource consumption of a specific node:

```
kubectl top {{[no|nodes]}} {{node_name}}
```

- Get resource consumption of all pods:

```
kubectl top {{[po|pods]}}
```

- Get resource consumption of a specific pod:

```
kubectl top {{[po|pods]}} {{pod_name}}
```

- Get resource consumption of all pods in a namespace:

```
kubectl top {{[po|pods]}} {{[-n|--namespace]}} {{namespace_name}}
```

- Get resource consumption of all containers in a pod:

```
kubectl top {{[po|pods]}} --containers
```

- Get resource consumption of all pods with the specified label:

```
kubectl top {{[po|pods]}} {{[-l|--selector]}} {{key=value}}
```

kubectl version

Print the client and server version information for the current context.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_version.

- Print the client and server version information for the current context:

```
kubectl version
```

- Print the client version only for the current context:

```
kubectl version --client
```

- Output the client and server version information for the current context in JSON:

```
kubectl version {{[-o|--output]}} json
```

- Output the client and server version information for the current context in YAML:

```
kubectl version {{[-o|--output]}} yaml
```

kubectl wait

Wait for resource(s) to reach a certain state.

More information: https://kubernetes.io/docs/reference/kubectl/generated/kubectl_wait.

- Wait for a deployment to become available:

```
kubectl wait --for condition=available deployment/{{deployment_name}}
```

- Wait for all pods with a certain [l]abel to be ready:

```
kubectl wait --for condition=ready pod {{[-l|--selector]}} {{label_key}}={{label_value}}
```

- Wait for a pod to be deleted:

```
kubectl wait --for delete pod {{pod_name}}
```

- Wait for a job to complete, within 120 seconds (if the condition isn't met on time, the exit status will be unsuccessful):

```
kubectl wait --for condition=complete job/{{job_name}} --timeout 120s
```

kubectl

Run commands against Kubernetes clusters.

Some subcommands such as **`run`** have their own usage documentation.

More information: https://kubernetes.io/docs/reference/kubectl/.

- List information about a resource with more details:

```
kubectl get {{pod|service|deployment|ingress|...}} {{[-o|--output]}} wide
```

- Update specified pod with the label `unhealthy` and the value `true`:

```
kubectl label pods {{name}} unhealthy=true
```

- List all resources with different types:

```
kubectl get all
```

- Display resource (CPU/Memory/Storage) usage of nodes or pods:

```
kubectl top {{pods|node}}
```

- Print the address of the master and cluster services:

```
kubectl cluster-info
```

- Display an explanation of a specific field:

```
kubectl explain {{pods.spec.containers}}
```

- Print the logs for a container in a pod or specified resource:

```
kubectl logs {{pod_name}}
```

- Run command in an existing pod:

```
kubectl exec {{pod_name}} -- {{ls /}}
```

kubectx

Utility to manage and switch between **`kubectl`** contexts.

More information: https://manned.org/kubectx.

- List the contexts:

```
kubectx
```

- Switch to a named context:

```
kubectx {{name}}
```

- Switch to the previous context:

```
kubectx -
```

- Rename a named context:

```
kubectx {{alias}}={{name}}
```

- Show the current named context:

```
kubectx {{[-c|--current]}}
```

- Delete a named context:

```
kubectx -d {{name}}
```

kubens

Utility to switch between Kubernetes namespaces.

More information: https://manned.org/kubens.

- List the namespaces:

```
kubens
```

- Change the active namespace:

```
kubens {{name}}
```

- Switch to the previous namespace:

```
kubens -
```

kubeseal

Client-side utility for encrypting Kubernetes secrets using the Bitnami Sealed Secrets controller.

Creates SealedSecret resources that can be safely stored in version control.

Requires a controller running in the cluster (e.g., installed via **`kubectl apply --filename controller.yaml`**).

More information: https://github.com/bitnami-labs/sealed-secrets.

- Encrypt a Kubernetes secret from a YAML file into a SealedSecret (default JSON output):

```
kubeseal < {{secret.yaml}} > {{sealedsecret.json}}
```

- Encrypt a secret, outputting it in YAML or JSON format, using a bearer token for API authentication:

```
kubeseal < {{secret.yaml}} {{[-o|--format]}} {{yaml|json}} --token {{my-bearer-token}} > {{sealedsecret.yaml}}
```

- Seal a secret using a specific controller namespace of sealed-secrets controller and name:

```
kubeseal < {{secret.yaml}} --controller-namespace {{controller-namespace}} --controller-name {{controller-name}} > {{sealedsecret.yaml}}
```

- Encrypt a raw secret value from a file with a specified name and scope:

```
kubeseal --raw --from-file {{path/to/secret.txt}} --name {{my-secret}} --scope {{strict|namespace-wide|cluster-wide}} > {{sealedsecret.yaml}}
```

- Fetch the controller's public certificate for offline sealing with basic auth:

```
kubeseal --fetch-cert --username {{username}} --password {{password}} > {{cert.pem}}
```

- Seal a secret offline using a fetched certificate:

```
kubeseal < {{secret.yaml}} --cert {{cert.pem}} > {{sealedsecret.yaml}}
```

- Merge a secret into an existing SealedSecret file in-place:

```
kubeseal < {{secret.yaml}} --merge-into {{sealedsecret.yaml}}
```

- Validate a SealedSecret without applying it:

```
kubeseal < {{sealedsecret.yaml}} --validate
```

kubetail

Utility to tail multiple Kubernetes pod logs at the same time.

More information: https://github.com/johanhaleby/kubetail.

- Tail the logs of multiple pods (whose name starts with "my_app") in one go:

```
kubetail {{my_app}}
```

- Tail only a specific container from multiple pods:

```
kubetail {{my_app}} {{[-c|--container]}} {{my_container}}
```

- To tail multiple containers from multiple pods:

```
kubetail {{my_app}} {{[-c|--container]}} {{my_container_1}}
{{[-c|--container]}} {{my_container_2}}
```

- To tail multiple applications at the same time separate them by comma:

```
kubetail {{my_app_1}},{{my_app_2}}
```

kubie

Utility to switch between **`kubectl`** contexts and namespaces.

More information: https://github.com/sbstp/kubie.

- Display a selectable menu of contexts:

```
kubie ctx
```

- Switch current shell to the given context:

```
kubie ctx {{context}}
```

- Switch current shell to the given namespace:

```
kubie ns {{namespace}}
```

- Switch current shell to the given context and namespace:

```
kubie ctx {{context}} -n {{namespace}}
```

- Execute a command in the given context and namespace, without spawning a shell:

```
kubie exec {{context}} {{namespace}} {{command}}
```

- Check the Kubernetes configuration files for issues:

```
kubie lint
```

kustomize

Easily deploy resources for Kubernetes.

More information: https://github.com/kubernetes-sigs/kustomize/blob/master/site/content/en/docs/Reference/CLI/_index.md.

- Create a kustomization file with resources and namespace:

```
kustomize create --resources {{deployment.yaml,service.yaml}} --namespace {{staging}}
```

- Build a kustomization file and deploy it with `kubectl`:

```
kustomize build . | kubectl apply {{[-f|--filename]}} -
```

- Set an image in the kustomization file:

```
kustomize edit set image {{busybox=alpine:3.6}}
```

- Search for Kubernetes resources in the current directory to be added to the kustomization file:

```
kustomize create --autodetect
```

l2ping

Send an L2CAP echo request and receive an answer.

More information: https://manned.org/l2ping.

- Ping a Bluetooth device:

```
sudo l2ping {{mac_address}}
```

- Reverse ping a Bluetooth device:

```
sudo l2ping -r {{mac_address}}
```

- Ping a Bluetooth device from a specified interface:

```
sudo l2ping -i {{hci0}} {{mac_address}}
```

- Ping Bluetooth device with a specified sized data package:

```
sudo l2ping -s {{byte_count}} {{mac_address}}
```

- Ping flood a Bluetooth device:

```
sudo l2ping -f {{mac_address}}
```

- Ping a Bluetooth device a specified amount of times:

```
sudo l2ping -c {{amount}} {{mac_address}}
```

- Ping a Bluetooth device with a specified delay between requests:

```
sudo l2ping -d {{seconds}} {{mac_address}}
```

lambo new

A super-powered **`laravel new`** for Laravel and Valet.

More information: https://github.com/tighten/lambo.

- Create a new Laravel application:

```
lambo new {{app_name}}
```

- Install the application in a specific path:

```
lambo new --path={{path/to/directory}} {{app_name}}
```

- Include authentication scaffolding:

```
lambo new --auth {{app_name}}
```

- Include a specific frontend:

```
lambo new --{{vue|bootstrap|react}} {{app_name}}
```

- Install `npm` dependencies after the project has been created:

```
lambo new --node {{app_name}}
```

- Create a Valet site after the project has been created:

```
lambo new --link {{app_name}}
```

- Create a new MySQL database with the same name as the project:

```
lambo new --create-db --dbuser={{user}} --
dbpassword={{password}} {{app_name}}
```

- Open a specific editor after the project has been created:

```
lambo new --editor="{{editor}}" {{app_name}}
```

lambo

A super-powered **`laravel new`** for Laravel and Valet.

See also: **`lambo new`** for additional command flags.

More information: https://github.com/tighten/lambo#usage.

- Create a new Laravel application:

  ```
  lambo new {{app_name}}
  ```

- Open the configuration in your default editor:

  ```
  lambo edit-config
  ```

- Open the configuration in a specific editor:

  ```
  lambo edit-config {{[-e|--editor]}} "{{path/to/editor}}"
  ```

- Open the configuration file that is run after new applications have been scaffolded:

  ```
  lambo edit-after
  ```

lame

Encode WAV to MP3 files.

More information: https://svn.code.sf.net/p/lame/svn/trunk/lame/USAGE.

- Encode an audio file to MP3 using CBR 320 kbit/second:

```
lame -b 320 {{path/to/file}}.wav {{path/to/output}}.mp3
```

- Encode an audio file to MP3 using the V0 preset:

```
lame -V 0 {{path/to/file}}.wav {{path/to/output}}.mp3
```

- Encode an audio file to AAC:

```
lame {{path/to/file}}.wav {{path/to/output}}.aac
```

lando

Local development environment and DevOps tool built on Docker.

More information: https://docs.lando.dev/cli/.

- Initialize code for use with `lando`:

  ```
  lando init
  ```

- Print information about your app:

  ```
  lando info
  ```

- Start your app:

  ```
  lando start
  ```

- Stop your app:

  ```
  lando stop
  ```

- Restart your app:

  ```
  lando restart
  ```

- Rebuild your app from scratch, preserving data:

  ```
  lando rebuild
  ```

- Display logs for your app:

  ```
  lando logs
  ```

- Destroy your app:

  ```
  lando destroy
  ```

laravel-zero

An installer for the Laravel Zero framework.

More information: https://laravel-zero.com.

- Create a new Laravel Zero application:

  ```
  laravel-zero new {{name}}
  ```

- Update the installer to the latest version:

  ```
  laravel-zero self-update
  ```

- List the available installer commands:

  ```
  laravel-zero list
  ```

laravel

Installer for the Laravel framework.

More information: https://laravel.com/docs/installation.

- Create a new Laravel application:

```
laravel new {{name}}
```

- Use the latest development release:

```
laravel new {{name}} --dev
```

- Overwrite if the directory already exists:

```
laravel new {{name}} --force
```

- Install the Laravel Jetstream scaffolding:

```
laravel new {{name}} --jet
```

- Install the Laravel Jetstream scaffolding with a specific stack:

```
laravel new {{name}} --jet --stack {{livewire|inertia}}
```

- Install the Laravel Jetstream scaffolding with support for teams:

```
laravel new {{name}} --jet --teams
```

- List the available installer commands:

```
laravel list
```

lastcomm

Show last commands executed.

More information: https://manpages.debian.org/latest/acct/lastcomm.1.en.html.

- Print information about all the commands in the acct (record file):

```
lastcomm
```

- Display commands executed by a given user:

```
lastcomm --user {{user}}
```

- Display information about a given command executed on the system:

```
lastcomm --command {{command}}
```

- Display information about commands executed on a given terminal:

```
lastcomm --tty {{terminal_name}}
```

latex

Compile a DVI document from LaTeX source files.

More information: https://www.latex-project.org.

- Compile a DVI document:

```
latex {{source.tex}}
```

- Compile a DVI document, specifying an output directory:

```
latex -output-directory={{path/to/directory}} {{source.tex}}
```

- Compile a DVI document, exiting on each error:

```
latex -halt-on-error {{source.tex}}
```

latexdiff

Determine differences between two LaTeX files.

More information: https://ctan.org/pkg/latexdiff.

- Determine changes between different versions of a LaTeX file (the resulting LaTeX file can be compiled to show differences underlined):

```
latexdiff {{old.tex}} {{new.tex}} > {{diff.tex}}
```

- Determine changes between different versions of a LaTeX file by highlighting differences in boldface:

```
latexdiff --type=BOLD {{old.tex}} {{new.tex}} > {{diff.tex}}
```

- Determine changes between different versions of a LaTeX file, and display minor changes in equations with both added and deleted graphics:

```
latexdiff --math-markup=fine --graphics-markup=both
{{old.tex}} {{new.tex}} > {{diff.tex}}
```

latexmk

Compile LaTeX source files into finished documents.

Automatically does multiple runs when needed.

More information: https://mg.readthedocs.io/latexmk.html.

- Compile a DVI (Device Independent file) document from every source:

```
latexmk
```

- Compile a DVI document from a specific source file:

```
latexmk {{path/to/source.tex}}
```

- Compile a PDF document:

```
latexmk -pdf {{path/to/source.tex}}
```

- Open the document in a viewer and continuously update it whenever source files change:

```
latexmk -pvc {{path/to/source.tex}}
```

- Force the generation of a document even if there are errors:

```
latexmk -f {{path/to/source.tex}}
```

- Clean up temporary TEX files created for a specific TEX file:

```
latexmk -c {{path/to/source.tex}}
```

- Clean up all temporary TEX files in the current directory:

```
latexmk -c
```

latexpand

Simplify LaTeX source files by removing comments and resolving **`\include`**s, **`\input`**s, etc.

More information: https://www.ctan.org/pkg/latexpand.

- Simplify the specified source file and save the result to the specified output file:

```
latexpand {{[-o|--output]}} {{path/to/output.tex}} {{path/to/file.tex}}
```

- Do not remove comments:

```
latexpand --keep-comments {{[-o|--output]}} {{path/to/output.tex}} {{path/to/file.tex}}
```

- Do not expand `\includes`, `\inputs` etc.:

```
latexpand --keep-includes {{[-o|--output]}} {{path/to/output.tex}} {{path/to/file.tex}}
```

- Expand `\usepackages` as far as the corresponding STY files can be found:

```
latexpand --expand-usepackage {{[-o|--output]}} {{path/to/output.tex}} {{path/to/file.tex}}
```

- Inline the specified BBL file:

```
latexpand --expand-bbl {{path/to/bibliography.bbl}} {{[-o|--output]}} {{path/to/output.tex}} {{path/to/file.tex}}
```

laydown

Command line application to prepare for Daily Standup.

More information: https://github.com/badjr13/laydown.

- Add an item to DID section:

  ```
  laydown did {{item}}
  ```

- Add an item to DOING section:

  ```
  laydown doing {{item}}
  ```

- Clear all items:

  ```
  laydown clear
  ```

- Use an editor to edit current data:

  ```
  laydown edit
  ```

- Archive and clear current data:

  ```
  laydown archive
  ```

lazydocker

A terminal UI for managing Docker containers, images, volumes, and more.

More information: https://github.com/jesseduffield/lazydocker.

- Open Lazydocker:

```
lazydocker
```

- Display the current default configuration:

```
lazydocker --config
```

- Run Lazydocker in debug mode:

```
lazydocker --debug
```

- Specify an alternate Docker Compose file:

```
lazydocker --file {{path/to/docker-compose.yml}}
```

- Display help:

```
lazydocker --help
```

- Display version:

```
lazydocker --version
```

lazygit

A simple terminal UI for Git commands, providing an intuitive interface for managing repositories.

More information: https://github.com/jesseduffield/lazygit.

- Open Lazygit in the current repository:

```
lazygit
```

- Open Lazygit for a specific Git repository:

```
lazygit --path {{path/to/repository}}
```

- Start Lazygit with focus on a specific panel:

```
lazygit {{status|branch|log|stash|...}}
```

- Print the default Lazygit configuration:

```
lazygit --config
```

- Tail the Lazygit logs (useful with debug mode in another terminal):

```
lazygit --logs
```

- Run Lazygit in debug mode:

```
lazygit --debug
```

- Print the configuration directory:

```
lazygit --print-config-dir
```

lb

Manage a blog contained in the current directory.

Drafts and posts to operate on are selected interactively when running the commands.

More information: https://github.com/LukeSmithxyz/lb.

- Make a new draft:

```
lb new
```

- Edit a draft:

```
lb edit
```

- Delete a draft:

```
lb trash
```

- Publish a draft:

```
lb publish
```

- Delete a published post:

```
lb delete
```

- Unpublish a published post to edit it as a draft again:

```
lb revise
```

lckdo

This command is deprecated and has been superseded by **`flock`**.

More information: https://manned.org/lckdo.

- View documentation for the recommended replacement:

```
tldr flock
```

ld

Link object files together.

More information: https://sourceware.org/binutils/docs/ld.html.

- Link a specific object file with no dependencies into an executable:

```
ld {{path/to/file.o}} {{[-o|--output]}} {{path/to/output_executable}}
```

- Link two object files together:

```
ld {{path/to/file1.o}} {{path/to/file2.o}} {{[-o|--output]}} {{path/to/output_executable}}
```

- Dynamically link an x86_64 program to glibc (file paths change depending on the system):

```
ld {{[-o|--output]}} {{path/to/output_executable}} {{[-I|--dynamic-linker]}} /lib/ld-linux-x86-64.so.2 /lib/crt1.o /lib/crti.o -lc {{path/to/file.o}} /lib/crtn.o
```

ldapsearch

Query an LDAP directory.

More information: https://docs.ldap.com/ldap-sdk/docs/tool-usages/ldapsearch.html.

- Query an LDAP server for all items that are a member of the given group and return the object's displayName value:

```
ldapsearch {{[-D|--bindDN]}} '{{admin_DN}}' {{[-w|--
bindPassword]}} '{{password}}' {{[-h|--hostname]}}
{{ldap_host}} {{[-b|--baseDN]}} {{base_ou}}
'{{memberOf=group1}}' displayName
```

- Query an LDAP server with a no-newline password file for all items that are a member of the given group and return the object's displayName value:

```
ldapsearch {{[-D|--bindDN]}} '{{admin_DN}}' {{[-u|--
keyStorePasswordFile]}} '{{password_file}}' {{[-h|--
hostname]}} {{ldap_host}} {{[-b|--baseDN]}} {{base_ou}}
'{{memberOf=group1}}' displayName
```

- Return 5 items that match the given filter:

```
ldapsearch {{[-D|--bindDN]}} '{{admin_DN}}' {{[-w|--
bindPassword]}} '{{password}}' {{[-h|--hostname]}}
{{ldap_host}} {{[-b|--baseDN]}} {{base_ou}}
'{{memberOf=group1}}' {{[-z|--sizeLimit]}} 5 displayName
```

- Wait up to 7 seconds for a response:

```
ldapsearch {{[-D|--bindDN]}} '{{admin_DN}}' {{[-w|--
bindPassword]}} '{{password}}' {{[-h|--hostname]}}
{{ldap_host}} {{[-b|--baseDN]}} {{base_ou}}
'{{memberOf=group1}}' {{[-l|--timeLimitSeconds]}} 7
displayName
```

- Invert the filter:

```
ldapsearch {{[-D|--bindDN]}} '{{admin_DN}}' {{[-w|--
bindPassword]}} '{{password}}' {{[-h|--hostname]}}
{{ldap_host}} {{[-b|--baseDN]}} {{base_ou}} '(!
(memberOf={{group1}}))' displayName
```

- Return all items that are part of multiple groups, returning the display name for each item:

```
ldapsearch {{[-D|--bindDN]}} '{{admin_DN}}' {{[-w|--
bindPassword]}} '{{password}}' {{[-h|--hostname]}}
{{ldap_host}} '(&({{memberOf=group1}})({{memberOf=group2}})
({{memberOf=group3}}))' "displayName"
```

- Return all items that are members of at least 1 of the specified groups:

```
ldapsearch {{[-D|--bindDN]}} '{{admin_DN}}' {{[-w|--
bindPassword]}} '{{password}}' {{[-h|--hostname]}}
{{ldap_host}} '(|({{memberOf=group1}})({{memberOf=group1}})
({{memberOf=group3}}))' displayName
```

- Combine multiple boolean logic filters:

```
ldapsearch {{[-D|--bindDN]}} '{{admin_DN}}' {{[-w|--
bindPassword]}} '{{password}}' {{[-h|--hostname]}}
{{ldap_host}} '(&({{memberOf=group1}})({{memberOf=group2}})(!
({{memberOf=group3}})))' displayName
```

ldc

D compiler using LLVM as a backend.

More information: https://wiki.dlang.org/Using_LDC.

- Compile a source code file into an executable binary:

```
ldc2 {{path/to/source.d}} -of={{path/to/output_executable}}
```

- Compile the source code file without linking:

```
ldc2 -c {{path/to/source.d}}
```

- Select the target architecture and OS:

```
ldc -mtriple={{architecture_OS}} -c {{path/to/source.d}}
```

- Display help:

```
ldc2 -h
```

- Display complete help:

```
ldc2 -help-hidden
```

leaftoppm

Convert an Interleaf image to a PPM image.

More information: https://netpbm.sourceforge.net/doc/leaftoppm.html.

- Generate a PPM image file as output for an Interleaf image file as input:

```
leaftoppm {{path/to/file.pl}}
```

- Display version:

```
leaftoppm {{[-v|-version]}}
```

leave

Set a reminder for when it's time to leave.

To remove reminders use **`kill $(pidof leave)`**.

More information: https://www.freebsd.org/cgi/man.cgi?query=leave.

- Set a reminder at a given time:

```
leave {{time_to_leave}}
```

- Set a reminder to leave at noon:

```
leave {{1200}}
```

- Set a reminder in a specific amount of time:

```
leave +{{amount_of_time}}
```

- Set a reminder to leave in 4 hours and 4 minutes:

```
leave +{{0404}}
```

lebab

A JavaScript modernizer for transpiling code to ES6/ES7.

Transformations must be provided for all examples.

More information: https://github.com/lebab/lebab.

- Transpile using one or more comma-separated transformations:

```
lebab --transform {{transformation1,transformation2,...}}
```

- Transpile a file to `stdout`:

```
lebab {{path/to/input_file}}
```

- Transpile a file to the specified output file:

```
lebab {{path/to/input_file}} --out-file {{path/to/
output_file}}
```

- Replace all `.js` files in-place in the specified directory, glob or file:

```
lebab --replace {{directory|glob|file}}
```

- Display help:

```
lebab --help
```

ledger

A powerful, double-entry accounting system.

More information: https://www.ledger-cli.org.

- Print a balance report showing totals:

```
ledger balance --file {{path/to/ledger.journal}}
```

- List all postings in Expenses ordered by amount:

```
ledger register {{expenses}} --sorted {{amount}}
```

- Print total Expenses other than Drinks and Food:

```
ledger balance {{Expenses}} and not ({{Drinks}} or {{Food}})
```

- Print a budget report:

```
ledger budget
```

- Print summary information about all the postings:

```
ledger stats
```

lein

Manage Clojure projects with declarative configuration.

More information: https://leiningen.org.

- Generate scaffolding for a new project based on a template:

  ```
  lein new {{template_name}} {{project_name}}
  ```

- Start a REPL session either with the project or standalone:

  ```
  lein repl
  ```

- Run the project's `-main` function with optional args:

  ```
  lein run {{args}}
  ```

- Run the project's tests:

  ```
  lein test
  ```

- Package up the project files and all its dependencies into a jar file:

  ```
  lein uberjar
  ```

lerna

Manage JavaScript projects with multiple packages.

More information: https://lerna.js.org.

- Initialize project files (`lerna.json`, `package.json`, `.git`, etc.):

```
lerna init
```

- Install all external dependencies of each package and symlink together local dependencies:

```
lerna bootstrap
```

- Run a specific script for every package that contains it in its `package.json`:

```
lerna run {{script}}
```

- Execute an arbitrary shell command in every package:

```
lerna exec -- {{ls}}
```

- Publish all packages that have changed since the last release:

```
lerna publish
```

<>

Open a file descriptor for read and write.

More information: https://www.gnu.org/software/bash/manual/bash.html#Opening-File-Descriptors-for-Reading-and-Writing.

- Open a file in a file descriptor for read an write:

```
exec {{3}}<>{{path/to/file}}
```

- Open a file descriptor to a remote connection:

```
exec {{3}}<>/dev/{{tcp}}/{{remote_host}}/{{port_number}}
```

- Close a file descriptor:

```
exec {{3}}<>-
```

<

Redirect data to **`stdin`**.

More information: https://gnu.org/software/bash/manual/bash.html#Redirecting-Input.

- Redirect a file to `stdin` (achieves the same effect as `cat file.txt |`):

```
{{command}} < {{path/to/file.txt}}
```

- Create a here document and pass that into `stdin` (requires a multiline command):

```
{{command}} << {{EOF}} <Enter> {{multiline_text}} <Enter>
{{EOF}}
```

- Create a here string and pass that into `stdin` (achieves the same effect as `echo string |`):

```
{{command}} <<< {{string}}
```

- Process data from a file and write the output to another file:

```
{{command}} < {{path/to/file.txt}} > {{path/to/file2.txt}}
```

- Write a here document into a file:

```
cat << {{EOF}} > {{path/to/file.txt}} <Enter>
{{multiline_data}} <Enter> {{EOF}}
```

- Disregard leading tabs (good for scripts with indentation but does not work for spaces):

```
cat <<- {{EOF}} > {{path/to/file.txt}} <Enter>
{{multiline_data}} <Enter> {{EOF}}
```

- Pass command output to a program as a file descriptor:

```
diff <({{command1}}) <({{command2}})
```

- Open a persistent file descriptor:

```
exec {{3}}<{{path/to/file}}
```

less

Open a file for interactive reading, allowing scrolling and search.

More information: https://greenwoodsoftware.com/less/.

- Open a file:

```
less {{source_file}}
```

- Page down/up:

```
{{<Space>|<b>}}
```

- Go to end/start of file:

```
{{<G>|<g>}}
```

- Forward search for a string (press <n>/<N> to go to next/previous match):

```
</>{{something}}
```

- Backward search for a string (press <n>/<N> to go to next/previous match):

```
<?>{{something}}
```

- Follow the output of the currently opened file:

```
<F>
```

- Open the current file in an editor:

```
<v>
```

- Exit:

```
<q>
```

let

Evaluate arithmetic expressions in shell.

Supports variables, operators, and conditional expressions.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-let.

- Evaluate a simple arithmetic expression:

```
let "{{result = a + b}}"
```

- Use post-increment and assignment in an expression:

```
let "{{x++}}"
```

- Use conditional operator in an expression:

```
let "{{result = (x > 10) ? x : 0}}"
```

- Display help:

```
let --help
```

lex

Lexical analyzer generator.

Given the specification for a lexical analyzer, generates C code implementing it.

Note: On most major OSes, this command is an alias for **`flex`**.

More information: https://manned.org/lex.

- Generate an analyzer from a Lex file, storing it to the file `lex.yy.c`:

```
lex {{analyzer.l}}
```

- Specify the output file:

```
lex -t {{analyzer.l}} > {{analyzer.c}}
```

- Compile a C file generated by Lex:

```
c99 {{path/to/lex.yy.c}} -o {{executable}}
```

libreoffice

This command is an alias of **`soffice`**.

- View documentation for the original command:

  ```
  tldr soffice
  ```

license

Create license files for open-source projects.

More information: https://nishanths.github.io/license.

- Print a license to `stdout`, using the defaults (auto-detected author name, and current year):

```
license {{license_name}}
```

- Generate a license and save it to a file:

```
license -o {{path/to/file}} {{license_name}}
```

- List all available licenses:

```
license ls
```

- Generate a license with custom author name and year:

```
license --name {{author}} --year {{release_year}}
{{license_name}}
```

licensor

Write licenses to **`stdout`**.

More information: https://github.com/raftario/licensor.

- Write the MIT license to a file named `LICENSE`:

```
licensor {{MIT}} > {{LICENSE}}
```

- Write the MIT license with a placeholder copyright notice to a file named `LICENSE`:

```
licensor {{[-p|--keep-placeholder]}} {{MIT}} > {{LICENSE}}
```

- Specify a copyright holder named Bobby Tables:

```
licensor {{MIT}} "{{Bobby Tables}}" > {{LICENSE}}
```

- Specify license exceptions with a WITH expression:

```
licensor "{{Apache-2.0 WITH LLVM-exception}}" > {{LICENSE}}
```

- List all available licenses:

```
licensor {{[-l|--licenses]}}
```

- List all available exceptions:

```
licensor {{[-e|--exceptions]}}
```

light-arionum-cli

The PHP light wallet for the Arionum cryptocurrency.

More information: https://github.com/arionum/lightWalletCLI.

- Generate a new public/private key pair:

```
light-arionum-cli
```

- Display the balance of the current address:

```
light-arionum-cli balance
```

- Display the balance of the specified address:

```
light-arionum-cli balance {{address}}
```

- Send a transaction with an optional message:

```
light-arionum-cli send {{address}} {{value}}
{{optional_message}}
```

- Export the current wallet information:

```
light-arionum-cli export
```

- Display information about the current block:

```
light-arionum-cli block
```

- Display information about the current address' transactions:

```
light-arionum-cli transactions
```

- Display information about a specific transaction:

```
light-arionum-cli transaction {{transaction_id}}
```

lighthouse

Analyzes web applications and web pages, collecting modern performance metrics and insights on developer best practices.

More information: https://github.com/GoogleChrome/lighthouse.

- Generate an HTML report for a specific website and save it to a file in the current directory:

```
lighthouse {{https://example.com}}
```

- Generate a JSON report and print it:

```
lighthouse --output {{json}} {{https://example.com}}
```

- Generate a JSON report and save it to a specific file:

```
lighthouse --output {{json}} --output-path {{path/to/
file.json}} {{https://example.com}}
```

- Generate a report using the browser in headless mode without logging to `stdout`:

```
lighthouse --quiet --chrome-flags="{{--headless}}" {{https://
example.com}}
```

- Generate a report, using the HTTP header key/value pairs in the specified JSON file for all requests:

```
lighthouse --extra-headers={{path/to/file.json}} {{https://
example.com}}
```

- Generate a report for specific categories only:

```
lighthouse --only-
categories={{performance,accessibility,best-
practices,seo,pwa}} {{https://example.com}}
```

- Generate a report with device emulation and all throttling disabled:

```
lighthouse --screenEmulation.disabled --throttling-
method={{provided}} --no-emulatedUserAgent {{https://
example.com}}
```

- Display help:

```
lighthouse --help
```

lilypond

Typeset music and/or produce MIDI from file.

See also: **`musescore`**.

More information: https://lilypond.org.

- Compile a lilypond file into a PDF:

```
lilypond {{path/to/file}}
```

- Compile into the specified format:

```
lilypond {{[-f|--format]}} {{format_dump}} {{path/to/file}}
```

- Compile the specified file, suppressing progress updates:

```
lilypond {{[-s|--silent]}} {{path/to/file}}
```

- Compile the specified file, and also specify the output filename:

```
lilypond {{[-o|--output]}} {{path/to/output_file}} {{path/to/input_file}}
```

- Display version:

```
lilypond {{[-v|--version]}}
```

lima

This command is an alias of **`limactl shell`** for the default VM instance.

You can also set the **`$LIMA_INSTANCE`** environment variable to work on a different instance.

- View documentation for the original command:

```
tldr limactl
```

limactl

Virtual machine manager for Linux guests, with multiple VM templates available.

Can be used to run containers on macOS, but also for generic virtual machine use cases on macOS and Linux hosts.

More information: https://github.com/lima-vm/lima.

- List VMs:

```
limactl list
```

- Create a VM using the default settings and optionally provide a name and/or a template (see `limactl create --list-templates` for available templates):

```
limactl create --name {{vm_name}} template://{{debian|fedora|ubuntu|...}}
```

- Start a VM (this might install some dependencies in it and take a few minutes):

```
limactl start {{vm_name}}
```

- Open a remote shell inside a VM:

```
limactl shell {{vm_name}}
```

- Run a command inside a VM:

```
limactl shell {{vm_name}} {{command}}
```

- Stop/shutdown a VM:

```
limactl stop {{vm_name}}
```

- Delete a VM:

```
limactl remove {{vm_name}}
```

link

Create a hard link to an existing file.

For more options, see the **`ln`** command.

More information: https://www.gnu.org/software/coreutils/manual/html_node/link-invocation.html.

- Create a hard link from a new file to an existing file:

```
link {{path/to/existing_file}} {{path/to/new_file}}
```

linkchecker

Client to check HTML documents and websites for broken links.

More information: https://linkchecker.github.io/linkchecker/man/linkchecker.html.

- Find broken links on https://example.com/:

```
linkchecker {{https://example.com/}}
```

- Also check URLs that point to external domains:

```
linkchecker --check-extern {{https://example.com/}}
```

- Ignore URLs that match a specific `regex`:

```
linkchecker --ignore-url {{regex}} {{https://example.com/}}
```

- Output results to a CSV file:

```
linkchecker --file-output {{csv}}/{{path/to/file}} {{https://example.com/}}
```

linode-cli account

Manage Linode accounts.

See also: **`linode-cli`**.

More information: https://techdocs.akamai.com/cloud-computing/docs/cli-commands-for-account-management.

- View account:

```
linode-cli account view
```

- View account settings:

```
linode-cli account settings
```

- Make a payment:

```
linode-cli account payment-create --cvv {{cvv}} --usd {{amount_in_dollars}}
```

- View account notifications:

```
linode-cli account notifications-list
```

linode-cli domains

Manage Linode Domains and DNS configuration.

See also: **`linode-cli`**.

More information: https://techdocs.akamai.com/cloud-computing/docs/cli-commands-for-the-dns-manager.

- List all managed domains:

```
linode-cli domains list
```

- Create a new managed domain:

```
linode-cli domains create --domain {{domain_name}} --type
{{master|slave}} --soa-email {{email}}
```

- View details of a specific domain:

```
linode-cli domains view {{domain_id}}
```

- Delete a managed domain:

```
linode-cli domains delete {{domain_id}}
```

- List records for a specific domain:

```
linode-cli domains records-list {{domain_id}}
```

- Add a DNS record to a domain:

```
linode-cli domains records-create {{domain_id}} --type {{A|
AAAA|CNAME|MX|...}} --name {{subdomain}} --target
{{target_value}}
```

- Update a DNS record for a domain:

```
linode-cli domains records-update {{domain_id}} {{record_id}}
--target {{new_target_value}}
```

- Delete a DNS record from a domain:

```
linode-cli domains records-delete {{domain_id}} {{record_id}}
```

linode-cli events

Manage Linode events.

See also: **`linode-cli`**.

More information: https://techdocs.akamai.com/cloud-computing/docs/cli-commands-for-viewing-events.

- List the events on your account:

```
linode-cli events list
```

- View details about a specific event:

```
linode-cli events view {{event_id}}
```

- Mark an event as read:

```
linode-cli events mark-read {{event_id}}
```

linode-cli linodes

Manage Linode instances.

See also: **`linode-cli`**.

More information: https://techdocs.akamai.com/cloud-computing/docs/cli-commands-for-compute-instances.

- List all Linodes:

```
linode-cli linodes list
```

- Create a new Linode:

```
linode-cli linodes create --type {{linode_type}} --region {{region}} --image {{image_id}}
```

- View details of a specific Linode:

```
linode-cli linodes view {{linode_id}}
```

- Update settings for a Linode:

```
linode-cli linodes update {{linode_id}} --label {{new_label}}
```

- Delete a Linode:

```
linode-cli linodes delete {{linode_id}}
```

- Perform a power management operation on a Linode:

```
linode-cli linodes {{boot|reboot|shutdown}} {{linode_id}}
```

- List available backups for a Linode:

```
linode-cli linodes backups-list {{linode_id}}
```

- Restore a backup to a Linode:

```
linode-cli linodes backups-restore {{linode_id}} --backup-id {{backup_id}}
```

linode-cli lke

Manage Linode Kubernetes Engine (LKE) clusters.

See also: **`linode-cli`**.

More information: https://techdocs.akamai.com/cloud-computing/docs/cli-commands-for-lke.

- List all LKE clusters:

```
linode-cli lke clusters list
```

- Create a new LKE cluster:

```
linode-cli lke clusters create --region {{region}} --type {{type}} --node-type {{node_type}} --nodes-count {{count}}
```

- View details of a specific LKE cluster:

```
linode-cli lke clusters view {{cluster_id}}
```

- Update an existing LKE cluster:

```
linode-cli lke clusters update {{cluster_id}} --node-type {{new_node_type}}
```

- Delete an LKE cluster:

```
linode-cli lke clusters delete {{cluster_id}}
```

linode-cli nodebalancers

Manage Linode NodeBalancers.

See also: **`linode-cli`**.

More information: https://techdocs.akamai.com/cloud-computing/docs/cli-commands-for-nodebalancers.

- List all NodeBalancers:

```
linode-cli nodebalancers list
```

- Create a new NodeBalancer:

```
linode-cli nodebalancers create --region {{region}}
```

- View details of a specific NodeBalancer:

```
linode-cli nodebalancers view {{nodebalancer_id}}
```

- Update an existing NodeBalancer:

```
linode-cli nodebalancers update {{nodebalancer_id}} --label {{new_label}}
```

- Delete a NodeBalancer:

```
linode-cli nodebalancers delete {{nodebalancer_id}}
```

- List configurations for a NodeBalancer:

```
linode-cli nodebalancers configs list {{nodebalancer_id}}
```

- Add a new configuration to a NodeBalancer:

```
linode-cli nodebalancers configs create {{nodebalancer_id}} --port {{port}} --protocol {{protocol}}
```

linode-cli object-storage

Manage Linode Object Storage.

See also: **`linode-cli`**.

More information: https://techdocs.akamai.com/cloud-computing/docs/cli-commands-for-object-storage.

- List all Object Storage buckets:

```
linode-cli object-storage buckets list
```

- Create a new Object Storage bucket:

```
linode-cli object-storage buckets create --cluster
{{cluster_id}} --label {{bucket_label}}
```

- Delete an Object Storage bucket:

```
linode-cli object-storage buckets delete {{cluster_id}}
{{bucket_label}}
```

- List Object Storage cluster regions:

```
linode-cli object-storage clusters list
```

- List access keys for Object Storage:

```
linode-cli object-storage keys list
```

- Create a new access key for Object Storage:

```
linode-cli object-storage keys create --label {{label}}
```

- Revoke an access key for Object Storage:

```
linode-cli object-storage keys revoke {{access_key_id}}
```

linode-cli tickets

Manage Linode Support Tickets.

See also: **`linode-cli`**.

More information: https://techdocs.akamai.com/cloud-computing/docs/cli-commands-for-account-management.

- List your Support Tickets:

```
linode-cli tickets list
```

- Open a new Ticket:

```
linode-cli tickets create --summary "{{Summary or quick title for the Ticket}}" --description "{{Detailed description of the issue}}"
```

- List replies to a Ticket:

```
linode-cli tickets replies {{ticket_id}}
```

- Reply to a specific Ticket:

```
linode-cli tickets reply {{ticket_id}} --description "{{The content of your reply}}"
```

linode-cli volumes

Manage Linode Volumes.

See also: **`linode-cli`**.

More information: https://techdocs.akamai.com/cloud-computing/docs/cli-commands-for-block-storage-volumes.

- List current Volumes:

```
linode-cli volumes list
```

- Create a new Volume and attach it to a specific Linode:

```
linode-cli volumes create --label {{volume_label}} --size {{size_in_GB}} --linode-id {{linode_id}}
```

- Attach a Volume to a specific Linode:

```
linode-cli volumes attach {{volume_id}} --linode-id {{linode_id}}
```

- Detach a Volume from a Linode:

```
linode-cli volumes detach {{volume_id}}
```

- Resize a Volume (Note: Size can only be increased):

```
linode-cli volumes resize {{volume_id}} --size {{new_size_in_GB}}
```

- Delete a Volume:

```
linode-cli volumes delete {{volume_id}}
```

linode-cli

Manage Linode cloud services.

Some subcommands such as **`events`** have their own usage documentation.

More information: https://techdocs.akamai.com/cloud-computing/docs/getting-started-with-the-linode-cli.

- List all Linodes:

```
linode-cli linodes list
```

- View documentation for managing Linode accounts:

```
tldr linode-cli account
```

- View documentation for managing Linodes:

```
tldr linode-cli linodes
```

- View documentation for managing Linode Kubernetes Engine (LKE) clusters:

```
tldr linode-cli lke
```

- View documentation for managing NodeBalancers:

```
tldr linode-cli nodebalancers
```

- View documentation for managing Object Storage:

```
tldr linode-cli object-storage
```

- View documentation for managing domains and DNS configuration:

```
tldr linode-cli domains
```

- View documentation for managing Linode Volumes:

```
tldr linode-cli volumes
```

liquidctl

Control liquid coolers.

More information: https://github.com/liquidctl/liquidctl.

- List available devices:

```
liquidctl list
```

- Initialize all supported devices:

```
sudo liquidctl initialize all
```

- Print the status of available liquid coolers:

```
liquidctl status
```

- Match a string in product name to pick a device and set its fan speed to 0% at 20°C, 50% at 50°C and 100% at 70°C:

```
liquidctl --match {{string}} set fan speed {{20 0 50 50 70 100}}
```

lispmtopgm

Convert a Lisp Machine bitmap into a PGM image.

See also: **`pgmtolispm`**.

More information: https://netpbm.sourceforge.net/doc/lispmtopgm.html.

- Convert the specified Lisp Machine bitmap file into a PGM image:

```
lispmtopgm {{path/to/input.lispm}} > {{path/to/output.pgm}}
```

lit

LLVM integrated tester for executing LLVM and Clang style test suites, summarizing results.

Part of LLVM.

More information: https://www.llvm.org/docs/CommandGuide/lit.html.

- Run a specified test case:

```
lit {{path/to/test_file.test}}
```

- Run all test cases in a specified directory:

```
lit {{path/to/test_suite}}
```

- Run all test cases and check the wall time for each cases, then report to summary output:

```
lit {{path/to/test_suite}} --time-tests
```

- Run individual tests with Valgrind (memory check and memory leak test):

```
lit {{path/to/test_file.test}} --vg --vg-leak --vg-args={{args_to_valgrind}}
```

live-server

A simple development HTTP server with live reload capability.

More information: https://github.com/tapio/live-server.

- Serve an `index.html` file and reload on changes:

  `live-server`

- Specify a port (default is 8080) from which to serve a file:

  `live-server --port={{8081}}`

- Specify a given file to serve:

  `live-server --open={{about.html}}`

- Proxy all requests for ROUTE to URL:

  `live-server --proxy={{/}}:{{http:localhost:3000}}`

llc

Compiles LLVM Intermediate Representation or bitcode to target-specific assembly language.

More information: https://www.llvm.org/docs/CommandGuide/llc.html.

- Compile a bitcode or IR file to an assembly file with the same base name:

```
llc {{path/to/file.ll}}
```

- Enable all optimizations:

```
llc -O3 {{path/to/input.ll}}
```

- Output assembly to a specific file:

```
llc --output {{path/to/output.s}}
```

- Emit fully relocatable, position independent code:

```
llc -relocation-model=pic {{path/to/input.ll}}
```

lldb

The LLVM Low-Level Debugger.

More information: https://lldb.llvm.org.

- Debug an executable:

```
lldb {{executable}}
```

- Attach `lldb` to a running process with a given PID:

```
lldb -p {{pid}}
```

- Wait for a new process to launch with a given name, and attach to it:

```
lldb -w -n {{process_name}}
```

lli

Directly execute programs from LLVM bitcode.

More information: https://www.llvm.org/docs/CommandGuide/lli.html.

- Execute a bitcode or IR file:

```
lli {{path/to/file.ll}}
```

- Execute with command-line arguments:

```
lli {{path/to/file.ll}} {{argument1 argument2 ...}}
```

- Enable all optimizations:

```
lli -O3 {{path/to/file.ll}}
```

- Load a dynamic library before linking:

```
lli --dlopen={{path/to/library.dll}} {{path/to/file.ll}}
```

llm

Interact with Large Language Models (LLMs) via remote APIs and models that can be installed and run on your machine.

More information: https://llm.datasette.io/en/stable/help.html.

- Set up an OpenAI API Key:

```
llm keys set openai
```

- Run a prompt:

```
llm "{{Ten fun names for a pet pelican}}"
```

- Run a system prompt against a file:

```
cat {{path/to/file.py}} | llm {{[-s|--system]}} "{{Explain
this code}}"
```

- Install packages from PyPI into the same environment as LLM:

```
llm install {{package1 package2 ...}}
```

- Download and run a prompt against a model:

```
llm {{[-m|--model]}} {{orca-mini-3b-gguf2-q4_0}} "{{What is
the capital of France?}}"
```

- Create a system prompt and save it with a template name:

```
llm {{[-s|--system]}} '{{You are a sentient cheesecake}}' --
save {{sentient_cheesecake}}
```

- Have an interactive chat with a specific model using a specific template:

```
llm chat {{[-m|--model]}} {{chatgpt}} {{[-t|--template]}}
{{sentient_cheesecake}}
```

llvd

Linkedin Learning Video Downloader.

More information: https://github.com/knowbee/llvd.

- Download a course using cookie-based authentication:

```
llvd {{[-c|--course]}} {{course-slug}} --cookies
```

- Download a course at a specific resolution:

```
llvd {{[-c|--course]}} {{course-slug}} {{[-r|--resolution]}}
720
```

- Download a course with captions (subtitles):

```
llvd {{[-c|--course]}} {{course-slug}} {{[-ca|--caption]}}
```

- Download a course path with throttling between 10 to 30 seconds:

```
llvd {{[-p|--path]}} {{path-slug}} {{[-t|--throttle]}}
{{10,30}} --cookies
```

llvm-ar

This command is an alias of **ar**.

- View documentation for the original command:

```
tldr ar
```

llvm-as

LLVM Intermediate Representation (`.ll`) to Bitcode (`.bc`) assembler.

More information: https://llvm.org/docs/CommandGuide/llvm-as.html.

- Assemble an IR file:

```
llvm-as -o {{path/to/out.bc}} {{path/to/source.ll}}
```

- Assemble an IR file and include a module hash in the produced Bitcode file:

```
llvm-as --module-hash -o {{path/to/out.bc}} {{path/to/
source.ll}}
```

- Read an IR file from `stdin` and assemble it:

```
cat {{path/to/source.ll}} | llvm-as -o {{path/to/out.bc}}
```

llvm-bcanalyzer

LLVM Bitcode (`.bc`) analyzer.

More information: https://llvm.org/docs/CommandGuide/llvm-bcanalyzer.html.

- Print statistics about a Bitcode file:

```
llvm-bcanalyzer {{path/to/file.bc}}
```

- Print an SGML representation and statistics about a Bitcode file:

```
llvm-bcanalyzer -dump {{path/to/file.bc}}
```

- Read a Bitcode file from `stdin` and analyze it:

```
cat {{path/to/file.bc}} | llvm-bcanalyzer
```

llvm-cat

Concatenate LLVM Bitcode (**`.bc`**) files.

More information: https://github.com/llvm/llvm-project/blob/main/llvm/tools/llvm-cat/llvm-cat.cpp.

- Concatenate Bitcode files:

```
llvm-cat {{path/to/file1.bc}} {{path/to/file2.bc}} -o {{path/to/out.bc}}
```

llvm-config

Get various configuration information needed to compile programs which use LLVM.

Typically called from build systems, like in Makefiles or configure scripts.

More information: https://llvm.org/docs/CommandGuide/llvm-config.html.

- Compile and link an LLVM based program:

```
clang++ $(llvm-config --cxxflags --ldflags --libs) --output {{path/to/output_executable}} {{path/to/source.cc}}
```

- Print the PREFIX of your LLVM installation:

```
llvm-config --prefix
```

- Print all targets supported by your LLVM build:

```
llvm-config --targets-built
```

llvm-dis

Convert LLVM bitcode files into human-readable LLVM Intermediate Representation (IR).

More information: https://www.llvm.org/docs/CommandGuide/llvm-dis.html.

- Convert a bitcode file as LLVM IR and write the result to `stdout`:

```
llvm-dis {{path/to/input.bc}} -o -
```

- Convert a bitcode file to an LLVM IR file with the same filename:

```
llvm-dis {{path/to/file.bc}}
```

- Convert a bitcode file to LLVM IR, writing the result to the specified file:

```
llvm-dis {{path/to/input.bc}} -o {{path/to/output.ll}}
```

llvm-g++

This command is an alias of **clang++**.

- View documentation for the original command:

```
tldr clang++
```

llvm-gcc

This command is an alias of **clang**.

- View documentation for the original command:

```
tldr clang
```

llvm-mc

LLVM Machine Code Playground. It provides a set of tools for working with LLVM machine code.

Part of LLVM.

More information: https://llvm.org/docs/CommandGuide/llvm-mc.html.

- Assemble assembly code file into object file with machine code:

```
llvm-mc --filetype=obj -o {{path/to/output.o}} {{path/to/input.s}}
```

- Disassemble object file with machine code into assembly code file:

```
llvm-mc --disassemble -o {{path/to/output.s}} {{path/to/input.o}}
```

- Compile LLVM bit code file into assembly code:

```
llvm-mc -o {{path/to/output.s}} {{path/to/input.bc}}
```

- Assemble assembly code from `stdin` and show encoding to `stdout`:

```
echo "{{addl %eax, %ebx}}" | llvm-mc -show-encoding -show-inst
```

- Disassemble machine code from `stdin` for specified triple:

```
echo "{{0xCD 0x21}}" | llvm-mc --disassemble -triple={{target_name}}
```

llvm-nm

This command is an alias of **nm**.

- View documentation for the original command:

```
tldr nm
```

llvm-objdump

This command is an alias of **`objdump`**.

- View documentation for the original command:

  ```
  tldr objdump
  ```

llvm-strings

This command is an alias of **`strings`**.

- View documentation for the original command:

```
tldr strings
```

lmms

Free, open source, cross-platform digital audio workstation.

Render a **`.mmp`** or **`.mmpz`** project file, dump a **`.mmpz`** as XML, or start the GUI.

See also: **`mixxx`**.

More information: https://lmms.io.

- Start the GUI:

```
lmms
```

- Start the GUI and load external config:

```
lmms --config {{path/to/config.xml}}
```

- Start the GUI and import MIDI or Hydrogen file:

```
lmms --import {{path/to/midi_or_hydrogen_file}}
```

- Start the GUI with a specified window size:

```
lmms --geometry {{x_size}}x{{y_size}}+{{x_offset}}+
{{y_offset}}
```

- Dump a `.mmpz` file:

```
lmms dump {{path/to/mmpz_file.mmpz}}
```

- Render a project file:

```
lmms render {{path/to/mmpz_or_mmp_file}}
```

- Render the individual tracks of a project file:

```
lmms rendertracks {{path/to/mmpz_or_mmp_file}} {{path/to/
dump_directory}}
```

- Render with custom samplerate, format, and as a loop:

```
lmms render --samplerate {{88200}} --format {{ogg}} --loop --
output {{path/to/output_file.ogg}}
```

ln

Create links to files and directories.

More information: https://www.gnu.org/software/coreutils/manual/html_node/ln-invocation.html.

- Create a symbolic link to a file or directory:

```
ln {{[-s|--symbolic]}} /{{path/to/file_or_directory}} {{path/to/symlink}}
```

- Overwrite an existing symbolic link to point to a different file:

```
ln {{[-sf|--symbolic --force]}} /{{path/to/new_file}} {{path/to/symlink}}
```

- Create a hard link to a file:

```
ln /{{path/to/file}} {{path/to/hardlink}}
```

loadtest

Run a load test on the selected HTTP or WebSockets URL.

More information: https://github.com/alexfernandez/loadtest.

- Run with concurrent users and a specified amount of requests per second:

```
loadtest {{[-c|--concurrency]}} {{10}} {{[--rps|--
requestsPerSecond]}} {{200}} {{https://example.com}}
```

- Run with a custom HTTP header:

```
loadtest --headers "{{accept:text/plain;text-
html}}" {{https://example.com}}
```

- Run with a specific HTTP method:

```
loadtest {{[-m|--method]}} {{GET}} {{https://example.com}}
```

loc

Count lines of code. Written in Rust.

More information: https://github.com/cgag/loc.

- Print lines of code in the current directory:

```
loc
```

- Print lines of code in the target directory:

```
loc {{path/to/directory}}
```

- Print lines of code with stats for individual files:

```
loc --files
```

- Print lines of code without .gitignore (etc.) files (e.g. two -u flags will additionally count hidden files and dirs):

```
loc {{[-u|--unrestricted]}}
```

local

Declare local variables and give them attributes.

See also: **`declare`**, **`export`**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-local.

- Declare a string variable with the specified value:

```
local {{variable}}="{{value}}"
```

- Declare an integer variable with the specified value:

```
local -i {{variable}}="{{value}}"
```

- Declare an array variable with the specified value:

```
local {{variable}}=({{item_a item_b item_c}})
```

- Declare an associative array variable with the specified value:

```
local -A {{variable}}=({{[key_a]=item_a [key_b]=item_b [key_c]=item_c}})
```

- Declare a readonly variable with the specified value:

```
local -r {{variable}}="{{value}}"
```

- Display help:

```
local --help
```

locust

Load-testing tool to determine number of concurrent users a system can handle.

More information: https://locust.io.

- Load-test "example.com" with web interface using locustfile.py:

```
locust --host={{http://example.com}}
```

- Use a different test file:

```
locust --locustfile={{test_file.py}} --host={{http://example.com}}
```

- Run test without web interface, spawning 1 user a second until there are 100 users:

```
locust --no-web --clients={{100}} --hatch-rate={{1}} --host={{http://example.com}}
```

- Start Locust in master mode:

```
locust --master --host={{http://example.com}}
```

- Connect Locust slave to master:

```
locust --slave --host={{http://example.com}}
```

- Connect Locust slave to master on a different machine:

```
locust --slave --master-host={{master_hostname}} --host={{http://example.com}}
```

loft

Install and manage multi-tenant Kubernetes environments using virtual clusters.

More information: https://loft.sh/docs/cli/loft/.

- Install or upgrade Loft in the current Kubernetes cluster:

```
loft start
```

- Authenticate to a remote Loft instance:

```
loft login {{https://loft.example.com}}
```

- Create a virtual cluster with a specific space and cluster:

```
loft create vcluster {{vcluster_name}} {{[-s|--space]}}
{{space_name}} {{[-c|--cluster]}} {{cluster_name}}
```

- List all virtual clusters:

```
loft list vclusters
```

- Switch context to a specific virtual cluster:

```
loft use vcluster {{vcluster_name}}
```

- Delete a virtual cluster:

```
loft delete vcluster {{vcluster_name}}
```

- Show the current Loft username:

```
loft vars username
```

- Uninstall Loft from the cluster:

```
loft uninstall
```

logger

Add messages to syslog.

More information: https://manned.org/logger.1p.

- Log a message to syslog:

```
logger {{message}}
```

logname

Shows the user's login name.

More information: https://www.gnu.org/software/coreutils/manual/html_node/logname-invocation.html.

- Display the currently logged in user's name:

```
logname
```

logout

Exit a login shell.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-logout.

- Exit a login shell:

```
logout
```

- Exit a login shell and specify a return value:

```
logout {{exit_code}}
```

logstash

An Elasticsearch ETL (extract, transform and load) tool.

Commonly used to load data from various sources (such as databases and log files) into Elasticsearch.

More information: https://www.elastic.co/products/logstash.

- Check validity of a Logstash configuration:

```
logstash --configtest --config {{logstash_config.conf}}
```

- Run Logstash using configuration:

```
sudo logstash --config {{logstash_config.conf}}
```

- Run Logstash with the most basic inline configuration string:

```
sudo logstash -e 'input {} filter {} output {}'
```

lolcat

Put a rainbow in everything you **`cat`** to the console.

More information: https://github.com/busyloop/lolcat.

- Print a file to the console in rainbow colors:

```
lolcat {{path/to/file}}
```

- Print the result of a text-producing command in rainbow colors:

```
{{fortune}} | lolcat
```

- Print a file to the console with animated rainbow colors:

```
lolcat {{[-a|--animate]}} {{path/to/file}}
```

- Print a file to the console with 24-bit (truecolor) rainbow colors:

```
lolcat {{[-t|--truecolor]}} {{path/to/file}}
```

look

Display lines beginning with a prefix in a sorted file.

Note: The lines in the file must be sorted.

See also: **`grep`**, **`sort`**.

More information: https://man.openbsd.org/look.

- Search for lines beginning with a specific prefix in a specific file:

```
look {{prefix}} {{path/to/file}}
```

- Case-insensitively search only on alphanumeric characters:

```
look {{[-f|--ignore-case]}} {{[-d|--alphanum]}} {{prefix}} {{path/to/file}}
```

- Specify a string termination character (space by default):

```
look {{[-t|--terminate]}} {{,}}
```

- Search in `/usr/share/dict/words` (`--alphanum` and `--ignore-case` are assumed):

```
look {{prefix}}
```

lorem

Create some random Latin looking placeholder text.

More information: https://manned.org/lorem.

- Print the specified number of words:

```
lorem -n {{20}}
```

- Print 10 lines of Goethe's Faust:

```
lorem -l {{10}} --faust
```

- Print 5 sentences of Poe's Raven:

```
lorem -s {{5}} --raven
```

- Print 40 random characters from Boccaccio's Decameron:

```
lorem --randomize -c {{40}} --decamerone
```

lp

Print files.

More information: https://manned.org/lp.

- Print the output of a command to the default printer (see `lpstat` command):

```
echo "test" | lp
```

- Print a file to the default printer:

```
lp {{path/to/file}}
```

- Print a file to a named printer (see `lpstat` command):

```
lp -d {{printer_name}} {{path/to/file}}
```

- Print `n` copies of a file to the default printer:

```
lp -n {{n}} {{path/to/file}}
```

- Print only certain pages to the default printer (print pages 1, 3-5, and 16):

```
lp -P 1,3-5,16 {{path/to/file}}
```

- Resume printing a job:

```
lp -i {{job_id}} -H resume
```

lpadmin

Configure CUPS printers and classes.

See also: **`lpoptions`**.

More information: https://openprinting.github.io/cups/doc/man-lpadmin.html.

- Set the default printer:

```
lpadmin -d {{printer}}
```

- Delete a specific printer or class:

```
lpadmin -x {{printer|class}}
```

- Add a printer to a class:

```
lpadmin -p {{printer}} -c {{class}}
```

- Remove a printer from a class:

```
lpadmin -p {{printer}} -r {{class}}
```

lpass

Interface for the LastPass password manager.

More information: https://lastpass.github.io/lastpass-cli/lpass.1.html.

- Log in to your LastPass account, by entering your master password when prompted:

```
lpass login {{username}}
```

- Show login status:

```
lpass status
```

- List all sites grouped by category:

```
lpass ls
```

- Generate a new password for gmail.com with the identifier `myinbox` and add to LastPass:

```
lpass generate --username {{username}} --url {{gmail.com}} {{myinbox}} {{password_length}}
```

- Show password for a specified entry:

```
lpass show {{myinbox}} --password
```

lpinfo

List connected printers and installed drivers for the CUPS print server.

More information: https://openprinting.github.io/cups/doc/man-lpinfo.html.

- List all the currently connected printers:

```
lpinfo -v
```

- List all the currently installed printer drivers:

```
lpinfo -m
```

- Search installed printer drivers by make and model:

```
lpinfo --make-and-model "{{printer_model}}" -m
```

lpmove

Move a job or all jobs to another printer.

See also: **`cancel`**, **`lp`**, **`lpr`**, **`lprm`**.

More information: https://openprinting.github.io/cups/doc/man-lpmove.html.

- Move a specific job to `new_printer`:

```
lpmove {{job_id}} {{new_printer}}
```

- Move a job from `old_printer` to `new_printer`:

```
lpmove {{old_printer}}-{{job_id}} {{new_printer}}
```

- Move all jobs from `old_printer` to `new_printer`:

```
lpmove {{old_printer}} {{new_printer}}
```

- Move a specific job to `new_printer` on a specific server:

```
lpmove -h {{server}} {{job_id}} {{new_printer}}
```

lpoptions

Display or set printer options and defaults.

See also: **`lpadmin`**.

More information: https://openprinting.github.io/cups/doc/man-lpoptions.html.

- Set the default printer:

```
lpoptions -d {{printer}}/{{instance}}
```

- List printer-specific options of a specific printer:

```
lpoptions -d {{printer}} -l
```

- Set a new option on a specific printer:

```
lpoptions -d {{printer}} -o {{option}}
```

- Remove the options of a specific printer:

```
lpoptions -d {{printer}} -x
```

lpq

Show printer queue status.

More information: https://openprinting.github.io/cups/doc/man-lpq.html.

- Show the queued jobs of the default destination:

```
lpq
```

- Show the queued jobs of all printers enforcing encryption:

```
lpq -a -E
```

- Show the queued jobs in a long format:

```
lpq -l
```

- Show the queued jobs of a specific printer or class:

```
lpq -P {{destination}}/{{instance}}
```

- Show the queued jobs once every n seconds until the queue is empty:

```
lpq +{{interval}}
```

lpr

Print files.

See also: **`lpstat`**, **`lpadmin`**.

More information: https://openprinting.github.io/cups/doc/man-lpr.html.

- Print a file to the default printer:

```
lpr {{path/to/file}}
```

- Print 2 copies:

```
lpr -# {{2}} {{path/to/file}}
```

- Print to a named printer:

```
lpr -P {{printer}} {{path/to/file}}
```

- Print either a single page (e.g. 2) or a range of pages (e.g. 2-16):

```
lpr -o page-ranges={{2|2-16}} {{path/to/file}}
```

- Print double-sided either in portrait (long) or in landscape (short):

```
lpr -o sides={{two-sided-long-edge|two-sided-short-edge}}
{{path/to/file}}
```

- Set page size (more options may be available depending on setup):

```
lpr -o media={{a4|letter|legal}} {{path/to/file}}
```

- Print multiple pages per sheet:

```
lpr -o number-up={{2|4|6|9|16}} {{path/to/file}}
```

lprm

Cancel queued print jobs of a server.

See also: **`lpq`**.

More information: https://openprinting.github.io/cups/doc/man-lprm.html.

- Cancel current job on the default printer:

```
lprm
```

- Cancel a job of a specific server:

```
lprm -h {{server}} {{job_id}}
```

- Cancel a job of a specific server in a specific port:

```
lprm -h {{server}}:{{port}} {{job_id}}
```

- Cancel multiple jobs with a encrypted connection to the server:

```
lprm -E {{job_id1 job_id2 ...}}
```

- Cancel all jobs:

```
lprm -
```

- Cancel the current job of a specific printer or class:

```
lprm -P {{destination}}/{{instance}}
```

lpstat

Show status information about printers.

More information: https://manned.org/lpstat.

- List printers present on the machine and whether they are enabled for printing:

```
lpstat -p
```

- Show the default printer:

```
lpstat -d
```

- Display all available status information:

```
lpstat -t
```

- List print jobs queued by a specific user:

```
lpstat -u {{user}}
```

ls

List directory contents.

More information: https://www.gnu.org/software/coreutils/manual/html_node/ls-invocation.html.

- List files one per line:

```
ls -1
```

- List all files, including hidden files:

```
ls {{[-a|--all]}}
```

- List files with a trailing symbol to indicate file type (directory/, symbolic_link@, executable*, ...):

```
ls {{[-F|--classify]}}
```

- List all files in [l]ong format (permissions, ownership, size, and modification date):

```
ls {{[-la|-l --all]}}
```

- List files in [l]ong format with size displayed using human-readable units (KiB, MiB, GiB):

```
ls {{[-lh|-l --human-readable]}}
```

- List files in [l]ong format, sorted by [S]ize (descending) recursively:

```
ls {{[-lSR|-lS --recursive]}}
```

- List files in [l]ong format, sorted by [t]ime the file was modified and in reverse order (oldest first):

```
ls {{[-ltr|-lt --reverse]}}
```

- Only list directories:

```
ls {{[-d|--directory]}} */
```

lsar

List an archive file's contents.

See also: **`unar`**, **`ar`**.

More information: https://manned.org/lsar.

- List an archive file's contents:

```
lsar {{path/to/archive}}
```

- List a password protected archive file's contents:

```
lsar {{path/to/archive}} {{[-p|--password]}} {{password}}
```

- Print all available information about each file in the archive (it's very long):

```
lsar {{[-L|--verylong]}} {{path/to/archive}}
```

- Test the integrity of the files in the archive (if possible):

```
lsar {{[-t|--test]}} {{path/to/archive}}
```

- List the archive file's contents in JSON format:

```
lsar {{[-j|--json]}} {{path/to/archive}}
```

- Display help:

```
lsar {{[-h|--help]}}
```

lsd

List directory contents.

The next generation **`ls`** command, written in Rust.

More information: https://github.com/Peltoche/lsd.

- List files and directories, one per line:

```
lsd {{[-1|--oneline]}}
```

- List all files and directories, including hidden ones, in the current directory:

```
lsd {{[-a|--all]}}
```

- List files and directories with trailing `/` added to directory names:

```
lsd {{[-F|--classify]}}
```

- List all files and directories in long format (permissions, ownership, size in human-readable format, and modification date):

```
lsd {{[-lha|--long --human-readable --all]}}
```

- List files and directories in long format, sorted by size (descending):

```
lsd {{[-lS|--long --sizesort]}}
```

- List files and directories in long format, sorted by modification date (oldest first):

```
lsd {{[-ltr|--long --timesort --reverse]}}
```

- Only list directories:

```
lsd {{[-d|--directory-only]}} {{*/}}
```

- Recursively list all directories in a tree format:

```
lsd --tree {{[-d|--directory-only]}}
```

lsix

List and display images in sixel format on the terminal.

See also: **`img2sixel`**.

More information: https://github.com/hackerb9/lsix.

- List images in the current directory:

```
lsix
```

- List images in a specific directory:

```
lsix {{path/to/directory}}
```

- Display an image on the terminal:

```
lsix {{path/to/image}}
```

lsof

List open files and the corresponding processes.

Note: Root privileges are required to list files opened by others.

More information: https://manned.org/lsof.

- Find the processes that have a given file open:

```
lsof {{path/to/file}}
```

- Find the process that opened a local internet port:

```
lsof -i :{{port}}
```

- Only output the process ID (PID):

```
lsof -t {{path/to/file}}
```

- List files opened by the given user:

```
lsof -u {{username}}
```

- List files opened by the given command or process:

```
lsof -c {{process_or_command_name}}
```

- List files opened by a specific process, given its PID:

```
lsof -p {{PID}}
```

- List open files in a directory:

```
lsof +D {{path/to/directory}}
```

- Find the process that is listening on a local IPv6 TCP port and don't convert network or port numbers:

```
lsof -i6TCP:{{port}} -sTCP:LISTEN -n -P
```

lspath

List the contents of the **PATH** environment variable, with optional paging.

More information: https://github.com/clementi/lspath.

- Print the contents of the system PATH variable, with one element per line:

```
lspath
```

- Print the current contents of the system PATH variable, with one element per line, with the output paged:

```
lspath --page
```

lstopo

Show the hardware topology of the system.

More information: https://manned.org/lstopo.

- Show the summarized system topology in a graphical window (or print to console if no graphical display is available):

```
lstopo
```

- Show the full system topology without summarizations:

```
lstopo --no-factorize
```

- Show the summarized system topology with only physical indices (i.e. as seen by the OS):

```
lstopo {{[-p|--physical]}}
```

- Write the full system topology to a file in the specified format:

```
lstopo --no-factorize {{[--of|--output-format]}} {{console|ascii|tex|fig|svg|pdf|ps|png|xml}} {{path/to/file}}
```

- Output in monochrome or greyscale:

```
lstopo --palette {{none|grey}}
```

lsyncd

Watch files and directories and run **`rsync`** when they change.

It is often used to keep two directories on separate systems in sync, ensuring that changes made in one directory are immediately mirrored to the other.

More information: https://github.com/lsyncd/lsyncd.

- Watch the source for changes and run `rsync` to synchronize files to the destination on every change:

```
lsyncd -rsync {{path/to/source}} {{host::share_name}}
```

- Use SSH instead of `rsyncd` shares:

```
lsyncd -rsyncssh {{path/to/source}} {{host}} {{path/to/destination}}
```

lt

Localtunnel exposes your localhost to the world for easy testing and sharing.

More information: https://github.com/localtunnel/localtunnel.

- Start tunnel from a specific port:

```
lt {{[-p|--port]}} {{8000}}
```

- Specify the upstream server doing the forwarding:

```
lt {{[-p|--port]}} {{8000}} {{[-h|--host]}} {{host}}
```

- Request a specific subdomain:

```
lt {{[-p|--port]}} {{8000}} {{[-s|--subdomain]}} {{subdomain}}
```

- Print basic request info:

```
lt {{[-p|--port]}} {{8000}} --print-requests
```

- Open the tunnel URL in the default web browser:

```
lt {{[-p|--port]}} {{8000}} {{[-o|--open]}}
```

lua

A powerful, light-weight embeddable programming language.

More information: https://www.lua.org/manual/5.4/lua.html.

- Start an interactive Lua shell:

```
lua
```

- Execute a Lua script:

```
lua {{path/to/script.lua}} {{--optional-argument}}
```

- Execute a Lua expression:

```
lua -e '{{print("Hello World")}}'
```

luac

Lua bytecode compiler.

More information: https://www.lua.org/manual/5.4/luac.html.

- Compile a Lua source file to Lua bytecode:

```
luac -o {{byte_code.luac}} {{source.lua}}
```

- Do not include debug symbols in the output:

```
luac -s -o {{byte_code.luac}} {{source.lua}}
```

luajit

A just-in-time compiler (JIT) for the Lua programming language.

See also: **`lua`**.

More information: https://luajit.org/luajit.html.

- Start an interactive Lua shell:

```
luajit
```

- Execute a Lua script:

```
luajit {{path/to/script.lua}} {{--optional-argument}}
```

- Execute a Lua expression:

```
luajit -e '{{print("Hello World")}}'
```

lualatex

An extended version of TeX using Lua to compile.

More information: https://manned.org/lualatex.1.

- Start `texlua` to act as a Lua interpreter:

```
lualatex
```

- Compile a Tex file to PDF:

```
lualatex {{path/to/file.tex}}
```

- Compile a Tex file without error interruption:

```
lualatex -interaction nonstopmode {{path/to/file.tex}}
```

- Compile a Tex file with a specific output file name:

```
lualatex -jobname={{filename}} {{path/to/file.tex}}
```

ludusavi

Backup video game save data.

More information: https://github.com/mtkennerly/ludusavi/blob/master/docs/cli.md.

- Backup games:

```
ludusavi backup --path {{path/to/backup}}
```

- Restore games:

```
ludusavi restore --path {{path/to/backup}} {{"game1" "game2" ...}}
```

- List backups:

```
ludusavi backups --path {{path/to/backup}}
```

- Wrap launcher game:

```
ludusavi wrap --gui --infer {{heroic|lutris|steam}} -- {{game_launch_commands}}
```

- Wrap standalone game:

```
ludusavi wrap --gui --name {{name}} -- {{game_launch_commands}}
```

lumen

An installer for the Lumen micro-framework.

More information: https://lumen.laravel.com.

- Create a new Lumen application:

  ```
  lumen new {{application_name}}
  ```

- List the available installer commands:

  ```
  lumen list
  ```

lwp-request

Simple HTTP client.

Built with libwww-perl.

More information: https://metacpan.org/pod/lwp-request.

- Make a simple GET request:

```
lwp-request -m GET {{http://example.com/some/path}}
```

- Upload a file with a POST request:

```
lwp-request -m POST {{http://example.com/some/path}} <
{{path/to/file}}
```

- Make a request with a custom user agent:

```
lwp-request -H 'User-Agent: {{user_agent}} -m {{METHOD}}
{{http://example.com/some/path}}
```

- Make a request with HTTP authentication:

```
lwp-request -C {{username}}:{{password}} -m {{METHOD}}
{{http://example.com/some/path}}
```

- Make a request and print request headers:

```
lwp-request -U -m {{METHOD}} {{http://example.com/some/path}}
```

- Make a request and print response headers and status chain:

```
lwp-request -E -m {{METHOD}} {{http://example.com/some/path}}
```

lychee

Find broken URLs.

More information: https://github.com/lycheeverse/lychee/blob/master/README.md#commandline-usage.

- Scan a website for broken links:

```
lychee {{https://example.com}}
```

- Display a breakdown of error types:

```
lychee --format detailed {{https://example.com}}
```

- Limit the amount of connections to prevent DDOS protection:

```
lychee --max-concurrency {{5}} {{links.txt}}
```

- Check files in a directory structure for any broken URLs:

```
grep {{[-r|--recursive]}} "{{pattern}}" | lychee -
```

- Display help:

```
lychee --help
```

lynx

Command-line web browser.

More information: https://lynx.browser.org.

- Visit a website:

```
lynx {{example.com}}
```

- Apply restrictions for anonymous account:

```
lynx -anonymous {{example.com}}
```

- Turn on mouse support, if available:

```
lynx -use_mouse {{example.com}}
```

- Force color mode on, if available:

```
lynx -color {{example.com}}
```

- Open a link, using a specific file to read and write cookies:

```
lynx -cookie_file={{path/to/file}} {{example.com}}
```

- Navigate forwards and backwards through the links on a page:

```
{{<ArrowUp>|<ArrowDown>}}
```

- Go back to the previously displayed page:

```
{{<ArrowLeft>|<u>}}
```

- Exit:

```
<q><y>
```

lz4

Compress or decompress .lz4 files.

More information: https://github.com/lz4/lz4.

- Compress a file:

```
lz4 {{path/to/file}}
```

- Decompress a file:

```
lz4 {{[-d|--decompress]}} {{file.lz4}}
```

- Decompress a file and write to `stdout`:

```
lz4 {{[-dc|--decompress --stdout]}} {{file.lz4}}
```

- Package and compress a directory and its contents:

```
tar cvf - {{path/to/directory}} | lz4 - {{dir.tar.lz4}}
```

- Decompress and unpack a directory and its contents:

```
lz4 {{[-dc|--decompress --stdout]}} {{dir.tar.lz4}} | tar -xv
```

- Compress a file using the best compression:

```
lz4 {{[-12|--best]}} {{path/to/file}}
```

lzcat

This command is an alias of **`xz --format=lzma --decompress --stdout`**.

- View documentation for the original command:

  ```
  tldr xz
  ```

lzcmp

This command is an alias of **`xzcmp`**.

- View documentation for the original command:

  ```
  tldr xzcmp
  ```

lzegrep

This command is an alias of **`xzgrep --extended-regexp`**.

See also: **`egrep`**.

- View documentation for the original command:

  ```
  tldr xzgrep
  ```

lzfgrep

This command is an alias of **`xzgrep --fixed-strings`**.

See also: **`fgrep`**.

- View documentation for the original command:

  ```
  tldr xzgrep
  ```

lzgrep

This command is an alias of **`xzgrep`**.

- View documentation for the original command:

```
tldr xzgrep
```

lzip

A lossless data compressor with a user interface similar to **`gzip`** or **`bzip2`**.

Lzip uses a simplified form of the "Lempel-Ziv-Markovchain-Algorithm" (LZMA) stream format and provides 3-factor integrity checking to maximize interoperability and optimize safety.

More information: https://www.nongnu.org/lzip/manual/lzip_manual.html#Invoking-lzip.

- Archive a file, replacing it with with a compressed version:

```
lzip {{path/to/file}}
```

- Archive a file, keeping the input file:

```
lzip {{[-k|--keep]}} {{path/to/file}}
```

- Archive a file with the best compression (level=9):

```
lzip {{[-k|--keep]}} {{path/to/file}} --best
```

- Archive a file at the fastest speed (level=0):

```
lzip {{[-k|--keep]}} {{path/to/file}} --fast
```

- Test the integrity of compressed file:

```
lzip {{[-t|--test]}} {{path/to/archive.lz}}
```

- Decompress a file, replacing it with the original uncompressed version:

```
lzip {{[-d|--decompress]}} {{path/to/archive.lz}}
```

- Decompress a file, keeping the archive:

```
lzip {{[-d|--decompress]}} {{[-k|--keep]}} {{path/to/archive.lz}}
```

- List files which are in an archive and show compression stats:

```
lzip {{[-l|--list]}} {{path/to/archive.lz}}
```

lzless

This command is an alias of **`xzless`**.

- View documentation for the original command:

```
tldr xzless
```

lzma

This command is an alias of **`xz --format=lzma`**.

- View documentation for the original command:

```
tldr xz
```

lzmore

This command is an alias of **xzmore**.

- View documentation for the original command:

```
tldr xzmore
```

lzop

Compress or decompress files with LZO compression.

More information: https://www.lzop.org/lzop_man.php.

- Compress a file into a new file with the `.lzo` suffix:

```
lzop {{path/to/file}}
```

- Decompress a file:

```
lzop {{[-d|--decompress]}} {{path/to/file.lzo}}
```

- Compress a file, while specifying the compression level. 0 = Worst, 9 = Best (Default level is 3):

```
lzop -{{level}} {{path/to/file}}
```

- Compress a file with the best compression level:

```
lzop {{[-9|--best]}} {{path/to/file}}
```

- Compress a file with the fastest compression level:

```
lzop {{[-1|--fast]}} {{path/to/file}}
```

m4

Macro processor.

More information: https://www.gnu.org/software/m4/manual/m4.html#Invoking-m4.

- Process macros in a file:

```
m4 {{path/to/file}}
```

- Define a macro before processing files:

```
m4 {{[-D|--define]}} {{macro_name}}={{macro_value}} {{path/to/file}}
```

m4b-tool

Merge, split, and manipulate audiobook files with chapters.

More information: https://github.com/sandreas/m4b-tool.

- Create an audiobook with the audio files in the input directory:

```
m4b-tool merge {{path/to/input_directory}} --output-file={{path/to/merged.m4b}}
```

- Make chapters using the input files' names:

```
m4b-tool merge {{path/to/input_directory}} --output-file={{path/to/merged.m4b}} --use-filenames-as-chapters
```

macchina

Display information about your computer.

More information: https://github.com/Macchina-CLI/macchina.

- List out system information, with either default settings or those specified in your configuration file:

```
macchina
```

- Specify a custom configuration file path:

```
macchina --config {{path/to/configuration_file}}
```

- List system information, but lengthen uptime, shell and kernel output:

```
macchina --long-uptime --long-shell --long-kernel
```

- Check for any errors/system failures encountered when trying to fetch system information:

```
macchina --doctor
```

- List original artists of all the ASCII art:

```
macchina --ascii-artists
```

macptopbm

Read a MacPaint file as input and produce a PBM image as output.

See also: **pbmtomacp**.

More information: https://netpbm.sourceforge.net/doc/macptopbm.html.

- Convert a MacPaint file into a PGM image:

```
macptopbm {{path/to/file.macp}} > {{path/to/output.pbm}}
```

- Skip over n bytes when reading the file:

```
macptopbm {{[-e|-extraskip]}} {{n}} > {{path/to/output.pbm}}
```

- Suppress all informational messages:

```
macptopbm {{[-q|-quiet]}} > {{path/to/output.pbm}}
```

- Display version:

```
macptopbm {{[-v|-version]}}
```

maestral

A lightweight Dropbox client for macOS and Linux.

More information: https://maestral.app/cli.

- Start the GUI:

```
maestral gui
```

- Print current status of Maestral:

```
maestral status
```

- Pause syncing:

```
maestral pause
```

- Resume syncing:

```
maestral resume
```

- Print sync status of a specific file or folder:

```
maestral filestatus {{path/to/file_or_directory}}
```

magento

Manage the Magento PHP framework.

More information: https://experienceleague.adobe.com/en/docs/commerce-operations/tools/cli-reference/commerce-on-premises.

- Enable one or more modules:

  ```
  magento module:enable {{module1 module2 ...}}
  ```

- Disable one or more modules:

  ```
  magento module:disable {{module1 module2 ...}}
  ```

- Update the database after enabling modules:

  ```
  magento setup:upgrade
  ```

- Update code and dependency injection configuration:

  ```
  magento setup:di:compile
  ```

- Deploy static assets:

  ```
  magento setup:static-content:deploy
  ```

- Enable maintenance mode:

  ```
  magento maintenance:enable
  ```

- Disable maintenance mode:

  ```
  magento maintenance:disable
  ```

- List all available commands:

  ```
  magento list
  ```

magick compare

Create a comparison image to visually annotate the difference between two images.

See also: **magick**.

More information: https://imagemagick.org/script/compare.php.

- Compare two images:

```
magick compare {{path/to/image1.png}} {{path/to/image2.png}} {{path/to/diff.png}}
```

- Compare two images using the specified metric:

```
magick compare -verbose -metric {{PSNR}} {{path/to/image1.png}} {{path/to/image2.png}} {{path/to/diff.png}}
```

magick convert

Convert between image formats, scale, join, and create images, and much more.

Note: This tool (previously **`convert`**) has been replaced by **`magick`** in ImageMagick 7+.

More information: https://imagemagick.org/script/convert.php.

- Convert an image from JPEG to PNG:

```
magick convert {{path/to/input_image.jpg}} {{path/to/output_image.png}}
```

- Scale an image to 50% of its original size:

```
magick convert {{path/to/input_image.png}} -resize 50% {{path/to/output_image.png}}
```

- Scale an image keeping the original aspect ratio to a maximum dimension of 640x480:

```
magick convert {{path/to/input_image.png}} -resize 640x480 {{path/to/output_image.png}}
```

- Scale an image to have a specified file size:

```
magick convert {{path/to/input_image.png}} -define jpeg:extent={{512kb}} {{path/to/output_image.jpg}}
```

- Vertically/horizontally append images and have the empty space be transparent:

```
magick convert -background none {{path/to/image1.png path/to/image2.png ...}} {{-append|+append}} {{path/to/output_image.png}}
```

- Create a GIF from a series of images with 100ms delay between them:

```
magick convert {{path/to/image1.png path/to/image2.png ...}} -delay {{10}} {{path/to/animation.gif}}
```

- Create an image with nothing but a solid red background:

```
magick convert -size {{800x600}} "xc:{{#ff0000}}" {{path/to/image.png}}
```

- Create a favicon from several images of different sizes:

```
magick convert {{path/to/image1.png path/to/image2.png ...}}
{{path/to/favicon.ico}}
```

magick identify

Describe the format and characteristics of image files.

See also: **magick**.

More information: https://imagemagick.org/script/identify.php.

- Describe the format and basic characteristics of an image:

```
magick identify {{path/to/image}}
```

- Describe the format and verbose characteristics of an image:

```
magick identify -verbose {{path/to/image}}
```

- Collect dimensions of all JPEG files in the current directory and save them into a CSV file:

```
magick identify -format "{{%f,%w,%h\n}}" {{*.jpg}} > {{path/to/filelist.csv}}
```

magick import

Capture some or all of an X server screen and save the image to a file.

See also: **`magick`**.

More information: https://imagemagick.org/script/import.php.

- Capture the entire X server screen into a PostScript file:

```
magick import -window root {{path/to/output.ps}}
```

- Capture contents of a remote X server screen into a PNG image:

```
magick import -window root -display {{remote_host}}:
{{screen}}.{{display}} {{path/to/output.png}}
```

- Capture a specific window given its ID as displayed by `xwininfo` into a JPEG image:

```
magick import -window {{window_id}} {{path/to/output.jpg}}
```

magick mogrify

Perform operations on multiple images, such as resizing, cropping, flipping, and adding effects.

Changes are applied directly to the original file.

See also: **magick**.

More information: https://imagemagick.org/script/mogrify.php.

- Resize all JPEG images in the directory to 50% of their initial size:

```
magick mogrify -resize {{50%}} {{*.jpg}}
```

- Resize all images starting with DSC to 800x600:

```
magick mogrify -resize {{800x600}} {{DSC*}}
```

- Convert all PNGs in the directory to JPEG:

```
magick mogrify -format {{jpg}} {{*.png}}
```

- Halve the saturation of all image files in the current directory:

```
magick mogrify -modulate {{100,50}} {{*}}
```

- Double the brightness of all image files in the current directory:

```
magick mogrify -modulate {{200}} {{*}}
```

- Reduce file sizes of all GIF images in the current directory by reducing quality:

```
magick mogrify -layers 'optimize' -fuzz {{7%}} {{*.gif}}
```

- Display help:

```
magick mogrify -help
```

magick montage

Tile images into a customizable grid.

See also: **magick**.

More information: https://imagemagick.org/script/montage.php.

- Tile images into a grid, automatically resizing images larger than the grid cell size:

```
magick montage {{path/to/image1.jpg path/to/image2.jpg ...}} {{path/to/montage.jpg}}
```

- Tile images into a grid, automatically calculating the grid cell size from the largest image:

```
magick montage {{path/to/image1.jpg path/to/image2.jpg ...}} -geometry {{+0+0}} {{path/to/montage.jpg}}
```

- Specify the grid cell size and resize images to fit it before tiling:

```
magick montage {{path/to/image1.jpg path/to/image2.jpg ...}} -geometry {{640x480+0+0}} {{path/to/montage.jpg}}
```

- Limit the number of rows and columns in the grid, causing input images to overflow into multiple output montages:

```
magick montage {{path/to/image1.jpg path/to/image2.jpg ...}} -geometry {{+0+0}} -tile {{2x3}} {{montage_%d.jpg}}
```

- Resize and crop images to fill their grid cells before tiling:

```
magick montage {{path/to/image1.jpg path/to/image2.jpg ...}} -geometry {{+0+0}} -resize {{640x480^}} -gravity {{center}} -crop {{640x480+0+0}} {{path/to/montage.jpg}}
```

magick

Create, edit, compose, or convert between image formats.

This tool replaces **`convert`** in ImageMagick 7+. See **`magick convert`** to use the old tool in versions 7+.

Some subcommands, such as **`mogrify`** have their own usage documentation.

More information: https://imagemagick.org.

- Convert between image formats:

```
magick {{path/to/input_image.png}} {{path/to/
output_image.jpg}}
```

- Resize an image, making a new copy:

```
magick {{path/to/input_image.jpg}} -resize {{100x100}}
{{path/to/output_image.jpg}}
```

- Resize an image by a percentage:

```
magick {{path/to/input_image.png}} -resize {{50}}% {{path/to/
output_image.png}}
```

- Scale an image to have a specified file size:

```
magick {{path/to/input_image.png}} -define
jpeg:extent={{512kb}} {{path/to/output_image.jpg}}
```

- Create a GIF out of all JPEG images in the current directory:

```
magick {{*.jpg}} {{path/to/images.gif}}
```

- Create a checkerboard pattern:

```
magick -size {{640x480}} pattern:checkerboard {{path/to/
checkerboard.png}}
```

- Create a PDF file out of all JPEG images in the current directory:

```
magick {{*.jpg}} -adjoin {{path/to/file.pdf}}
```

mail

Operate on the user's mailbox.

To send an email the message body is built from **`stdin`**.

More information: https://manned.org/mail.

- Open an interactive prompt to check personal mail:

```
mail
```

- Send a typed email message with optional CC. The command-line below continues after pressing `<Enter>`. Input message text (can be multiline). Press `<Ctrl d>` to complete the message text:

```
mail --subject "{{subject line}}" {{to_user@example.com}} --cc "{{cc_email_address}}"
```

- Send an email that contains file content:

```
mail < {{path/to/file.txt}} --subject "{{$HOSTNAME filename.txt}}" {{to_user@example.com}}
```

- Send a `tar.gz` file as an attachment:

```
tar cvzf - {{path/to/directory1 path/to/directory2}} | uuencode {{data.tar.gz}} | mail --subject "{{subject_line}}" {{to_user@example.com}}
```

- Display help:

```
mail {{[-h|--help]}}
```

mailsy

Quickly generate a disposable email using **`mail.tm`** API.

More information: https://github.com/BalliAsghar/Mailsy.

- Create an email address:

  `mailsy g`

- Fetch emails (press `<Enter>` to open email in browser):

  `mailsy m`

- Show details about the account:

  `mailsy me`

- Dispose the email address:

  `mailsy d`

mailx

Send and receive mail.

More information: https://manned.org/mailx.

- Send mail (the content should be typed after the command, and ended with <Ctrl d>):

```
mailx {{[-s|--subject]}} "{{subject}}" {{to_addr}}
```

- Send mail with content passed from another command:

```
echo "{{content}}" | mailx {{[-s|--subject]}}
"{{subject}}" {{to_addr}}
```

- Send mail with content read from a file:

```
mailx < {{content.txt}} {{[-s|--subject]}}
"{{subject}}" {{to_addr}}
```

- Send mail to a recipient and CC to another address:

```
mailx {{[-s|--subject]}} "{{subject}}" {{[-c|--cc]}}
{{cc_addr}} {{to_addr}}
```

- Send mail specifying the sender address:

```
mailx {{[-s|--subject]}} "{{subject}}" {{[-r|--from-
address]}} {{from_addr}} {{to_addr}}
```

- Send mail with an attachment:

```
mailx {{[-a|--attach]}} {{path/to/file}} {{[-s|--subject]}}
"{{subject}}" {{to_addr}}
```

make

Task runner for targets described in Makefile.

Mostly used to control the compilation of an executable from source code.

More information: https://www.gnu.org/software/make/manual/make.html.

- Call the first target specified in the Makefile (usually named "all"):

```
make
```

- Call a specific target:

```
make {{target}}
```

- Call a specific target, executing 4 jobs at a time in parallel:

```
make {{[-j|--jobs]}} 4 {{target}}
```

- Use a specific Makefile:

```
make {{[-f|--file]}} {{path/to/file}}
```

- Execute make from another directory:

```
make {{[-C|--directory]}} {{path/to/directory}}
```

- Force making of a target, even if source files are unchanged:

```
make {{[-B|--always-make]}} {{target}}
```

- Override a variable defined in the Makefile:

```
make {{target}} {{variable}}={{new_value}}
```

- Override variables defined in the Makefile by the environment:

```
make {{[-e|--environment-overrides]}} {{target}}
```

makebuildserver

Create an F-Droid build server virtual machine.

More information: https://f-droid.org/en/docs/Build_Server_Setup/.

- Create a new virtual machine or update an existing one (if available):

```
makebuildserver
```

- Force creating a fresh virtual machine:

```
makebuildserver {{[-c|--clean]}}
```

makensis

Cross-platform compiler for NSIS installers.

It compiles a NSIS script into a Windows installer executable.

More information: https://nsis.sourceforge.io/Docs/Chapter3.html.

- Compile a NSIS script:

```
makensis {{path/to/file.nsi}}
```

- Compile a NSIS script in strict mode (treat warnings as errors):

```
makensis -WX {{path/to/file.nsi}}
```

- Display help for a specific command:

```
makensis -CMDHELP {{command}}
```

makepasswd

Generate and encrypt passwords.

More information: https://manpages.debian.org/latest/makepasswd/makepasswd.1.en.html.

- Generate a random password (8 to 10 characters long, containing letters and numbers):

```
makepasswd
```

- Generate a 10 characters long password:

```
makepasswd --chars {{10}}
```

- Generate a 5 to 10 characters long password:

```
makepasswd --minchars {{5}} --maxchars {{10}}
```

- Generate a password containing only the characters "b", "a" or "r":

```
makepasswd --string {{bar}}
```

mamba repoquery

Efficiently query conda and mamba package repositories and package dependencies.

More information: https://mamba.readthedocs.io/en/latest/user_guide/mamba.html#repoquery.

- Search for all available versions of a particular package:

```
mamba repoquery search {{package}}
```

- Search for all packages satisfying specific constraints:

```
mamba repoquery search "{{sphinx<5}}"
```

- List the dependencies of a package installed in the currently activated environment, in a tree format:

```
mamba repoquery depends {{[-t|--tree]}} {{scipy}}
```

- Print packages in the current environment that require a particular package to be installed (i.e. inverse of `depends`):

```
mamba repoquery whoneeds {{ipython}}
```

mamba

Fast, cross-platform package manager, intended as a drop-in replacement for conda.

Some subcommands such as **`repoquery`** have their own usage documentation.

More information: https://mamba.readthedocs.io/en/latest/user_guide/mamba.html.

- Create a new environment, installing the specified packages into it:

```
mamba create {{[-n|--name]}} {{environment_name}} {{python=3.10 matplotlib}}
```

- Install packages into the current environment, specifying the package channel:

```
mamba install {{[-c|--channel]}} {{conda-forge}} {{python=3.6 numpy}}
```

- Update all packages in the current environment:

```
mamba update {{[-a|--all]}}
```

- Search for a specific package across repositories:

```
mamba repoquery search {{numpy}}
```

- List all environments:

```
mamba info {{[-e|--envs]}}
```

- Remove unused packages and tarballs from the cache:

```
mamba clean {{[-pt|--packages --tarballs]}}
```

- Activate an environment:

```
mamba activate {{environment_name}}
```

- List all installed packages in the currently activated environment:

```
mamba list
```

man

Format and display manual pages.

More information: https://manned.org/man.

- Display the man page for a command:

```
man {{command}}
```

- Open the man page for a command in a browser (`BROWSER` environment variable can replace `=browser_name`):

```
man {{[-H|--html=]}}{{browser_name}} {{command}}
```

- Display the man page for a command from section 7:

```
man 7 {{command}}
```

- List all available sections for a command:

```
man {{[-f|--whatis]}} {{command}}
```

- Display the path searched for manpages:

```
man {{[-w|--path]}}
```

- Display the location of a manpage rather than the manpage itself:

```
man {{[-w|--where]}} {{command}}
```

- Display the man page using a specific locale:

```
man {{[-L|--locale]}} {{locale}} {{command}}
```

- Search for manpages containing a search string:

```
man {{[-k|--apropos]}} "{{search_string}}"
```

manim

Animation engine for explanatory math videos.

More information: https://docs.manim.community/en/stable/tutorials/quickstart.html.

- Render a scene from a Python script using the default settings:

```
manim {{path/to/file.py}} {{SceneName}}
```

- Render with live preview (auto-opens the video file after rendering):

```
manim {{[-pql|--preview --quality low]}} {{path/to/file.py}} {{SceneName}}
```

- Render at high quality (1080p 60fps):

```
manim {{[-pqh|--preview --quality high]}} {{path/to/file.py}} {{SceneName}}
```

- Specify a custom output file name:

```
manim {{[-o|--output_file]}} {{output_file_name}} {{path/to/file.py}} {{SceneName}}
```

- Render using a specific resolution and frame rate:

```
manim {{[-r|--resolution]}} {{1920,1080}} {{[-f|--fps]}} {{60}} {{path/to/file.py}} {{SceneName}}
```

- List available scenes in a file without rendering:

```
manim --list_scenes {{path/to/file.py}}
```

- Display help:

```
manim --help
```

mapfile

This command is an alias of **`readarray`**.

- View documentation for the original command:

```
tldr readarray
```

mariadb-install-db

Initialize a MariaDB database.

More information: https://manned.org/mariadb-install-db.

- Initialize a database:

```
sudo mariadb-install-db --user {{user}} --basedir {{/usr}} --datadir {{/var/lib/mysql}}
```

- Display help:

```
mariadb-install-db --help
```

mariadb

The mariadb client tool.

More information: https://mariadb.com/kb/en/mariadb-command-line-client/.

- Connect to a specific MariaDB database:

```
mariadb {{db_name}}
```

- Connect to a specific MariaDB database using username and password:

```
mariadb --user {{user_name}} --password {{your_password}} {{db_name}}
```

- Show warnings after every statement in interactive and batch mode:

```
mariadb --show-warning
```

- Display less verbose outputs (can be used multiple times to produce less output):

```
mariadb {{-s|-ss|-sss|--silent}}
```

- Execute SQL statements from a script file:

```
mariadb < {{path/to/script.sql}} {{db_name}} > {{path/to/output.tab}}
```

- Check memory and open file usage at exit:

```
mariadb --debug-check
```

- Connect using a socket file for local connections:

```
mariadb {{[-S|--socket]}} {{path/to/socket_name}}
```

- Display help:

```
mariadb {{[-?|--help]}}
```

marimo

A reactive Python notebook environment.

Combines features of Jupyter, Streamlit, and other notebook tools with reactive execution.

More information: https://docs.marimo.io/cli.

- Create or edit notebooks by starting a marimo server:

  ```
  marimo edit
  ```

- Start a marimo server on a specific port without launching a browser:

  ```
  marimo edit {{[-p|--port]}} {{port_number}} --headless
  ```

- Edit a specific notebook:

  ```
  marimo edit {{path/to/notebook.py}}
  ```

- Run a marimo notebook as an app in read-only mode:

  ```
  marimo run {{path/to/notebook.py}}
  ```

- Start an interactive tutorial to learn marimo:

  ```
  marimo tutorial {{intro|components|dataflow|io}}
  ```

- View command-specific help:

  ```
  marimo {{edit|run|tutorial|config|new|...}} --help
  ```

markdown2pdf

Convert markdown to PDF.

More information: https://github.com/theiskaa/markdown2pdf.

- Convert a Markdown file to a PDF:

```
markdown2pdf {{[-p|--path]}} {{path/to/input_file.md}}
```

- Convert a Markdown file to a PDF with a specific path:

```
markdown2pdf {{[-p|--path]}} {{path/to/input_file.md}} {{[-o|--output]}} {{path/to/output_file.pdf}}
```

- Convert Markdown content provided as a string:

```
markdown2pdf {{[-s|--string]}} {{markdown_text}} {{[-o|--output]}} {{path/to/output_file.pdf}}
```

- Convert from URL (this will convert a Markdown file at that URL to a local PDF file):

```
markdown2pdf {{[-u|--url]}} {{URL}} {{[-o|--output]}} {{path/to/output_file.pdf}}
```

masscan

A very fast network scanner.

Works best with elevated privileges. For help with Nmap compatibility, run **`masscan --nmap`**.

More information: https://manned.org/masscan.

- Scan an IP or network subnet for port 80:

```
masscan {{ip_address|network_prefix}} {{[-p|--ports]}} {{80}}
```

- Scan a class B subnet for the top 100 ports at 100,000 packets per second:

```
masscan {{10.0.0.0/16}} --top-ports {{100}} --rate {{100000}}
```

- Scan a class B subnet avoiding ranges from a specific exclude file:

```
masscan {{10.0.0.0/16}} --top-ports {{100}} --excludefile
{{path/to/file}}
```

- Scan a class B subnet with Nmap-like version detection (banner grabbing):

```
masscan {{10.0.0.0/16}} {{[-p|--ports]}} {{22,80}} --banners
--rate {{100000}}
```

- Scan the Internet for web servers running on port 80 and 443:

```
masscan {{0.0.0.0/0}} {{[-p|--ports]}} {{80,443}} --rate
{{10000000}}
```

- Scan the Internet for DNS servers running on UDP port 53:

```
masscan {{0.0.0.0/0}} {{[-p|--ports]}} {{U:53}} --rate
{{10000000}}
```

- Scan the Internet for a specific port range and export to a file:

```
masscan {{0.0.0.0/0}} {{[-p|--ports]}} {{0-65535}} --output-
format {{binary|grepable|json|list|xml}} --output-filename
{{path/to/file}}
```

- Read binary scan results from a file and output to `stdout`:

```
masscan --readscan {{path/to/file}}
```

mat2

Anonymise various file formats by removing metadata.

More information: https://0xacab.org/jvoisin/mat2.

- List supported file formats:

```
mat2 --list
```

- Remove metadata from a file:

```
mat2 {{path/to/file}}
```

- Remove metadata from a file and print detailed output to the console:

```
mat2 --verbose {{path/to/file}}
```

- Show metadata in a file without removing it:

```
mat2 --show {{path/to/file}}
```

- Partially remove metadata from a file:

```
mat2 --lightweight {{path/to/file}}
```

- Remove metadata from a file in place, without creating a backup:

```
mat2 --inplace {{path/to/file}}
```

mate-dictionary

Look up words on dictionaries.

More information: https://manned.org/mate-dictionary.

- Print a specific word definition:

```
mate-dictionary --no-window --look-up '{{word}}'
```

- Show similar words for a specific one in a new window:

```
mate-dictionary --match '{{word}}'
```

matlab

Numerical computation environment by MathWorks.

More information: https://www.mathworks.com/help/matlab/matlab_env/startup-options.html.

- Run without splash screen during startup:

```
matlab -nosplash
```

- Execute a MATLAB statement:

```
matlab -r "{{matlab_statement}}"
```

- Run a MATLAB script:

```
matlab -r "run({{path/to/script.m}})"
```

maza

Local ad blocker. Like Pi-hole but local and using the operating system.

More information: https://github.com/tanrax/maza-ad-blocking.

- Update the Maza database:

```
maza update
```

- Start Maza:

```
sudo maza start
```

- Stop Maza:

```
sudo maza stop
```

- Show the status of Maza:

```
maza status
```

mc

MinIO Client for object storage and filesystems.

May be named **`mc`** or **`mcli`** on some systems.

More information: https://minio.github.io/mc/.

- Add connection to a S3 server:

```
mc alias set {{local}} {{http://localhost:9000}}
{{access_key}} {{secret_key}}
```

- Create a bucket:

```
mc mb {{local/bucket_name}}
```

- List buckets and their content recursively:

```
mc ls {{local}} --recursive
```

mc

Midnight Commander, a TUI file manager.

Navigate the directory structure using the **<ArrowKeys>**, the mouse or by typing the commands into the terminal.

See also: **ranger**, **clifm**, **vifm**, **nautilus**.

More information: https://midnight-commander.org.

- Start Midnight Commander:

```
mc
```

- Start Midnight Commander in black and white:

```
mc {{[-b|--nocolor]}}
```

mc

mc can refer to multiple commands with the same name.

- View documentation for MinIO client:

  ```
  tldr mc.cli
  ```

- View documentation for Midnight Commander:

  ```
  tldr mc.fm
  ```

mcfly

A smart command history search and management tool.

Replaces your default shell history search (**<Ctrl r>**) with an intelligent search engine providing context and relevance to the commands.

More information: https://github.com/cantino/mcfly.

- Print the mcfly integration code for the specified shell:

```
mcfly init {{bash|fish|zsh}}
```

- Search the history for a command, with 20 results:

```
mcfly search --results {{20}} "{{search_terms}}"
```

- Add a new command to the history:

```
mcfly add "{{command}}"
```

- Record that a directory has moved and transfer the historical records from the old path to the new one:

```
mcfly move "{{path/to/old_directory}}" "{{path/to/
new_directory}}"
```

- Train the suggestion engine (developer tool):

```
mcfly train
```

- Display help for a specific subcommand:

```
mcfly help {{subcommand}}
```

mcli

This command is an alias of **`mc`** (MinIO client).

- View documentation for the original command:

```
tldr mc.cli
```

mcs

Mono C# Compiler.

More information: https://manned.org/mcs.1.

- Compile the specified files:

```
mcs {{path/to/input_file1.cs path/to/input_file2.cs ...}}
```

- Specify the output program name:

```
mcs -out:{{path/to/file.exe}} {{path/to/input_file1.cs path/to/input_file2.cs ...}}
```

- Specify the output program type:

```
mcs -target:{{exe|winexe|library|module}} {{path/to/input_file1.cs path/to/input_file2.cs ...}}
```

md-to-clip

Convert tldr-pages to Command Line Interface Pages.

See also: **`clip-view`**.

More information: https://github.com/command-line-interface-pages/v2-tooling/tree/main/md-to-clip.

- Convert tldr-pages files and save into the same directories:

```
md-to-clip {{path/to/page1.md path/to/page2.md ...}}
```

- Convert tldr-pages files and save into a specific directory:

```
md-to-clip --output-directory {{path/to/directory}} {{path/to/page1.md path/to/page2.md ...}}
```

- Convert a tldr-page file to `stdout`:

```
md-to-clip --no-file-save <(echo '{{page-content}}')
```

- Convert tldr-pages files while recognizing additional placeholders from a specific config:

```
md-to-clip --special-placeholder-config {{path/to/config.yaml}} {{path/to/page1.md path/to/page2.md ...}}
```

- Display help:

```
md-to-clip --help
```

- Display version:

```
md-to-clip --version
```

md5sum

Calculate MD5 cryptographic checksums.

More information: https://www.gnu.org/software/coreutils/manual/html_node/md5sum-invocation.html.

- Calculate the MD5 checksum for one or more files:

```
md5sum {{path/to/file1 path/to/file2 ...}}
```

- Calculate and save the list of MD5 checksums to a file:

```
md5sum {{path/to/file1 path/to/file2 ...}} > {{path/to/file.md5}}
```

- Calculate an MD5 checksum from `stdin`:

```
{{command}} | md5sum
```

- Read a file of MD5 checksums and filenames and verify all files have matching checksums:

```
md5sum {{[-c|--check]}} {{path/to/file.md5}}
```

- Only show a message for missing files or when verification fails:

```
md5sum {{[-c|--check]}} --quiet {{path/to/file.md5}}
```

- Only show a message when verification fails, ignoring missing files:

```
md5sum --ignore-missing {{[-c|--check]}} --quiet {{path/to/file.md5}}
```

- Check a known MD5 checksum of a file:

```
echo {{known_md5_checksum_of_the_file}} {{path/to/file}} | md5sum {{[-c|--check]}}
```

mdatopbm

Convert a Microdesign MDA file to a PBM image.

See also: **pbmtomda**.

More information: https://netpbm.sourceforge.net/doc/mdatopbm.html.

- Convert a MDA file to a PBM image:

```
mdatopbm {{path/to/image.mda}} > {{path/to/output.pbm}}
```

- Invert the colors in the input image:

```
mdatopbm -i {{path/to/image.mda}} > {{path/to/output.pbm}}
```

- Double the input image's height:

```
mdatopbm -d {{path/to/image.mda}} > {{path/to/output.pbm}}
```

mdk4

A proof-of-concept tool to exploit common IEEE 802.11 protocol weaknesses.

Note: Exercise extreme caution as this tool can disrupt Wi-Fi networks and disconnect nearby users.

See also: **`airodump-ng`**, **`airmon-ng`**.

More information: https://github.com/aircrack-ng/mdk4.

- Flood access points with beacon frames to create fake networks (set interface to monitor mode with `sudo airmon-ng start wifi_interface` if needed):

```
sudo mdk4 {{wifi_interface}} b -f {{path/to/beacons.txt}}
```

- Perform deauthentication attack against all clients on all BSSIDs:

```
sudo mdk4 {{wifi_interface}} d
```

- Perform deauthentication attack on a specific BSSID (list BSSIDs using `sudo airodump-ng wifi_interface`):

```
sudo mdk4 {{wifi_interface}} d -B {{BSSID}}
```

mdp

Make presentations from Markdown files.

More information: https://manned.org/mdp.

- Launch a presentation in the terminal from a Markdown file:

```
mdp {{presentation.md}}
```

- Disable fading transitions:

```
mdp {{[-f|--nofade]}} {{presentation.md}}
```

- Invert font colors to use in terminals with light background:

```
mdp {{[-i|--invert]}} {{presentation.md}}
```

- Disable transparency in transparent terminals:

```
mdp {{[-t|--notrans]}} {{presentation.md}}
```

mediainfo

Display metadata from video and audio files.

More information: https://mediaarea.net/MediaInfo.

- Display metadata for a given file in the console:

```
mediainfo {{file}}
```

- Store the output to a given file along with displaying in the console:

```
mediainfo --Logfile={{out.txt}} {{file}}
```

- List metadata attributes that can be extracted:

```
mediainfo --Info-Parameters
```

medusa

A modular and parallel login brute-forcer for a variety of protocols.

More information: https://manned.org/medusa.

- List all installed modules:

```
medusa -d
```

- Show usage example of a specific module (use `medusa -d` for listing all installed modules):

```
medusa -M {{ssh|http|web-form|postgres|ftp|mysql|...}} -q
```

- Execute brute force against an FTP server using a file containing usernames and a file containing passwords:

```
medusa -M ftp -h host -U {{path/to/username_file}} -P {{path/to/password_file}}
```

- Execute a login attempt against an HTTP server using the username, password and user-agent specified:

```
medusa -M HTTP -h host -u {{username}} -p {{password}} -m USER-AGENT:"{{Agent}}"
```

- Execute a brute force against a MySQL server using a file containing usernames and a hash:

```
medusa -M mysql -h host -U {{path/to/username_file}} -p {{hash}} -m PASS:HASH
```

- Execute a brute force against a list of SMB servers using a username and a pwdump file:

```
medusa -M smbnt -H {{path/to/hosts_file}} -C {{path/to/pwdump_file}} -u {{username}} -m PASS:HASH
```

meld

Graphical diffing and merging tool.

More information: https://manned.org/meld.

- Start `meld`:

```
meld
```

- Compare 2 files:

```
meld {{path/to/file_1}} {{path/to/file_2}}
```

- Compare 2 directories:

```
meld {{path/to/directory_1}} {{path/to/directory_2}}
```

- Compare 3 files:

```
meld {{path/to/file_1}} {{path/to/file_2}} {{path/to/file_3}}
```

- Open a comparison as a new tab in a pre-existing meld instance:

```
meld --newtab {{path/to/file_1}} {{path/to/file_2}}
```

- Compare multiple sets of files:

```
meld --diff {{path/to/file_1}} {{path/to/file_2}} --diff
{{path/to/file_3}} {{path/to/file_4}}
```

melt

Backup and restore Ed25519 SSH keys using memorizable seed phrases.

More information: https://github.com/charmbracelet/melt#usage.

- Generate a seed phrase from an existing Ed25519 private key:

```
melt {{~/.ssh/id_ed25519}}
```

- Generate a seed phrase from `stdin`:

```
{{cat ~/.ssh/id_ed25519}} | melt
```

- Restore an SSH key from a seed phrase:

```
melt restore {{path/to/key}} --seed "{{seed_phrase}}"
```

- Restore an SSH key from a seed phrase via `stdin`:

```
{{cat path/to/file}} | melt restore -
```

mesg

Check or set a terminal's ability to receive messages from other users, usually from the **`write`** command.

See also: **`write`**, **`talk`**.

More information: https://manned.org/mesg.1p.

- Check terminal's openness to write messages:

```
mesg
```

- Disallow receiving messages from the write command:

```
mesg n
```

- Allow receiving messages from the write command:

```
mesg y
```

meshlabserver

Interface for the MeshLab 3D mesh processing software.

More information: https://manned.org/meshlabserver.

- Convert an STL file to an OBJ file:

```
meshlabserver -i {{input.stl}} -o {{output.obj}}
```

- Convert a WRL file to a OFF file, including the vertex and face normals in the output mesh:

```
meshlabserver -i {{input.wrl}} -o {{output.off}} -om vn fn
```

- Dump a list of all the available processing filters into a file:

```
meshlabserver -d {{path/to/file}}
```

- Process a 3D file using a filter script created in the MeshLab GUI (Filters > Show current filter script > Save Script):

```
meshlabserver -i {{input.ply}} -o {{output.ply}} -s
{{filter_script.mlx}}
```

- Process a 3D file using a filter script, writing the output of the filters into a log file:

```
meshlabserver -i {{input.x3d}} -o {{output.x3d}} -s
{{filter_script.mlx}} -l {{logfile}}
```

meshnamed

Distributed naming system for IPv6 mesh networks.

More information: https://github.com/zhoreeq/meshname/.

- Start a local meshname DNS server:

  ```
  meshnamed
  ```

- Convert an IPv6 address into a meshname:

  ```
  meshnamed -getname {{200:6fc8:9220:f400:5cc2:305a:4ac6:967e}}
  ```

- Convert a meshname to an IPv6 address:

  ```
  meshnamed -getip {{aiag7sesed2aaxgcgbnevruwpy}}
  ```

meson

SCons-like build system that uses Python as a front-end language and Ninja as a building backend.

More information: https://mesonbuild.com/Commands.html.

- Generate a C project with a given name and version:

```
meson init {{[-l|--language]}} c {{[-n|--name]}} {{myproject}} --version {{0.1}}
```

- Configure the `builddir` with default values:

```
meson setup {{build_directory}}
```

- Build the project:

```
meson compile -C {{path/to/build_directory}}
```

- Run all tests in the project:

```
meson test
```

- Display help:

```
meson {{[-h|--help]}}
```

- Display version:

```
meson {{[-v|--version]}}
```

meteor

Full-stack JavaScript platform for building web applications.

More information: https://meteor.com.

- Run a meteor project from its root directory in development mode:

```
meteor
```

- Create a project under the given directory:

```
meteor create {{path/to/directory}}
```

- Display the list of packages the project is currently using:

```
meteor list
```

- Add a package to the project:

```
meteor add {{package}}
```

- Remove a package from the project:

```
meteor remove {{package}}
```

- Create a production build of the project as a tarball under the given directory:

```
meteor build {{path/to/directory}}
```

mg

A small, fast, and portable text editor based on **emacs**.

More information: https://manned.org/mg.

- Open a file for editing:

```
mg {{path/to/file}}
```

- Open a file at a specified line number:

```
mg +{{line_number}} {{path/to/file}}
```

- Open files in a read-only mode:

```
mg -R {{path/to/file1 path/to/file2 ...}}
```

- Disable ~ backup files while editing:

```
mg -n {{path/to/file}}
```

mgrtopbm

Convert a MGR bitmap into a PBM file.

See also: **pbmtomgr**.

More information: https://netpbm.sourceforge.net/doc/mgrtopbm.html.

- Convert a MGR bitmap into a PBM file:

```
mgrtopbm {{path/to/image.mgr}} > {{path/to/output.pbm}}
```

mh_copyright

Adjust copyright headers for MATLAB or Octave code.

More information: https://misshit.org.

- Update the year (range) to include the current year for the specified files:

```
mh_copyright --primary-entity="{{entity}}" --update-year
{{path/to/file_or_directory1.m path/to/
file_or_director2.m ...}}
```

- Update the year (range) to include the current year for all files:

```
mh_copyright --primary-entity="{{entity}}" --update-year
```

mh_lint

Attempt to find bugs in MATLAB or Octave code.

Please note that this tool is neither sound nor complete.

More information: https://misshit.org.

- Check the current directory:

```
mh_lint
```

- Check a specific directory recursively:

```
mh_lint {{path/to/directory}}
```

- Check a MATLAB file:

```
mh_lint {{path/to/file.m}}
```

- Check an Octave file:

```
mh_lint --octave {{path/to/file.m}}
```

mh_metric

Calculate and enforce code metrics for MATLAB or Octave code.

More information: https://misshit.org.

- Print the code metrics for the specified files:

```
mh_metric {{path/to/file1.m path/to/file2.m ...}}
```

- Print the code metrics for the specified Octave files:

```
mh_metric --octave {{path/to/file1.m path/to/file2.m ...}}
```

- Print the code metrics for the specified directory recursively:

```
mh_metric {{path/to/directory}}
```

- Print the code metrics for the current directory:

```
mh_metric
```

- Print the code metrics report in HTML or JSON format:

```
mh_metric --{{html|json}} {{path/to/output_file}}
```

micro

A modern and intuitive terminal-based text editor.

You can use your keyboard, but also your mouse to navigate and/or select text.

More information: https://micro-editor.github.io.

- Open a file:

```
micro {{path/to/file}}
```

- Save a file:

```
<Ctrl s>
```

- Cut the entire line:

```
<Ctrl k>
```

- Search for a pattern in the file (press `<Ctrl n>`/`<Ctrl p>` to go to next/previous match):

```
<Ctrl f>{{pattern}}<Enter>
```

- Execute a command:

```
<Ctrl e>{{command}}<Enter>
```

- Perform a substitution in the whole file:

```
<Ctrl e>replaceall "{{pattern}}" "{{replacement}}"<Enter>
```

- Quit:

```
<Ctrl q>
```

micromamba

A fast, minimal, standalone package and environment manager for **conda** packages.

Drop-in replacement for **conda**, ideal for CI, Docker, and lightweight setups.

More information: https://manned.org/micromamba.

- Create a new environment at a specific path, installing named packages into it:

```
micromamba create {{[-p|--prefix]}} /{{path/to/env}} {{python=3.11 numpy}}
```

- Activate an environment by name or path:

```
micromamba activate {{[-p|--prefix]}} /{{path/to/env}}
```

- Run a command inside an environment without activating it in the shell:

```
micromamba run {{[-p|--prefix]}} /{{path/to/env}} {{pytest tests/}}
```

- Install packages into the currently active environment:

```
micromamba install {{scipy pandas}}
```

- List all installed packages in the current environment:

```
micromamba list
```

- Search for packages in channels or current environment:

```
micromamba search {{package_name}}
```

- Query tree-like dependencies of a package:

```
micromamba repoquery depends {{[-t|--tree]}} {{package_name}}
```

- Show information about the current `micromamba` setup:

```
micromamba info
```

microsoft-edge

Modern web browser developed by Microsoft based on the Chromium web browser developed by Google.

This command is available instead as **`msedge`** for Windows.

Note: Additional command arguments from **`chromium`** may also be usable to control Microsoft Edge.

More information: https://microsoft.com/edge.

- Open a specific URL or file:

```
microsoft-edge {{https://example.com|path/to/file.html}}
```

- Open in InPrivate mode:

```
microsoft-edge --inprivate {{example.com}}
```

- Open in a new window:

```
microsoft-edge --new-window {{example.com}}
```

- Open in application mode (without toolbars, URL bar, buttons, etc.):

```
microsoft-edge --app={{https://example.com}}
```

- Use a proxy server:

```
microsoft-edge --proxy-server="{{socks5://hostname:
66}}" {{example.com}}
```

- Open with a custom profile directory:

```
microsoft-edge --user-data-dir={{path/to/directory}}
```

- Open without CORS validation (useful to test an API):

```
microsoft-edge --user-data-dir={{path/to/directory}} --
disable-web-security
```

- Open with a DevTools window for each tab opened:

```
microsoft-edge --auto-open-devtools-for-tabs
```

mid3v2

Edit audio tags.

See also: **id3v2**.

More information: https://mutagen.readthedocs.io/en/latest/man/mid3v2.html.

- List all supported ID3v2.3 or ID3v2.4 frames and their meanings:

```
mid3v2 --list-frames {{path/to/file1.mp3 path/to/
file2.mp3 ...}}
```

- List all supported ID3v1 numeric genres:

```
mid3v2 --list-genres {{path/to/file1.mp3 path/to/
file2.mp3 ...}}
```

- List all tags in specific files:

```
mid3v2 --list {{path/to/file1.mp3 path/to/file2.mp3 ...}}
```

- Set specific artist, album, or song information:

```
mid3v2 {{--artist|--album|--song}}={{string}} {{path/to/
file1.mp3 path/to/file2.mp3 ...}}
```

- Set specific picture information:

```
mid3v2 --
picture={{filename:description:image_type:mime_type}} {{path/
to/file1.mp3 path/to/file2.mp3 ...}}
```

- Set specific year information:

```
mid3v2 --year={{YYYY}} {{path/to/file1.mp3 path/to/
file2.mp3 ...}}
```

- Set specific date information:

```
mid3v2 --date={{YYYY-MM-DD}} {{path/to/file1.mp3 path/to/
file2.mp3 ...}}
```

middleman

Static site generator written in Ruby.

More information: https://middlemanapp.com/basics/install/.

- Create a new Middleman project:

```
middleman init "{{project_name}}"
```

- Start local server for current project on port 4567:

```
middleman server
```

- Start local server for current project on a specified port:

```
middleman server -p "{{port}}"
```

- Build the project in the current directory to prepare for deployment:

```
bundle exec middleman build
```

- Deploy the Middleman project in the current directory:

```
middleman deploy
```

Minecraft

Run a headless Minecraft server.

More information: https://minecraft.wiki/w/Tutorial:Setting_up_a_Java_Edition_server.

- Start a Minecraft server and generate a world if it doesn't exist:

```
java -jar {{path/to/server.jar}} --nogui
```

- Set the minimum and maximum amount of memory a server is allowed to have (Note: Setting them the same prevents lag caused by heap scaling):

```
java -Xms{{1024M}} -Xmx{{2048M}} -jar {{path/to/server.jar}} --nogui
```

- Start a server with a GUI:

```
java -jar {{path/to/server.jar}}
```

- Shut the server down:

```
stop
```

minetest

Multiplayer infinite-world block sandbox.

See also: **`minetestserver`**, the server-only binary.

More information: https://wiki.minetest.org/Minetest.

- Start Minetest in client mode:

```
minetest
```

- Start Minetest in server mode by hosting a specific world:

```
minetest --server --world {{name}}
```

- Write logs to a specific file:

```
minetest --logfile {{path/to/file}}
```

- Only write errors to the console:

```
minetest --quiet
```

minetestserver

Multiplayer infinite-world block sandbox server.

See also: **`minetest`**, the graphical client.

More information: https://wiki.minetest.org/Setting_up_a_server.

- Start the server:

```
minetestserver
```

- List available worlds:

```
minetestserver --world list
```

- Load the specified world:

```
minetestserver --world {{world_name}}
```

- List the available game IDs:

```
minetestserver --gameid list
```

- Use the specified game:

```
minetestserver --gameid {{game_id}}
```

- Listen on a specific port:

```
minetestserver --port {{34567}}
```

- Migrate to a different data backend:

```
minetestserver --migrate {{sqlite3|leveldb|redis}}
```

- Start an interactive terminal after starting the server:

```
minetestserver --terminal
```

mingle

Bundle the edges of a graph layout.

Graphviz filters: **`acyclic`**, **`bcomps`**, **`comps`**, **`edgepaint`**, **`gvcolor`**, **`gvpack`**, **`mingle`**, **`nop`**, **`sccmap`**, **`tred`**, & **`unflatten`**.

More information: https://www.graphviz.org/pdf/mingle.1.pdf.

- Bundle the edges of one or more graph layouts (that already have layout information):

```
mingle {{path/to/layout1.gv path/to/layout2.gv ...}} > {{path/to/output.gv}}
```

- Perform layout, bundling, and output to a picture with one command:

```
dot {{path/to/input.gv}} | mingle | dot -T {{png}} > {{path/to/output.png}}
```

- Display help:

```
mingle -?
```

minidlna

ReadyMedia (formerly MiniDLNA) is a lightweight media server compatible with DLNA/UPnP-AV clients.

Used to stream media to smart TVs, consoles, and other DLNA-compatible devices.

Configuration is typically done via the **`minidlna.conf`** file.

More information: https://manned.org/minidlna.

- Start the MiniDLNA daemon using the default configuration at `/etc/minidlna.conf`:

```
minidlna
```

- Start MiniDLNA with a specific configuration file:

```
minidlna -f {{path/to/minidlna.conf}}
```

- Force a database rescan on startup:

```
minidlna -R
```

- Run MiniDLNA in the foreground (useful for debugging):

```
minidlna -d
```

- Enable verbose debug output:

```
minidlna -d -v
```

- Specify a custom media directory (overrides config):

```
minidlna -m {{path/to/media}}
```

- Specify a custom database directory:

```
minidlna -P {{path/to/pidfile}}
```

- Specify a custom log file path:

```
minidlna -l {{path/to/logfil.log}}
```

minifab

Automate the setup and deployment of Hyperledger Fabric networks.

More information: https://github.com/hyperledger-labs/minifabric.

- Bring up the default Hyperledger Fabric network:

```
minifab up {{[-i|--fabric-release]}} {{minifab_version}}
```

- Bring down the Hyperledger Fabric network:

```
minifab down
```

- Install chaincode onto a specified channel:

```
minifab install {{[-n|--chaincode-name]}} {{chaincode_name}}
```

- Install a specific chaincode version onto a channel:

```
minifab install {{[-n|--chaincode-name]}} {{chaincode_name}}
{{[-v|--chaincode-version]}} {{chaincode_version}}
```

- Initialize the chaincode after installation/upgrade:

```
minifab approve,commit,initialize,discover
```

- Invoke a chaincode method with the specified arguments:

```
minifab invoke {{[-n|--chaincode-name]}} {{chaincode_name}}
{{[-p|--chaincode-parameters]}} '"{{method_name}}",
{{"argument1", "argument2", ...}}'
```

- Make a query on the ledger:

```
minifab blockquery {{block_number}}
```

- Quickly run an application:

```
minifab apprun {{[-l|--chaincode-language]}}
{{app_programming_language}}
```

minikube start

Start **`minikube`** with different configurations.

More information: https://minikube.sigs.k8s.io/docs/commands/start/.

- Start `minikube` with a specific Kubernetes version:

  ```
  minikube start --kubernetes-version {{v1.24.0}}
  ```

- Start `minikube` with specific resource allocations (e.g., memory and CPU):

  ```
  minikube start --memory {{2048}} --cpus {{2}}
  ```

- Start `minikube` with a specific driver (e.g., VirtualBox):

  ```
  minikube start --driver {{virtualbox}}
  ```

- Start `minikube` in the background (headless mode):

  ```
  minikube start --background
  ```

- Start `minikube` with custom add-ons (e.g., the metrics server):

  ```
  minikube start --addons {{metrics-server}}
  ```

minikube

Run Kubernetes locally.

More information: https://minikube.sigs.k8s.io/docs/.

- Start the cluster:

  ```
  minikube start
  ```

- Get the IP address of the cluster:

  ```
  minikube ip
  ```

- Access a service named my_service exposed via a node port and get the URL:

  ```
  minikube service {{my_service}} --url
  ```

- Open the Kubernetes dashboard in a browser:

  ```
  minikube dashboard
  ```

- Stop the running cluster:

  ```
  minikube stop
  ```

- Delete the cluster:

  ```
  minikube delete
  ```

- Connect to LoadBalancer services:

  ```
  minikube tunnel
  ```

minio-client

This command is an alias of **`mc`** (MinIO client).

- View documentation for the original command:

```
tldr mc.cli
```

minio server

MinIO server command for starting the MinIO S3 compatible storage engine.

More information: https://docs.min.io/enterprise/aistor-object-store/reference/aistor-server/.

- Start a server using the default configuration:

```
minio server {{path/to/data_directory}}
```

- Start a server binding to a different port for the API and web console:

```
minio server --address ":{{port}}" --console-address ":{{port}}" {{path/to/data_directory}}
```

- Start a server as part of a cluster of 2 nodes:

```
minio server {{node1_hostname}} {{path/to/data_directory}} {{node2_hostname}} {{path/to/data_directory}}
```

miniserve

Simple HTTP file server.

More information: https://github.com/svenstaro/miniserve.

- Serve a directory:

```
miniserve {{path/to/directory}}
```

- Serve a single file:

```
miniserve {{path/to/file}}
```

- Serve a directory using HTTP basic authentication:

```
miniserve --auth {{username}}:{{password}} {{path/to/
directory}}
```

minisign

A dead simple tool to sign files and verify signatures.

More information: https://jedisct1.github.io/minisign/.

- Generate a new keypair at the default location:

```
minisign -G
```

- Sign a file:

```
minisign -Sm {{path/to/file}}
```

- Sign a file, adding a trusted (signed) and an untrusted (unsigned) comment in the signature:

```
minisign -Sm {{path/to/file}} -c "{{Untrusted comment}}" -t "{{Trusted comment}}"
```

- Verify a file and the trusted comments in its signature using the specified public key file:

```
minisign -Vm {{path/to/file}} -p {{path/to/publickey.pub}}
```

- Verify a file and the trusted comments in its signature, specifying a public key as a Base64 encoded literal:

```
minisign -Vm {{path/to/file}} -P "{{public_key_base64}}"
```

mise

Manage language runtimes like Node.js, Python, Ruby, Go, Java, etc and various tools.

More information: https://mise.jdx.dev.

- List all available plugins:

```
mise plugins list-all
```

- Install a plugin:

```
mise plugins add {{name}}
```

- List runtime versions available for install:

```
mise ls-remote {{name}}
```

- Install a specific version of a package:

```
mise install {{name}}@{{version}}
```

- Set global version for a package:

```
mise use --global {{name}}@{{version}}
```

- Set local version for a package:

```
mise use {{name}}@{{version}}
```

- Set environment variable in configuration:

```
mise set {{variable}}={{value}}
```

- Pass plugin options:

```
mise use {{name}}\[{{option1}}={{option1_value}},{{option2}}
={{option2_value}}\]@{{version}}
```

mispipe

Pipe two commands and return the exit status of the first command.

More information: https://manned.org/mispipe.

- Pipe two commands and return the exit status of the first command:

```
mispipe {{command1}} {{command2}}
```

mitmdump

View, record, and programmatically transform HTTP traffic.

The command-line counterpart to mitmproxy.

More information: https://docs.mitmproxy.org/stable/#mitmdump.

- Start a proxy and save all output to a file:

```
mitmdump {{[-w|--wfile]}} {{path/to/file}}
```

- Filter a saved traffic file to just POST requests:

```
mitmdump {{[-nr|--no-server --read-flows]}}
{{input_filename}} {{[-w|--wfile]}} {{output_filename}} "{{~m
post}}"
```

- Replay a saved traffic file:

```
mitmdump {{[-nc|--no-server --client-replay]}} {{path/to/
file}}
```

- Intercept DNS traffic (starts an intercepting DNS server on 127.0.0.1:53):

```
sudo mitmdump {{[-m|--mode]}} dns
```

mitmproxy

An interactive man-in-the-middle HTTP proxy.

See also: **`mitmweb`**, **`mitmdump`**.

More information: https://docs.mitmproxy.org/stable/.

- Start `mitmproxy` with default settings (will listen on port `8080`):

```
mitmproxy
```

- Start `mitmproxy` bound to a custom address and port:

```
mitmproxy --listen-host {{ip_address}} {{[-p|--listen-port]}} {{port}}
```

- Start `mitmproxy` using a script to process traffic:

```
mitmproxy {{[-s|--scripts]}} {{path/to/script.py}}
```

- Export the logs with SSL/TLS master keys to external programs (wireshark, etc.):

```
SSLKEYLOGFILE="{{path/to/file}}" mitmproxy
```

- Specify mode of operation of the proxy server (`regular` is the default):

```
mitmproxy {{[-m|--mode]}} {{regular|transparent|socks5|...}}
```

- Set the console layout:

```
mitmproxy --console-layout {{horizontal|single|vertical}}
```

- Save all proxied traffic to a file for later analysis:

```
mitmproxy {{[-w|--save-stream-file]}} {{path/to/dump.mitm}}
```

- Replay a previously saved HTTP flow file:

```
mitmproxy {{[-nr|--no-server --rfile]}} {{path/to/dump.mitm}}
```

mitmweb

A web-based interactive man-in-the-middle HTTP proxy.

See also: **`mitmproxy`**.

More information: https://docs.mitmproxy.org/stable/concepts-options.

- Start `mitmweb` with default settings:

```
mitmweb
```

- Start `mitmweb` bound to a custom address and port:

```
mitmweb --listen-host {{ip_address}} --listen-port {{port}}
```

- Start `mitmweb` using a script to process traffic:

```
mitmweb --scripts {{path/to/script.py}}
```

mix

Build tool that provides tasks for creating, compiling, and testing Elixir projects, managing its dependencies, and more.

More information: https://hexdocs.pm/mix.

- Execute a particular file:

  ```
  mix run {{my_script.exs}}
  ```

- Create a new project:

  ```
  mix new {{project_name}}
  ```

- Compile project:

  ```
  mix compile
  ```

- Run project tests:

  ```
  mix test
  ```

- Display help:

  ```
  mix help
  ```

mixxx

Free and open source cross-platform DJ software.

See also: **`lmms`**.

More information: https://mixxx.org/manual/latest/chapters/appendix.html#command-line-options.

- Start the Mixxx GUI in fullscreen:

```
mixxx --fullScreen
```

- Start in safe developer mode to debug a crash:

```
mixxx --developer --safeMode
```

- Debug a malfunction:

```
mixxx --debugAssertBreak --developer --loglevel trace
```

- Start Mixxx using the specified settings file:

```
mixxx --resourcePath {{mixxx/res/controllers}} --settingsPath {{path/to/settings-file}}
```

- Debug a custom controller mapping:

```
mixxx --controllerDebug --resourcePath {{path/to/mapping-directory}}
```

- Display help:

```
mixxx --help
```

mk

Task runner for targets described in Mkfile.

Mostly used to control the compilation of an executable from source code.

More information: https://doc.cat-v.org/plan_9/4th_edition/papers/mk.

- Call the first target specified in the Mkfile (usually named "all"):

```
mk
```

- Call a specific target:

```
mk {{target}}
```

- Call a specific target, executing 4 jobs at a time in parallel:

```
NPROC=4 mk {{target}}
```

- Force mking of a target, even if source files are unchanged:

```
mk -w{{target}} {{target}}
```

- Assume all targets to be out of date. Thus, update `target` and all of its dependencies:

```
mk -a {{target}}
```

- Keep going as far as possible on error:

```
mk -k
```

mkcert

Make locally-trusted development certificates.

More information: https://github.com/FiloSottile/mkcert.

- Install the local CA in the system trust store:

```
mkcert -install
```

- Generate certificate and private key for a given domain:

```
mkcert {{example.org}}
```

- Generate certificate and private key for multiple domains:

```
mkcert {{example.org}} {{myapp.dev}} {{127.0.0.1}}
```

- Generate wildcard certificate and private key for a given domain and its subdomains:

```
mkcert "{{*.example.it}}"
```

- Uninstall the local CA:

```
mkcert -uninstall
```

mkdir

Create directories and set their permissions.

More information: https://www.gnu.org/software/coreutils/manual/html_node/mkdir-invocation.html.

- Create specific directories:

```
mkdir {{path/to/directory1 path/to/directory2 ...}}
```

- Create specific directories and their parents if needed:

```
mkdir {{[-p|--parents]}} {{path/to/directory1 path/to/directory2 ...}}
```

- Create directories with specific permissions:

```
mkdir {{[-m|--mode]}} {{rwxrw-r--}} {{path/to/directory1 path/to/directory2 ...}}
```

- Create multiple nested directories recursively:

```
mkdir {{[-p|--parents]}} {{path/to/{a,b}/{x,y,z}/{h,i,j}}}
```

mkdocs

Project documentation with Markdown.

More information: https://www.mkdocs.org.

- Create a new mkdocs project:

  ```
  mkdocs new {{project_name}}
  ```

- Serve the project in the current directory using the mkdocs dev-server:

  ```
  mkdocs serve
  ```

- Build the documentation in the current directory:

  ```
  mkdocs build
  ```

- Deploy the documentation in the current directory to GitHub pages:

  ```
  mkdocs gh-deploy
  ```

mkfifo

Make FIFOs (named pipes).

More information: https://www.gnu.org/software/coreutils/manual/html_node/mkfifo-invocation.html.

- Create a named pipe at a given path:

```
mkfifo {{path/to/pipe}}
```

- Send data through a named pipe and send the command to the background:

```
echo "{{Hello World}}" > {{path/to/pipe}} &
```

- Receive data through a named pipe:

```
cat {{path/to/pipe}}
```

- Share your terminal session in real-time:

```
mkfifo {{path/to/pipe}}; script {{[-f|--flush]}} {{path/to/pipe}}
```

mkfile

Create empty files of any size.

More information: https://manned.org/mkfile.

- Create an empty file of 15 kilobytes:

```
mkfile -n {{15k}} {{path/to/file}}
```

- Create a file of a given size and unit (bytes, KB, MB, GB):

```
mkfile -n {{size}}{{b|k|m|g}} {{path/to/file}}
```

- Create two files of 4 megabytes each:

```
mkfile -n {{4m}} {{first_filename}} {{second_filename}}
```

mktemp

Create a temporary file or directory.

More information: https://man.openbsd.org/mktemp.1.

- Create an empty temporary file and print its absolute path:

```
mktemp
```

- Use a custom directory if `$TMPDIR` is not set (the default is platform-dependent, but usually `/tmp`):

```
mktemp -p /{{path/to/temporary_directory}}
```

- Use a custom path template (`X`s are replaced with random alphanumeric characters):

```
mktemp {{/tmp/example.XXXXXXXX}}
```

- Use a custom file name template:

```
mktemp -t {{example.XXXXXXXX}}
```

- Create an empty temporary directory and print its absolute path:

```
mktemp -d
```

mktorrent

Create BitTorrent metainfo files.

More information: https://github.com/Rudde/mktorrent.

- Create a torrent with 2^21 KB as the piece size:

```
mktorrent {{[-a|--announce]}} {{tracker_announce_url}} {{[-
l|--piece-length]}} {{21}} {{[-o|--output]}} {{path/to/
example.torrent}} {{path/to/file_or_directory}}
```

- Create a private torrent with a 2^21 KB piece size:

```
mktorrent {{[-p|--private]}} {{[-a|--announce]}}
{{tracker_announce_url}} {{[-l|--piece-length]}} {{21}} {{[-
o|--output]}} {{path/to/example.torrent}} {{path/to/
file_or_directory}}
```

- Create a torrent with a comment:

```
mktorrent {{[-c|--comment]}} "{{comment}}" {{[-a|--
announce]}} {{tracker_announce_url}} {{[-l|--piece-length]}}
{{21}} {{[-o|--output]}} {{path/to/example.torrent}} {{path/
to/file_or_directory}}
```

- Create a torrent with multiple trackers:

```
mktorrent {{[-a|--announce]}}
{{tracker_announce_url,tracker_announce_url_2}} {{[-l|--
piece-length]}} {{21}} {{[-o|--output]}} {{path/to/
example.torrent}} {{path/to/file_or_directory}}
```

- Create a torrent with web seed URLs:

```
mktorrent {{[-a|--announce]}} {{tracker_announce_url}} -w
{{web_seed_url}} {{[-l|--piece-length]}} {{21}} {{[-o|--
output]}} {{path/to/example.torrent}} {{path/to/
file_or_directory}}
```

mkvmerge

Merge and extract multimedia streams.

More information: https://mkvtoolnix.download/doc/mkvmerge.html.

- Display information about a Matroska file:

```
mkvmerge --identify {{path/to/file.mkv}}
```

- Extract the audio from track 1 of a specific file:

```
mkvextract tracks {{path/to/file.mkv}} {{1}}:{{path/to/
output.webm}}
```

- Extract the subtitle from track 3 of a specific file:

```
mkvextract tracks {{path/to/file.mkv}} {{3}}:{{path/to/
subs.srt}}
```

- Add a subtitle track to a file:

```
mkvmerge --output {{path/to/output.mkv}} {{path/to/file.mkv}}
{{path/to/subs.srt}}
```

mlr

Miller is like **awk**, **sed**, **cut**, **join**, and **sort** for name-indexed data such as CSV, TSV, and tabular JSON.

More information: https://miller.readthedocs.io.

- Pretty-print a CSV file in a tabular format:

```
mlr --icsv --opprint cat {{example.csv}}
```

- Receive JSON data and pretty print the output:

```
{{echo '{"key":"value"}'}} | mlr --ijson --opprint cat
```

- Sort alphabetically on a field:

```
mlr --icsv --opprint sort -f {{field}} {{example.csv}}
```

- Sort in descending numerical order on a field:

```
mlr --icsv --opprint sort -nr {{field}} {{example.csv}}
```

- Convert CSV to JSON, performing calculations and display those calculations:

```
mlr --icsv --ojson put '${{newField1}} = ${{oldFieldA}}/$
{{oldFieldB}}' {{example.csv}}
```

- Receive JSON and format the output as vertical JSON:

```
{{echo '{"key1":"value1", "key2":"value2"}'}} | mlr --ijson
--ojson --jvstack cat
```

- Filter lines of a compressed CSV file treating numbers as [S]trings:

```
mlr --prepipe 'gunzip' {{[-c|--csv]}} filter {{[-S|--infer-
none]}} '${{fieldName}} =~ "{{regex}}"' {{example.csv.gz}}
```

mm2gv

Convert a graph from Matrix Market **mm** format to **gv** format.

Converters: **gml2gv**, **gv2gml**, **gv2gxl**, **gxl2gv**, **graphml2gv** & **mm2gv**.

More information: https://graphviz.org/pdf/mm2gv.1.pdf.

- Convert a graph from `mm` to `gv` format:

```
mm2gv -o {{output.gv}} {{input.mm}}
```

- Convert a graph using `stdin` and `stdout`:

```
cat {{input.mm}} | mm2gv > {{output.gv}}
```

- Display help:

```
mm2gv -?
```

mmdc

CLI for mermaid, a diagram generation tool with a domain-specific language.

A mermaid definition file is taken as input and a SVG, PNG, or PDF file is generated as output.

More information: https://mermaid-js.github.io/mermaid/.

- Convert a file to the specified format (automatically determined from the file extension):

```
mmdc {{[-i|--input]}} {{input.mmd}} {{[-o|--output]}} {{output.svg}}
```

- Specify the theme of the chart:

```
mmdc {{[-i|--input]}} {{input.mmd}} {{[-o|--output]}} {{output.svg}} {{[-t|--theme]}} {{forest|dark|neutral|default}}
```

- Specify the background color of the chart (e.g. `lime`, `"#D8064F"`, or `transparent`):

```
mmdc {{[-i|--input]}} {{input.mmd}} {{[-o|--output]}} {{output.svg}} {{[-b|--backgroundColor]}} {{color}}
```

mmls

Display the partition layout of a volume system.

More information: https://wiki.sleuthkit.org/index.php?title=Mmls.

- Display the partition table stored in an image file:

```
mmls {{path/to/image_file}}
```

- Display the partition table with an additional column for the partition size:

```
mmls -B -i {{path/to/image_file}}
```

- Display the partition table in a split EWF image:

```
mmls -i ewf {{image.e01}} {{image.e02}}
```

- Display nested partition tables:

```
mmls -t {{nested_table_type}} -o {{offset}} {{path/to/
image_file}}
```

mmv

Move and rename files in bulk.

More information: https://manned.org/mmv.1.

- Rename all files with a certain extension to a different extension:

```
mmv "*{{.old_extension}}" "#1{{.new_extension}}"
```

- Copy `report6part4.txt` to `./french/rapport6partie4.txt` along with all similarly named files:

```
mmv {{[-c|--copy]}} "{{report*part*.txt}}" "{{./french/
rapport#1partie#2.txt}}"
```

- Append all `.txt` files into one file:

```
mmv {{[-a|--append]}} "{{*.txt}}" "{{all.txt}}"
```

- Convert dates in filenames from "M-D-Y" format to "D-M-Y" format:

```
mmv "{{[0-1][0-9]-[0-3][0-9]-[0-9][0-9][0-9][0-9].txt}}"
"{{#3#4-#1#2-#5#6#7#8.txt}}"
```

mocha

A feature-rich JavaScript test framework.

More information: https://mochajs.org.

- Run tests with default configuration or as configured in `mocha.opts`:

  ```
  mocha
  ```

- Run tests contained at a specific location:

  ```
  mocha {{path/to/test_directory}}
  ```

- Run tests that match a specific `grep` pattern:

  ```
  mocha --grep {{regex}}
  ```

- Run tests on changes to JavaScript files in the current directory and once initially:

  ```
  mocha --watch
  ```

- Run tests with a specific reporter:

  ```
  mocha --reporter {{reporter}}
  ```

mods

AI for the command-line, built for pipelines.

More information: https://github.com/charmbracelet/mods.

- Ask a generic question:

```
mods "{{write me a poem about platypuses}}"
```

- Open settings in your `$EDITOR`:

```
mods --settings
```

- Ask for comments on your code, in markdown format:

```
mods < {{path/to/file}} --format "{{what are your thoughts on
improving this code?}}"
```

- Ask for help with your documentation, in markdown format:

```
mods < {{README.md}} --format "{{write a new section to this
readme for a feature that sends you a free rabbit if you hit
r}}"
```

- Organize your videos, in markdown format:

```
ls {{path/to/videos}} | mods --format "{{organize these by
decade and summarize}}"
```

- Read through raw HTML and summarize the contents, in markdown format:

```
curl "{{https://api.open-meteo.com/v1/forecast?
latitude=29.00&longitude=-90.00&current_weather=true&hourly=temperatur
| mods --format "{{summarize this weather data for a human}}"
```

- Display help:

```
mods --help
```

modutil

Manage PKCS #11 module information within the NSS security module database.

More information: https://manned.org/modutil.

- Add a PKCS #11 module to a NSS database (e.g. a Firefox profile: `$HOME/.mozilla/firefox/default-release`):

```
modutil -dbdir sql:{{path/to/nss_database_directory}} -add
"{{module_label}}" -libfile {{path/to/pkcs11_mod.so}}
```

- List PKCS #11 modules in a NSS database:

```
modutil -dbdir sql:{{path/to/nss_database_directory}} -list
```

moe

A WYSIWYG text editor for ISO-8859-15 encoded text.

More information: https://www.gnu.org/software/moe/moe.html.

- Open moe and create a backup file (file~) when saving edits:

```
moe {{path/to/file}}
```

- Open a file as read-only:

```
moe {{[-o|--read-only]}} {{path/to/file}}
```

- Edit a file without creating backups:

```
moe {{[-B|--no-backup]}} {{path/to/file}}
```

- Edit a file ignoring case in searches:

```
moe {{[-i|--ignore-case]}} {{path/to/file}}
```

- Save and Quit:

```
<Ctrl x>
```

mogrify

This command is an alias of **`magick mogrify`**.

- View documentation for the original command:

```
tldr magick mogrify
```

molecule

Molecule helps testing Ansible roles.

More information: https://molecule.readthedocs.io.

- Create a new Ansible role:

  ```
  molecule init role --role-name {{role_name}}
  ```

- Run tests:

  ```
  molecule test
  ```

- Start the instance:

  ```
  molecule create
  ```

- Configure the instance:

  ```
  molecule converge
  ```

- List scenarios of the instance:

  ```
  molecule matrix converge
  ```

- Log in into the instance:

  ```
  molecule login
  ```

mongo

The legacy MongoDB shell. See **mongosh** for the new shell.

Note: All connection options can be replaced with one string: **mongodb://user@host:port/db_name?authSource=authdb_name**.

More information: https://docs.mongodb.com/manual/reference/program/mongo.

- Connect to a local database on the default port (`mongodb://localhost:27017`):

```
mongo
```

- Connect to a database:

```
mongo --host {{host}} --port {{port}} {{db_name}}
```

- Authenticate using the specified username on the specified database (you will be prompted for a password):

```
mongo --host {{host}} --port {{port}} --username {{username}} --authenticationDatabase {{authdb_name}} {{db_name}}
```

- Evaluate a JavaScript expression on a database:

```
mongo --eval '{{JSON.stringify(db.foo.findOne())}}' {{db_name}}
```

mongod

The MongoDB database server.

More information: https://docs.mongodb.com/manual/reference/program/mongod.

- Specify the storage directory (default: `/data/db` on Linux and macOS, `C:\data\db` on Windows):

```
mongod --dbpath {{path/to/directory}}
```

- Specify a configuration file:

```
mongod --config {{path/to/file}}
```

- Specify the port to listen on (default: 27017):

```
mongod --port {{port}}
```

- Specify the database profiling level. 0 is off, 1 is only slow operations, 2 is all (default: 0):

```
mongod --profile {{0|1|2}}
```

mongodump

Utility to export the contents of a MongoDB instance.

More information: https://docs.mongodb.com/database-tools/mongodump/.

- Create a dump of all databases (this will place the files inside a directory called "dump"):

```
mongodump
```

- Specify an output location for the dump:

```
mongodump {{[-o|--out]}} {{path/to/directory}}
```

- Create a dump of a given database:

```
mongodump {{[-d|--db]}} {{database_name}}
```

- Create a dump of a given collection within a given database:

```
mongodump {{[-c|--collection]}} {{collection_name}} {{[-d|--db]}} {{database_name}}
```

- Connect to a given host running on a given port, and create a dump:

```
mongodump {{[-h|--host]}} {{host}} --port {{port}}
```

- Create a dump of a given database with a given username; user will be prompted for password:

```
mongodump {{[-u|--username]}} {{username}} {{database}} {{[-p|--password]}}
```

- Create a dump from a specific instance; host, user, password and database will be defined in the connection string:

```
mongodump --uri {{connection_string}}
```

mongoexport

Produce exports of data stored in a MongoDB instance formatted as JSON or CSV.

More information: https://docs.mongodb.com/database-tools/mongoexport/.

- Export a collection to `stdout`, formatted as JSON:

```
mongoexport --uri={{connection_string}} --
collection={{collection_name}}
```

- Export the documents in the specified collection that match a query to a JSON file:

```
mongoexport --db={{database_name}} --
collection={{collection_name}} --query="{{query_object}}" --
out={{path/to/file.json}}
```

- Export documents as a JSON array instead of one object per line:

```
mongoexport --collection={{collection_name}} --jsonArray
```

- Export documents to a CSV file:

```
mongoexport --collection={{collection_name}} --type={{csv}}
--fields="{{field1,field2,...}}" --out={{path/to/file.csv}}
```

- Export documents that match the query in the specified file to a CSV file, omitting the list of field names on the first line:

```
mongoexport --collection={{collection_name}} --type={{csv}}
--fields="{{field1,field2,...}}" --queryFile={{path/to/file}}
--noHeaderLine --out={{path/to/file.csv}}
```

- Export documents to `stdout`, formatted as human-readable JSON:

```
mongoexport --uri={{mongodb_uri}} --
collection={{collection_name}} --pretty
```

- Display help:

```
mongoexport --help
```

mongoimport

Imports content from a JSON, CSV, or TSV file into a MongoDB database.

More information: https://docs.mongodb.com/database-tools/mongoimport/.

- Import a JSON file into a specific collection:

```
mongoimport --file {{path/to/file.json}} --uri
{{mongodb_uri}} {{[-c|--collection]}} {{collection_name}}
```

- Import a CSV file, using the first line of the file to determine field names:

```
mongoimport --type {{csv}} --file {{path/to/file.csv}} {{[-
d|--db]}} {{database_name}} {{[-c|--collection]}}
{{collection_name}}
```

- Import a JSON array, using each element as a separate document:

```
mongoimport --jsonArray --file {{path/to/file.json}}
```

- Import a JSON file using a specific mode and a query to match existing documents:

```
mongoimport --file {{path/to/file.json}} --mode {{delete|
merge|upsert}} --upsertFields "{{field1,field2,...}}"
```

- Import a CSV file, reading field names from a separate CSV file and ignoring fields with empty values:

```
mongoimport --type {{csv}} --file {{path/to/file.csv}} --
fieldFile {{path/to/field_file.csv}} --ignoreBlanks
```

- Display help:

```
mongoimport --help
```

mongorestore

Utility to import a collection or database from a binary dump into a MongoDB instance.

More information: https://docs.mongodb.com/database-tools/mongorestore/.

- Import a BSON data dump from a directory to a MongoDB database:

```
mongorestore --db {{database_name}} {{path/to/directory}}
```

- Import a BSON data dump from a directory to a given database in a MongoDB server host, running at a given port, with user authentication (user will be prompted for password):

```
mongorestore --host {{database_host:port}} --db
{{database_name}} --username {{username}} {{path/to/
directory}} --password
```

- Import a collection from a BSON file to a MongoDB database:

```
mongorestore --db {{database_name}} {{path/to/file}}
```

- Import a collection from a BSON file to a given database in a MongoDB server host, running at a given port, with user authentication (user will be prompted for password):

```
mongorestore --host {{database_host:port}} --db
{{database_name}} --username {{username}} {{path/to/file}} --
password
```

mongosh

A new shell for MongoDB, replacement for **`mongo`**.

Note: All connection options can be replaced with one string: **`mongodb://user@host:port/db_name?authSource=authdb_name`**.

More information: https://www.mongodb.com/docs/mongodb-shell.

- Connect to a local database on the default port (`mongodb://localhost:27017`):

  ```
  mongosh
  ```

- Connect to a database:

  ```
  mongosh --host {{host}} --port {{port}} {{db_name}}
  ```

- Authenticate using the specified username on the specified database (you will be prompted for a password):

  ```
  mongosh --host {{host}} --port {{port}} --username {{username}} --authenticationDatabase {{authdb_name}} {{db_name}}
  ```

- Evaluate a JavaScript expression on a database:

  ```
  mongosh --eval '{{JSON.stringify(db.foo.findOne())}}' {{db_name}}
  ```

monodevelop

Cross platform IDE for C#, F# and more.

More information: https://www.monodevelop.com/documentation/.

- Start MonoDevelop:

```
monodevelop
```

- Open a specific file:

```
monodevelop {{path/to/file}}
```

- Open a specific file with the caret at a specific position:

```
monodevelop {{path/to/file}};{{line_number}};
{{column_number}}
```

- Force opening a new window instead of switching to an existing one:

```
monodevelop --new-window
```

- Disable redirection of `stdout` and `stderr` to a log file:

```
monodevelop --no-redirect
```

- Enable performance monitoring:

```
monodevelop --perf-log
```

monodis

The Mono Common Intermediate Language (CIL) disassembler.

More information: https://www.mono-project.com/docs/tools+libraries/tools/monodis/.

- Disassemble an assembly to textual CIL:

```
monodis {{path/to/assembly.exe}}
```

- Save the output to a file:

```
monodis --output={{path/to/output.il}} {{path/to/assembly.exe}}
```

- Show information about an assembly:

```
monodis --assembly {{path/to/assembly.dll}}
```

- List the references of an assembly:

```
monodis --assemblyref {{path/to/assembly.exe}}
```

- List all the methods in an assembly:

```
monodis --method {{path/to/assembly.exe}}
```

- List resources embedded within an assembly:

```
monodis --manifest {{path/to/assembly.dll}}
```

- Extract all the embedded resources to the current directory:

```
monodis --mresources {{path/to/assembly.dll}}
```

monolith

Save a web page as a single HTML file.

More information: https://github.com/Y2Z/monolith.

- Save a webpage as a single HTML file:

```
monolith {{url}}
```

- Save a webpage as a single HTML file, excluding audio:

```
monolith {{url}} {{[-a|--no-audio]}}
```

- Save a webpage as a single HTML file, excluding CSS:

```
monolith {{url}} {{[-c|--no-css]}}
```

- Save a webpage as a single HTML file, excluding images:

```
monolith {{url}} {{[-i|--no-images]}}
```

- Save a webpage as a single HTML file, excluding videos:

```
monolith {{url}} {{[-v|--no-video]}}
```

- Save a webpage as a single HTML file, excluding JavaScript:

```
monolith {{url}} {{[-j|--no-js]}}
```

- Save a webpage as a single HTML file, accepting invalid TLS certificates:

```
monolith {{url}} {{[-k|--insecure]}}
```

- Save a webpage as a single HTML file, specifying a specific output file:

```
monolith {{url}} {{[-o|--output]}} {{path/to/file.html}}
```

monop

Finds and displays signatures of Types and methods inside .NET assemblies.

More information: https://manned.org/monop.

- Show the structure of a Type built-in of the .NET Framework:

```
monop {{System.String}}
```

- List the types in an assembly:

```
monop -r:{{path/to/assembly.exe}}
```

- Show the structure of a Type in a specific assembly:

```
monop -r:{{path/to/assembly.dll}} {{Namespace.Path.To.Type}}
```

- Only show members defined in the specified Type:

```
monop -r:{{path/to/assembly.dll}} {{[-d|--declared-only]}}
{{Namespace.Path.To.Type}}
```

- Show private members:

```
monop -r:{{path/to/assembly.dll}} {{[-p|--private]}}
{{Namespace.Path.To.Type}}
```

- Hide obsolete members:

```
monop -r:{{path/to/assembly.dll}} {{[-f|--filter-obsolete]}}
{{Namespace.Path.To.Type}}
```

- List the other assemblies that a specified assembly references:

```
monop -r:{{path/to/assembly.dll}} --refs
```

montage

This command is an alias of **`magick montage`**.

- View documentation for the original command:

```
tldr magick montage
```

more

Interactively display a file, allowing scrolling and searching.

See also: **`less`**.

More information: https://manned.org/more.1p.

- Open a file:

```
more {{path/to/file}}
```

- Display a specific line:

```
more +{{line_number}} {{path/to/file}}
```

- Go to the next page:

```
<Space>
```

- Search for a string (press <n> to go to the next match):

```
</>{{something}}<Enter>
```

- Exit:

```
<q>
```

- Display help about interactive commands:

```
<h>
```

moreutils

A collection of UNIX tools.

Note: **`moreutils`** isn't a command, but a set of commands.

More information: https://joeyh.name/code/moreutils/.

- View documentation for pages related to standard streams:

```
tldr {{ifne|mispipe|pee|sponge|vipe|vidir}}
```

- View documentation for other pages:

```
tldr {{combine|errno|ifdata|isutt8|lckdo|parallel|zrun}}
```

moro

Track work time.

More information: https://moro.js.org.

- Invoke `moro` without parameters, to set the current time as the start of the working day:

  ```
  moro
  ```

- Specify a custom time for the start of the working day:

  ```
  moro hi {{09:30}}
  ```

- Invoke `moro` without parameters a second time, to set the current time at the end of the working day:

  ```
  moro
  ```

- Specify a custom time for the end of the working day:

  ```
  moro bye {{17:30}}
  ```

- Add a note on the current working day:

  ```
  moro note {{3 hours on project Foo}}
  ```

- Show a report of time logs and notes for the current working day:

  ```
  moro report
  ```

- Show a report of time logs and notes for all working days on record:

  ```
  moro report --all
  ```

mosh

Mobile Shell (**mosh**) is a robust and responsive replacement for SSH.

mosh persists connections to remote servers while roaming between networks.

More information: https://manned.org/mosh.

- Connect to a remote server:

```
mosh {{username}}@{{remote_host}}
```

- Connect to a remote server with a specific identity (private key):

```
mosh --ssh="ssh -i {{path/to/key_file}}" {{username}}
@{{remote_host}}
```

- Connect to a remote server using a specific port:

```
mosh --ssh="ssh -p {{2222}}" {{username}}@{{remote_host}}
```

- Run a command on a remote server:

```
mosh {{remote_host}} -- {{command -with -flags}}
```

- Select Mosh UDP port (useful when `remote_host` is behind a NAT):

```
mosh -p {{124}} {{username}}@{{remote_host}}
```

- Usage when `mosh-server` binary is outside standard path:

```
mosh --server={{path/to/mosh-server}} {{remote_host}}
```

mosquitto

An MQTT broker.

More information: https://mosquitto.org/man/mosquitto-8.html.

- Start Mosquitto:

```
mosquitto
```

- Specify a configuration file to use:

```
mosquitto {{[-c|--config-file]}} {{path/to/file.conf}}
```

- Listen on a specific port:

```
mosquitto {{[-p|--port]}} {{8883}}
```

- Daemonize by forking into the background:

```
mosquitto {{[-d|--daemon]}}
```

mosquitto_passwd

Manage password files for mosquitto.

See also: **`mosquitto`**, the MQTT server that this manages.

More information: https://mosquitto.org/man/mosquitto_passwd-1.html.

- Add a new user to a password file (will prompt to enter the password):

```
mosquitto_passwd {{path/to/password_file}} {{username}}
```

- Create the password file if it doesn't already exist:

```
mosquitto_passwd -c {{path/to/password_file}} {{username}}
```

- Delete the specified username instead:

```
mosquitto_passwd -D {{path/to/password_file}} {{username}}
```

- Upgrade an old plain-text password file to a hashed password file:

```
mosquitto_passwd -U {{path/to/password_file}}
```

mosquitto_pub

A simple MQTT version 3.1.1 client that will publish a single message on a topic and exit.

More information: https://mosquitto.org/man/mosquitto_pub-1.html.

- Publish a temperature value of 32 on the topic `sensors/temperature` to 192.168.1.1 (defaults to `localhost`) with Quality of Service (`QoS`) set to 1:

```
mosquitto_pub {{[-h|--host]}} {{192.168.1.1}} {{[-t|--
topic]}} {{sensors/temperature}} {{[-m|--message]}} {{32}}
{{[-q|--qos]}} {{1}}
```

- Publish timestamp and temperature data on the topic `sensors/temperature` to a remote host on a non-standard port:

```
mosquitto_pub {{[-h|--host]}} {{192.168.1.1}} {{[-p|--port]}}
{{1885}} {{[-t|--topic]}} {{sensors/temperature}} {{[-m|--
message]}} "{{1266193804 32}}"
```

- Publish light switch status and retain the message on the topic `switches/kitchen_lights/status` to a remote host because there may be a long period of time between light switch events:

```
mosquitto_pub {{[-r|--retain]}} {{[-h|--host]}}
"{{iot.eclipse.org}}" {{[-t|--topic]}} {{switches/
kitchen_lights/status}} {{[-m|--message]}} "{{on}}"
```

- Send the contents of a file (`data.txt`) as a message and publish it to `sensors/temperature` topic:

```
mosquitto_pub {{[-t|--topic]}} {{sensors/temperature}} {{[-
f|--file]}} {{data.txt}}
```

- Send the contents of a file (`data.txt`), by reading from `stdin` and send the entire input as a message and publish it to `sensors/temperature` topic:

```
mosquitto_pub < {{data.txt}} {{[-t|--topic]}} {{sensors/
temperature}} {{[-s|--stdin-file]}}
```

- Read newline delimited data from `stdin` as a message and publish it to `sensors/temperature` topic:

```
{{echo data.txt}} | mosquitto_pub {{[-t|--topic]}} {{sensors/
temperature}} {{[-l|--stdin-line]}}
```

mosquitto_sub

A simple MQTT version 3.1.1 client that will subscribe to topics and print the messages that it receives.

More information: https://mosquitto.org/man/mosquitto_sub-1.html.

- Subscribe to the topic `sensors/temperature` information with Quality of Service (`QoS`) set to 1. (The default hostname is `localhost` and port 1883):

```
mosquitto_sub {{[-t|--topic]}} {{sensors/temperature}} {{[-q|--qos]}} {{1}}
```

- Subscribe to all broker status messages publishing on `iot.eclipse.org` port 1885 and print published messages verbosely:

```
mosquitto_sub {{[-v|--verbose]}} {{[-h|--host]}} "iot.eclipse.org" {{[-p|--port]}} 1885 {{[-t|--topic]}} {{\$SYS/#}}
```

- Subscribe to multiple topics matching a given pattern. (+ takes any metric name):

```
mosquitto_sub {{[-t|--topic]}} {{sensors/machines/+/temperature/+}}
```

most

Open one or several files for interactive reading, allowing scrolling and search.

More information: https://manned.org/most.

- Open a file:

```
most {{path/to/file}}
```

- Open several files:

```
most {{path/to/file1 path/to/file2 ...}}
```

- Open a file at the first occurrence of "string":

```
most {{path/to/file}} +/{{string}}
```

- Move through opened files:

```
<:><n>{{<ArrowUp>|<ArrowDown>}}
```

- Jump to the 100th line:

```
<j>{{100}}<Enter>
```

- Edit current file:

```
<e>
```

- Split the current window in half:

```
<CTRL x><o>
```

- Exit:

```
<q>
```

mozillavpn

A Virtual Private Network from the makers of Firefox.

See also: **`fastd`**, **`ivpn`**, **`mullvad`**, **`warp-cli`**.

More information: https://github.com/mozilla-mobile/mozilla-vpn-client/wiki/Command-line-interface.

- Log in with an interactive prompt:

```
mozillavpn login
```

- Connect to Mozilla VPN:

```
mozillavpn activate
```

- Display the connection status:

```
mozillavpn status
```

- List available servers:

```
mozillavpn servers
```

- Select a specific server:

```
mozillavpn select {{server_name}}
```

- Disconnect from Mozilla VPN:

```
mozillavpn deactivate
```

- Log out:

```
mozillavpn logout
```

- Display help for a subcommand:

```
mozillavpn {{subcommand}} --help
```

mp3info

Viewer/editor for ID3v1 (but not ID3v2) tags of MP3 files.

More information: https://www.ibiblio.org/mp3info/mp3info.html.

- Show all ID3v1 tags of a specific MP3 file:

```
mp3info {{path/to/file.mp3}}
```

- Edit ID3v1 tags [i]nteractively:

```
mp3info -i {{path/to/file.mp3}}
```

- Set values for ID3v1 tags in a specific MP3 file ([a]rtist, [t]itle, a[l]bum, [y]ear, and [c]omment):

```
mp3info -a "{{artist_name}}" -t "{{song_title}}" -l
"{{album_title}}" -y {{year}} -c "{{comment_text}}" {{path/
to/file.mp3}}
```

- Set the [n]umber of the track in the album for a specific MP3 file:

```
mp3info -n {{track_number}} {{path/to/file.mp3}}
```

- [G]et a list of valid genres and their numeric codes:

```
mp3info -G
```

- Set the music [g]enre for a specific MP3 file:

```
mp3info -g {{genre_number}} {{path/to/file.mp3}}
```

mp4box

MPEG-4 Systems Toolbox: muxe streams into MP4 container.

More information: https://gpac.wp.imt.fr/mp4box.

- Display information about an existing MP4 file:

```
mp4box -info {{path/to/file}}
```

- Add an SRT subtitle file into an MP4 file:

```
mp4box -add {{input_subs.srt}}:lang=eng -add {{input.mp4}} {{output.mp4}}
```

- Combine audio from one file and video from another:

```
mp4box -add {{input1.mp4}}#audio -add {{input2.mp4}}#video {{output.mp4}}
```

mpc

Music Player Client: control the Music Player Daemon (MPD).

See also: **`mpd`**, **`ncmpcpp`**, **`cmus`**.

More information: https://www.musicpd.org/doc/mpc/html.

- Toggle play/pause:

```
mpc toggle
```

- Stop playing:

```
mpc stop
```

- Show information about the currently playing song:

```
mpc status
```

- Play the next song:

```
mpc next
```

- Play the previous song:

```
mpc prev
```

- Seek `n` seconds forward (`+`) or backward (`-`):

```
mpc {{+n|-n}}
```

mpd

Music Player Daemon.

See also: **`mpc`**, **`ncmpcpp`**.

More information: https://mpd.readthedocs.io/en/stable/mpd.1.html.

- Start MPD:

```
mpd
```

- Start MPD but don't read from the configuration file:

```
mpd --no-config
```

- Start MPD and don't detach it from the console:

```
mpd --no-daemon
```

- Kill the currently running MPD session:

```
mpd --kill
```

mpg321

High Performance MPEG 1.0/2.0/2.5 Audio Player for Layer 1, 2, and 3.

Mpg321 was written (sometime in 1999) to be a drop-in replacement for the (previously) non-free mpg123 player.

More information: https://manned.org/mpg321.

- Play an audio source exactly `n` times (0 means forever):

```
mpg321 {{[-l|--loop]}} {{n}} {{path/to/file_a|URL path/to/file_b|URL ...}}
```

- Play a directory recursively:

```
mpg321 -B {{path/to/directory}}
```

- Enable Basic Keys (`*` or `/` - Increase or decrease volume, `n` - Skip song, `m` - Mute/unmute.) while playing:

```
mpg321 -K {{path/to/file_a|URL path/to/file_b|URL ...}}
```

- Play files randomly until interrupted:

```
mpg321 {{[-Z|--random]}} {{path/to/file_a|URL path/to/file_b|URL ...}}
```

- Shuffle the files before playing them once:

```
mpg321 {{[-z|--shuffle]}} {{path/to/file_a|URL path/to/file_b|URL ...}}
```

- Play all files in the current directory and subdirectories, randomly (until interrupted), with Basic Keys enabled:

```
mpg321 -B {{[-Z|--random]}} -K .
```

mpic++

Open MPI wrapper compiler for C++.

See also: **`mpirun`**.

More information: https://manned.org/mpicxx.

- Compile an Open MPI program:

```
mpic++ {{path/to/source_file}}
```

- Show all the wrapper-supplied flags:

```
mpic++ --showme
```

mpicc

Open MPI C wrapper compiler.

More information: https://www.mpich.org/static/docs/latest/www1/mpicc.html.

- Compile a source code file into an object file:

```
mpicc -c {{path/to/file.c}}
```

- Link an object file and make an executable:

```
mpicc -o {{executable}} {{path/to/object_file.o}}
```

- Compile and link source code in a single command:

```
mpicc -o {{executable}} {{path/to/file.c}}
```

mpicxx

This command is an alias of **`mpic++`**.

- View documentation for the original command:

```
tldr mpic++
```

mpiexec

This command is an alias of **`mpirun`**.

- View documentation for the original command:

  ```
  tldr mpirun
  ```

mpijavac

Open MPI compiler wrapper for Java.

See also: **`mpirun`**.

More information: https://docs.open-mpi.org/en/main/features/java.html#building-java-mpi-applications.

- Compile a Java source file:

```
mpijavac {{path/to/source_file.java}}
```

- Pass application-specific classpaths to compiler:

```
mpijavac -cp {{path/to/app.jar}} {{path/to/source_file.java}}
```

- Show the flags necessary to build MPI Java applications:

```
mpijavac --showme
```

- Show the flags necessary to compile MPI Java applications:

```
mpijavac --showme:compile
```

- Show full invoked Java compiler command line:

```
mpijavac {{path/to/source_file.java}} --showme
```

mpirun

Execute serial and parallel jobs in Open MPI.

See also: **`mpic++`**.

More information: https://docs.open-mpi.org/en/main/man-openmpi/man1/mpirun.1.html.

- Execute an Open MPI program:

```
mpirun {{path/to/executable}}
```

- Execute an Open MPI program with `n` parallel processes:

```
mpirun -n {{n}} {{path/to/executable}}
```

- Allow more processes than available physical cores:

```
mpirun -oversubscribe {{path/to/executable}}
```

mplayer

Cross-platform multimedia player.

More information: https://mplayerhq.hu/DOCS/HTML/en/commandline.html.

- Play the specified file or URL:

```
mplayer {{path/to/file|url}}
```

- Play multiple files:

```
mplayer {{path/to/file1 path/to/file2 ...}}
```

- Play a specific file repeatedly:

```
mplayer -loop {{0}} {{path/to/file}}
```

- Pause playback:

```
<Space>
```

- Quit mplayer:

```
<Esc>
```

- Seek backward or forward 10 seconds:

```
{{<ArrowLeft>|<ArrowRight>}}
```

mpremote

Remotely control MicroPython devices.

More information: https://docs.micropython.org/en/latest/reference/mpremote.html.

- List all connected MicroPython devices:

```
mpremote connect list
```

- Open an interactive REPL session with a connected device:

```
mpremote connect {{device}}
```

- Run a local script on a connected device:

```
mpremote run {{path/to/script.py}}
```

- Mount a local directory to the device:

```
mpremote mount {{path/to/directory}}
```

- Install a mip package on the device:

```
mpremote mip install {{package}}
```

mpv

A audio/video player based on MPlayer.

See also: **`mplayer`**, **`vlc`**.

More information: https://mpv.io/manual/stable/.

- Play a video or audio from a URL or file:

```
mpv {{url|path/to/file}}
```

- Jump backward/forward 5 seconds:

```
{{<ArrowLeft>|<ArrowRight>}}
```

- Jump backward/forward 1 minute:

```
{{<ArrowDown>|<ArrowUp>}}
```

- Decrease or increase playback speed by 10%:

```
{{<[>|<]>}}
```

- Add subtitles from a file:

```
mpv --sub-file={{path/to/file}}
```

- Take a screenshot of the current frame (saved to `./mpv-shotNNNN.jpg` by default):

```
<s>
```

- Play a file at a specified speed (1 by default):

```
mpv --speed {{0.01..100}} {{path/to/file}}
```

- Play a file using a profile defined in the `mpv.conf` file:

```
mpv --profile {{profile_name}} {{path/to/file}}
```

mqtt_check.py

Simple utility for testing and validating MQTT login credentials.

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- Check MQTT login credentials for a target (MQTT broker's hostname):

```
mqtt_check.py {{domain}}/{{username}}:{{password}}
@{{targetName}}
```

- Specify a custom client ID for authentication:

```
mqtt_check.py -client-id {{client_id}} {{domain}}/
{{username}}:{{password}}@{{targetName}}
```

- Enable SSL for the connection:

```
mqtt_check.py -ssl {{domain}}/{{username}}:{{password}}
@{{targetName}}
```

- Connect to a specific port (default is 1883):

```
mqtt_check.py -port {{port}} {{domain}}/{{username}}:
{{password}}@{{targetName}}
```

- Enable debug output:

```
mqtt_check.py -debug {{domain}}/{{username}}:{{password}}
@{{targetName}}
```

- Display help:

```
mqtt_check.py --help
```

mr

Manage all of your version control repositories at once.

More information: https://myrepos.branchable.com.

- Register a repository:

```
mr register
```

- Update repositories in 5 concurrent jobs:

```
mr {{[-j|--jobs]}} {{5}} update
```

- Print the status of all repositories:

```
mr status
```

- Checkout all repositories to the latest version:

```
mr checkout
```

msbuild

The Microsoft build tool for Visual Studio project solutions.

More information: https://learn.microsoft.com/visualstudio/msbuild.

- Build the first project file in the current directory:

```
msbuild
```

- Build a specific project file:

```
msbuild {{path/to/project_file}}
```

- Specify one or more semicolon-separated targets to build:

```
msbuild {{path/to/project_file}} /target:{{targets}}
```

- Specify one or more semicolon-separated properties:

```
msbuild {{path/to/project_file}} /property:{{name=value}}
```

- Specify the build tools version to use:

```
msbuild {{path/to/project_file}} /toolsversion:{{version}}
```

- Display detailed information at the end of the log about how the project was configured:

```
msbuild {{path/to/project_file}} /detailedsummary
```

- Display help:

```
msbuild /help
```

mscore

This command is an alias of **musescore**.

- View documentation for the original command:

```
tldr musescore
```

msedge

The Microsoft Edge command-line utility is available as **`msedge`** on Windows and **`microsoft-edge`** for other platforms.

More information: https://microsoft.com/edge.

- View the documentation for Microsoft Edge for Windows:

```
tldr {{[-p|--platform]}} windows msedge
```

- View the documentation for Microsoft Edge for other platforms:

```
tldr {{[-p|--platform]}} common microsoft-edge
```

msfconsole

Console for the Metasploit Framework.

More information: https://docs.rapid7.com/metasploit/msf-overview.

- Launch the console:

```
msfconsole
```

- Launch the console quietly without any banner:

```
msfconsole {{[-q|--quiet]}}
```

- Do not enable database support:

```
msfconsole {{[-n|--no-database]}}
```

- Execute console commands (Note: Use `;` for passing multiple commands):

```
msfconsole {{[-x|--execute-command]}} "{{use auxiliary/
server/capture/ftp; set SRVHOST 0.0.0.0; set SRVPORT 21;
run}}"
```

- Display help:

```
msfconsole {{[-h|--help]}}
```

- Display version:

```
msfconsole {{[-v|--version]}}
```

msfvenom

Manually generate payloads for metasploit.

More information: https://docs.metasploit.com/docs/using-metasploit/basics/how-to-use-msfvenom.html.

- List payloads:

```
msfvenom {{[-l|--list]}} payloads
```

- List formats:

```
msfvenom {{[-l|--list]}} formats
```

- Show payload options:

```
msfvenom {{[-p|--payload]}} {{payload}} --list-options
```

- Create an ELF binary with a reverse TCP handler:

```
msfvenom {{[-p|--payload]}} linux/x64/meterpreter/reverse_tcp
LHOST={{local_ip}} LPORT={{local_port}} {{[-f|--format]}} elf
{{[-o|--out]}} {{path/to/binary}}
```

- Create an EXE binary with a reverse TCP handler:

```
msfvenom {{[-p|--payload]}} windows/x64/meterpreter/
reverse_tcp LHOST={{local_ip}} LPORT={{local_port}} {{[-f|--
format]}} exe {{[-o|--out]}} {{path/to/binary.exe}}
```

- Create a raw Bash with a reverse TCP handler:

```
msfvenom {{[-p|--payload]}} cmd/unix/reverse_bash
LHOST={{local_ip}} LPORT={{local_port}} {{[-f|--format]}} raw
```

msgattrib

Filter and manipulate message attributes in **`.po`** translation files.

More information: https://www.gnu.org/software/gettext/manual/gettext.html#msgattrib-Invocation.

- Keep only translated messages:

```
msgattrib --translated {{input.po}} > {{translated.po}}
```

- Keep only untranslated messages:

```
msgattrib --untranslated {{input.po}} > {{untranslated.po}}
```

- Remove fuzzy messages:

```
msgattrib --no-fuzzy {{input.po}} > {{clean.po}}
```

- Keep only fuzzy messages:

```
msgattrib --only-fuzzy {{input.po}} > {{fuzzy.po}}
```

- Mark all messages as fuzzy:

```
msgattrib --set-fuzzy {{input.po}} > {{fuzzy.po}}
```

- Clear fuzzy marks:

```
msgattrib --clear-fuzzy {{input.po}} > {{clean.po}}
```

- Sort output by file location:

```
msgattrib {{[-F|--sort-by-file]}} {{input.po}} > {{sorted.po}}
```

- Display help:

```
msgattrib {{[-h|--help]}}
```

msgcat

Concatenate and merge multiple **`.po`** translation files.

Useful in software localization pipelines to combine message catalogs with filtering options.

More information: https://www.gnu.org/software/gettext/manual/gettext.html#msgcat-Invocation.

- Combine multiple `.po` files into one:

```
msgcat {{file1.po file2.po ...}} {{[-o|--output-file]}} {{combined.po}}
```

- Combine input files listed in a text file:

```
msgcat {{[-f|--files-from]}} {{path/to/file_list.txt}} {{[-o|--output-file]}} {{combined.po}}
```

- Set the output encoding (e.g. UTF-8):

```
msgcat {{[-t|--to-code]}} {{UTF-8}} {{input.po}} {{[-o|--output-file]}} {{output.po}}
```

- Output only unique messages (appearing in one file only):

```
msgcat {{[-u|--unique]}} {{file1.po file2.po ...}} {{[-o|--output-file]}} {{unique.po}}
```

- Use the first available translation for duplicate entries:

```
msgcat --use-first {{file1.po file2.po ...}} {{[-o|--output-file]}} {{output.po}}
```

- Display help:

```
msgcat {{[-h|--help]}}
```

msgfmt

Compile message catalog to binary format.

More information: https://www.gnu.org/software/gettext/manual/gettext.html#msgfmt-Invocation.

- Compile a file to `messages.mo`:

```
msgfmt {{file.po}}
```

- Convert a `.po` file to a `.mo` file:

```
msgfmt {{path/to/file.po}} {{[-o|--output-file]}} {{path/to/file.mo}}
```

- Display help:

```
msgfmt {{[-h|--help]}}
```

msginit

Generate language specific translation files based on Portable Object Templates.

More information: https://www.gnu.org/software/gettext/manual/gettext.html#msginit-Invocation.

- Generate Portable Object files in system locale from `messages.pot`:

```
msginit
```

- Define locale to generate from a specific template:

```
msginit {{[-l|--locale]}} {{locale}} {{[-i|--input]}} {{path/to/messages.pot}}
```

- Display help:

```
msginit {{[-h|--help]}}
```

msgmerge

Update an existing translation file from a new template.

More information: https://www.gnu.org/software/gettext/manual/gettext.html#msgmerge-Invocation.

- Update a translation file:

```
msgmerge {{[-U|--update]}} {{path/to/file.po}} {{path/to/messages.pot}}
```

- Display help:

```
msgmerge {{[-h|--help]}}
```

msgunfmt

Decompile message catalog from the binary format.

More information: https://www.gnu.org/software/gettext/manual/gettext.html#msgunfmt-Invocation.

- Output conversion:

```
msgunfmt {{path/to/file.mo}}
```

- Convert a `.mo` file to a `.po` file:

```
msgunfmt {{path/to/file.mo}} > {{path/to/file.po}}
```

msmtp

An SMTP client.

It reads text from **`stdin`** and sends it to an SMTP server.

More information: https://marlam.de/msmtp.

- Send an email using the default account configured in `~/.msmtprc`:

```
echo "{{Hello world}}" | msmtp {{to@example.org}}
```

- Send an email using a specific account configured in `~/.msmtprc`:

```
echo "{{Hello world}}" | msmtp --account={{account_name}}
{{to@example.org}}
```

- Send an email without a configured account. The password should be specified in the `~/.msmtprc` file:

```
echo "{{Hello world}}" | msmtp --host={{localhost}} --
port={{999}} --from={{from@example.org}} {{to@example.org}}
```

mssqlclient.py

Connect to Microsoft SQL Server instances and execute queries.

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- Connect to an MSSQL server using Windows authentication:

```
mssqlclient.py -windows-auth {{domain}}/{{username}}:{{password}}@{{target}}
```

- Connect using SQL server authentication:

```
mssqlclient.py {{username}}:{{password}}@{{target}}
```

- Connect using pass-the-hash authentication:

```
mssqlclient.py {{domain}}/{{username}}@{{target}} -hashes {{LM_Hash}}:{{NT_Hash}}
```

- Connect using Kerberos authentication (requires valid tickets):

```
mssqlclient.py -k {{domain}}/{{username}}@{{target}}
```

- Execute a specific SQL command upon connection:

```
mssqlclient.py {{username}}:{{password}}@{{target}} -query "{{SELECT user_name();}}"
```

- Execute multiple SQL commands from a file:

```
mssqlclient.py {{username}}:{{password}}@{{target}} -file {{path/to/sql_file.sql}}
```

- Connect to a specific database instance (default is `None`):

```
mssqlclient.py {{username}}:{{password}}@{{target}} -db {{database_name}}
```

- Display SQL queries before execution:

```
mssqlclient.py {{username}}:{{password}}@{{target}} -show
```

mtm

Minimal terminal multiplexer.

See also: **`tmux`**, **`screen`**.

More information: https://github.com/deadpixi/mtm.

- Start the program with default command chord (`<Ctrl g>`):

```
mtm
```

- Use `<Ctrl q>` as the command chord:

```
mtm -c {{q}}
```

- Split and stack horizontally:

```
<Ctrl g><h>
```

- Split and stack vertically:

```
<Ctrl g><v>
```

- View scrollback buffer:

```
<Ctrl g><PageUp>
```

- Switch terminals:

```
<Ctrl g><ArrowKeys>
```

- Close current terminal:

```
<Ctrl g><w>
```

mtr

Matt's Traceroute: combined traceroute and ping tool.

More information: https://manned.org/mtr.

- Traceroute to a host and continuously ping all intermediary hops:

```
mtr {{example.com}}
```

- Disable IP address and host name mapping:

```
mtr {{[-n|--no-dns]}} {{example.com}}
```

- Generate output after pinging each hop 10 times:

```
mtr {{[-w|--report-wide]}} {{example.com}}
```

- Force IP IPv4 or IPV6:

```
mtr -4 {{example.com}}
```

- Wait for a given time (in seconds) before sending another packet to the same hop:

```
mtr {{[-i|--interval]}} {{10}} {{example.com}}
```

- Display the Autonomous System Number (ASN) for each hop:

```
mtr {{[-z|--aslookup]}} {{example.com}}
```

- Display both IP address and reverse DNS name:

```
mtr {{[-b|--show-ips]}} {{example.com}}
```

mtvtoppm

Convert an MTV or PRT ray tracer file to a PPM image.

More information: https://netpbm.sourceforge.net/doc/mtvtoppm.html.

- Convert an MTV or PRT ray tracer file to a PPM image:

```
mtvtoppm {{path/to/file.mtv}} > {{path/to/output.ppm}}
```

mu

Index and search emails from a local Maildir.

More information: https://man.cx/mu.

- Initialize the email database, optionally specifying the Maildir directory and email addresses:

```
mu init --maildir={{path/to/directory}} --my-address={{name@example.com}}
```

- Index new emails:

```
mu index
```

- Find messages using a specific keyword (in message body, subject, sender, ...):

```
mu find {{keyword}}
```

- Find messages to Alice with subject `jellyfish` containing the words `apples` or `oranges`:

```
mu find to:{{alice}} subject:{{jellyfish}} {{apples}} OR {{oranges}}
```

- Find unread messages about words starting with `soc` (the `*` only works at the end of the search term) in the Sent Items folder:

```
mu find 'subject:{{soc}}*' flag:{{unread}} maildir:'/{{Sent Items}}'
```

- Find messages from Sam with attached images, between 2 KiB and 2 MiB, written in 2021:

```
mu find 'mime:{{image/*}} size:{{2k..2m}} date:{{20210101..20211231}} from:{{sam}}
```

- List contacts with `Bob` in either name or email address:

```
mu cfind {{Bob}}
```

mullvad

CLI client for Mullvad VPN.

See also: **`fastd`**, **`ivpn`**, **`mozillavpn`**, **`warp-cli`**.

More information: https://mullvad.net/en/help/how-use-mullvad-cli.

- Link your Mullvad account with the specified account number:

```
mullvad account set {{account_number}}
```

- Enable LAN access while VPN is on:

```
mullvad lan set allow
```

- Select a server in a specific city:

```
mullvad relay set location {{se}} {{mma}}
```

- Select a specific server:

```
mullvad relay set location {{se-mma-wg-001}}
```

- Establish the VPN tunnel:

```
mullvad connect
```

- Check status of VPN tunnel:

```
mullvad status
```

- Check the account expiration date and obtain the device name:

```
mullvad account get
```

multipass

Manage Ubuntu virtual machines using native hypervisors.

More information: https://documentation.ubuntu.com/multipass/en/latest/reference/command-line-interface/.

- List the aliases that can be used to launch an instance:

```
multipass find
```

- Launch a new instance, set its name and use a cloud-init configuration file:

```
multipass launch {{[-n|--name]}} {{instance_name}} --cloud-init {{configuration_file}}
```

- List all the created instances and some of their properties:

```
multipass list
```

- Start a specific instance by name:

```
multipass start {{instance_name}}
```

- Show the properties of an instance:

```
multipass info {{instance_name}}
```

- Open a shell prompt on a specific instance by name:

```
multipass shell {{instance_name}}
```

- Delete an instance by name:

```
multipass delete {{instance_name}}
```

- Mount a directory into a specific instance:

```
multipass mount {{path/to/local_directory}} {{instance_name}}:{{path/to/target_directory}}
```

multitail

Extension of tail.

More information: https://manned.org/multitail.

- Tail all files matching a pattern in a single stream:

```
multitail -Q 1 '{{pattern}}'
```

- Tail all files in a directory in a single stream:

```
multitail -Q 1 '{{path/to/directory}}/*'
```

- Automatically add new files to a window:

```
multitail -Q {{pattern}}
```

- Show 5 logfiles while merging 2 and put them in 2 columns with only one in the left column:

```
multitail -s 2 -sn 1,3 {{path/to/mergefile}} -I {{path/to/file1}} {{path/to/file2}} {{path/to/file3}} {{path/to/file4}}
```

mumble

Low-latency, high quality voice chat software.

More information: https://www.mumble.info.

- Open Mumble:

```
mumble
```

- Open Mumble and immediately connect to a server:

```
mumble mumble://{{username}}@{{example.com}}
```

- Open Mumble and immediately connect to a password protected server:

```
mumble mumble://{{username}}:{{password}}@{{example.com}}
```

- Mute/unmute the microphone in a running Mumble instance:

```
mumble rpc {{mute|unmute}}
```

- Mute/unmute the microphone and the audio output of Mumble:

```
mumble rpc {{deaf|undeaf}}
```

mupdf

A lightweight PDF, XPS, and E-book viewer.

More information: https://mupdf.readthedocs.io/en/latest/tools/mupdf-gl.html.

- Open a PDF on the first page:

```
mupdf {{path/to/file}}
```

- Open a PDF on page 3:

```
mupdf {{path/to/file}} {{3}}
```

- Open a password secured PDF:

```
mupdf -p {{password}} {{path/to/file}}
```

- Open a PDF with an initial zoom level, specified as DPI, of 72:

```
mupdf -r {{72}} {{path/to/file}}
```

- Open a PDF with inverted color:

```
mupdf -I {{path/to/file}}
```

- Open a PDF tinted red #FF0000 (hexadecimal color syntax RRGGBB):

```
mupdf -C {{FF0000}}
```

- Open a PDF without anti-aliasing (0 = off, 8 = best):

```
mupdf -A {{0}}
```

musescore

MuseScore 3 sheet music editor.

See also: **`lilypond`**.

More information: https://musescore.org/en/handbook/4/command-line-options.

- Use a specific audio driver:

```
musescore --audio-driver {{jack|alsa|portaudio|pulse}}
```

- Set the MP3 output bitrate in kbit/s:

```
musescore --bitrate {{bitrate}}
```

- Start MuseScore in debug mode:

```
musescore --debug
```

- Enable experimental features, such as layers:

```
musescore --experimental
```

- Export the given file to the specified output file. The file type depends on the given extension:

```
musescore --export-to {{output_file}} {{input_file}}
```

- Print a diff between the given scores:

```
musescore --diff {{path/to/file1}} {{path/to/file2}}
```

- Specify a MIDI import operations file:

```
musescore --midi-operations {{path/to/file}}
```

musl-gcc

A wrapper around **gcc** that automatically sets options for linking against musl libc.

All options specified are passed directly to **gcc**.

More information: https://manned.org/musl-gcc.

- View documentation for `gcc`:

```
tldr gcc
```

mutagen

Real-time file synchronization and network forwarding tool.

More information: https://mutagen.io/documentation/introduction/.

- Start a synchronization session between a local directory and a remote host:

```
mutagen sync create --name={{session_name}} /{{path/to/
local_directory}}/ {{user}}@{{host}}:/{{path/to/
remote_directory}}/
```

- Start a synchronization session between a local directory and a Docker container:

```
mutagen sync create --name={{session_name}} /{{path/to/
local_directory}}/ docker://{{user}}@{{container_name}}/
{{path/to/remote_directory}}/
```

- Stop a running session:

```
mutagen sync terminate {{session_name}}
```

- Start a project:

```
mutagen project start
```

- Stop a project:

```
mutagen project terminate
```

- List running sessions for the current project:

```
mutagen project list
```

mutool

Convert, query information and extract data from PDF files.

More information: https://mupdf.readthedocs.io/en/latest/tools/mutool.html.

- Convert a range of pages to PNGs (Note: `%nd` in the output placeholder must be replaced with a print modifier like `%d` or `%2d`):

```
mutool convert -o {{path/to/output%nd.png}} {{path/to/input.pdf}} {{1-10}}
```

- Convert one or more pages of a PDF into text in `stdout`:

```
mutool draw -F txt {{path/to/input.pdf}} {{2,3,5,...}}
```

- Concatenate multiple PDF files:

```
mutool merge -o {{path/to/output.pdf}} {{path/to/input1.pdf path/to/input2.pdf ...}}
```

- Query information about all content embedded in a PDF:

```
mutool info {{path/to/input.pdf}}
```

- Extract all images, fonts and resources embedded in a PDF to the current directory:

```
mutool extract {{path/to/input.pdf}}
```

- Show the outline (table of contents) of a PDF:

```
mutool show {{path/to/input.pdf}} outline
```

mutt

Command-line email client.

More information: http://mutt.org/doc/mutt.1.txt.

- Open the specified mailbox:

```
mutt -f {{mailbox}}
```

- Send an email and specify a subject and a cc recipient:

```
mutt -s {{subject}} -c {{cc@example.com}}
{{recipient@example.com}}
```

- Send an email with files attached:

```
mutt -a {{file1 file2 ...}} -- {{recipient@example.com}}
```

- Specify a file to include as the message body:

```
mutt -i {{path/to/file}} {{recipient@example.com}}
```

- Specify a draft file containing the header and the body of the message, in RFC 5322 format:

```
mutt -H {{path/to/file}} {{recipient@example.com}}
```

mv

Move or rename files and directories.

More information: https://www.gnu.org/software/coreutils/manual/html_node/mv-invocation.html.

- Rename a file or directory when the target is not an existing directory:

```
mv {{path/to/source}} {{path/to/target}}
```

- Move a file or directory into an existing directory:

```
mv {{path/to/source}} {{path/to/existing_directory}}
```

- Move multiple files into an existing directory, keeping the filenames unchanged:

```
mv {{path/to/source1 path/to/source2 ...}} {{path/to/existing_directory}}
```

- Do not prompt for confirmation before overwriting existing files:

```
mv {{[-f|--force]}} {{path/to/source}} {{path/to/target}}
```

- Prompt for confirmation interactively before overwriting existing files, regardless of file permissions:

```
mv {{[-i|--interactive]}} {{path/to/source}} {{path/to/target}}
```

- Do not overwrite existing files at the target:

```
mv {{[-n|--no-clobber]}} {{path/to/source}} {{path/to/target}}
```

- Move files in verbose mode, showing files after they are moved:

```
mv {{[-v|--verbose]}} {{path/to/source}} {{path/to/target}}
```

- Specify target directory so that you can use external tools to gather movable files:

```
{{find /var/log -type f -name '*.log' -print0}} | {{xargs -0}} mv {{[-t|--target-directory]}} {{path/to/target_directory}}
```

mvn archetype

Generate a new Maven project from a predefined template (archetype).

More information: https://maven.apache.org/archetype/maven-archetype-plugin/usage.html.

- Generate a new project interactively:

```
mvn archetype:generate
```

- Generate a project non-interactively with specific archetype and project parameters:

```
mvn archetype:generate --define archetypeGroupId={{group_id}}
--define archetypeArtifactId={{artifact_id}} --define
archetypeVersion={{version}} --define
groupId={{project_group_id}} --define
artifactId={{project_name}} --define interactiveMode=false
```

mvn compile

Compile a Maven project's source code.

More information: https://manned.org/mvn.

- Compile the project's source code:

```
mvn compile
```

- Clean compiled files and recompile:

```
mvn clean compile
```

- Compile a specific module in a multi-module project:

```
mvn compile {{[-pl|--projects]}} {{module_name}}
```

- Skip tests while compiling:

```
mvn compile {{[-D|--define]}} skipTests
```

mvn dependency

Manage and analyze Maven project dependencies.

Provides goals for viewing, resolving, and copying project dependencies.

More information: https://maven.apache.org/plugins/maven-dependency-plugin/usage.html.

- Display the full dependency tree, including direct and transitive dependencies:

  ```
  mvn dependency:tree
  ```

- Analyze the dependencies and highlight unused or undeclared ones:

  ```
  mvn dependency:analyze
  ```

- Copy all project dependencies (by default, to `target/dependency/`):

  ```
  mvn dependency:copy-dependencies
  ```

- Resolve and download all project dependencies to the local Maven repository:

  ```
  mvn dependency:resolve
  ```

- Force Maven to update all dependencies from remote repositories:

  ```
  mvn dependency:resolve {{[-U|--update-snapshots]}}
  ```

mvn deploy

Add an artifact to a remote repository.

More information: https://manned.org/mvn.

- Copy the final artifact into the remote repository configured in the `settings.xml` file:

```
mvn deploy
```

- Copy an artifact, that is not built using Maven to the remote repository:

```
mvn deploy:deploy-file {{[-D|--define]}}
url={{URLOfTheRemoteRepository}} {{[-D|--define]}}
repositoryId={{ServerIdFromSettingsXML}} {{[-D|--define]}}
file={{FileToBeDeployed}}
```

mvn generate-sources

Generate source code for a Maven project before the main compilation phase.

This phase runs after **`initialize`** and before **`process-sources`**.

More information: https://manned.org/mvn.

- Run all lifecycle phases up to `generate-sources`:

  ```
  mvn generate-sources
  ```

- Run the next phase to generate resources:

  ```
  mvn generate-resources
  ```

- Clean and regenerate sources:

  ```
  mvn clean generate-sources
  ```

mvn idea

Generate IntelliJ IDEA project files (**`.ipr`**, **`.iml`**, and **`.iws`**) for a Maven project.

Note: This plugin is retired. It is no longer maintained.

More information: https://maven.apache.org/plugins/maven-idea-plugin/usage.html.

- Generate all IntelliJ IDEA project files:

  ```
  mvn idea:idea
  ```

- Generate only the project (`.ipr`) file:

  ```
  mvn idea:project
  ```

- Generate only the workspace (`.iws`) file:

  ```
  mvn idea:workspace
  ```

- Generate only module (`.iml`) files:

  ```
  mvn idea:module
  ```

- Delete all generated project files:

  ```
  mvn idea:clean
  ```

mvn install

Install third-party Maven dependencies and build the project.

More information: https://manned.org/mvn.

- Compile, test, package, and install the project into the local repository:

```
mvn install
```

- Skip tests during installation:

```
mvn install {{[-D|--define]}} skipTests
```

- Force update of dependencies before installing:

```
mvn install {{[-U|--update-snapshots]}}
```

- Skip test compilation and execution:

```
mvn install {{[-D|--define]}} maven.test.skip=true
```

mvn package

Package the compiled code of a Maven project into its distributable format (such as a JAR or WAR).

More information: https://manned.org/mvn.

- Package a project:

```
mvn package
```

- Package a project while skipping test execution:

```
mvn package {{[-D|--define]}} skipTests
```

- Package a project and force Maven to update all dependencies:

```
mvn package {{[-U|--update-snapshots]}}
```

mvn site

Generate a project website based on the information in the **`pom.xml`** file.

More information: https://maven.apache.org/plugins/maven-site-plugin/usage.html.

- Generate a project site using the site plugin:

```
mvn site
```

- Generate a site for a specific Maven project (multi-module build):

```
mvn site {{[-pl|--projects]}} {{module_name}}
```

- Clean previous site output before generating a new one:

```
mvn clean site
```

- Skip tests while generating the site:

```
mvn site {{[-D|--define]}} skipTests
```

- Generate and deploy the site to the remote server:

```
mvn site-deploy
```

mvn

Apache Maven: build and manage Java-based projects.

More information: https://manned.org/mvn.

- Compile a project:

```
mvn compile
```

- Compile and package the compiled code in its distributable format, such as a `jar`:

```
mvn package
```

- Compile and package, skipping unit tests:

```
mvn package {{[-D|--define]}} skipTests
```

- Install the built package in local maven repository. (This will invoke the compile and package commands too):

```
mvn install
```

- Delete build artifacts from the target directory:

```
mvn clean
```

- Do a clean and then invoke the package phase:

```
mvn clean package
```

- Clean and then package the code with a given build profile:

```
mvn clean {{[-P|--activate-profiles]}} {{profile}} package
```

- Run a class with a main method:

```
mvn exec:java {{[-D|--define]}}
exec.mainClass="{{com.example.Main}}" {{[-D|--define]}}
exec.args="{{argument1 argument2 ...}}"
```

mycli

A CLI for MySQL, MariaDB, and Percona that can do auto-completion and syntax highlighting.

More information: https://manned.org/mycli.

- Connect to a local database on port 3306, using the current user's username:

```
mycli {{database_name}}
```

- Connect to a database (user will be prompted for a password):

```
mycli {{[-u|--user]}} {{username}} {{database_name}}
```

- Connect to a database on another host:

```
mycli {{[-h|--host]}} {{database_host}} {{[-P|--port]}} {{port}} {{[-u|--user]}} {{username}} {{database_name}}
```

mypy

Type check Python code.

More information: https://mypy.readthedocs.io/en/stable/running_mypy.html.

- Type check a specific file:

```
mypy {{path/to/file.py}}
```

- Type check a specific module:

```
mypy {{[-m|--module]}} {{module_name}}
```

- Type check a specific package:

```
mypy {{[-p|--package]}} {{package_name}}
```

- Type check a string of code:

```
mypy {{[-c|--command]}} "{{code}}"
```

- Ignore missing imports:

```
mypy --ignore-missing-imports {{path/to/file_or_directory}}
```

- Show detailed error messages:

```
mypy {{[--tb|--show-traceback]}} {{path/to/
file_or_directory}}
```

- Specify a custom configuration file:

```
mypy --config-file {{path/to/config_file}}
```

- Display help:

```
mypy {{[-h|--help]}}
```

mysql

The MySQL tool.

More information: https://manned.org/mysql.

- Connect to a database:

```
mysql {{database_name}}
```

- Connect to a database, user will be prompted for a password:

```
mysql {{[-u|--user]}} {{user}} {{[-p|--password]}} {{database_name}}
```

- Connect to a database on another host:

```
mysql {{[-h|--host]}} {{database_host}} {{database_name}}
```

- Connect to a database through a Unix socket:

```
mysql {{[-S|--socket]}} {{path/to/socket.sock}}
```

- Execute SQL statements in a script file (batch file):

```
mysql {{[-e|--execute]}} "source {{filename.sql}}" {{database_name}}
```

- Restore a database from a backup created with `mysqldump` (user will be prompted for a password):

```
mysql < {{path/to/backup.sql}} {{[-u|--user]}} {{user}} {{[-p|--password]}} {{database_name}}
```

- Restore all databases from a backup (user will be prompted for a password):

```
mysql < {{path/to/backup.sql}} {{[-u|--user]}} {{user}} {{[-p|--password]}}
```

mysql_secure_installation

Set up MySQL to have better security.

More information: https://dev.mysql.com/doc/refman/en/mysql-secure-installation.html.

- Start an interactive setup:

```
mysql_secure_installation
```

- Use specific host and port:

```
mysql_secure_installation --host {{host}} --port {{port}}
```

- Display help:

```
mysql_secure_installation --help
```

mysqlbinlog

Utility for processing MySQL binary log files.

More information: https://dev.mysql.com/doc/refman/en/mysqlbinlog.html.

- Show events from a specific binary log file:

```
mysqlbinlog {{path/to/binlog}}
```

- Show entries from a binary log for a specific database:

```
mysqlbinlog --database {{database_name}} {{path/to/binlog}}
```

- Show events from a binary log between specific dates:

```
mysqlbinlog --start-datetime='{{2022-01-01 01:00:00}}' --
stop-datetime='{{2022-02-01 01:00:00}}' {{path/to/binlog}}
```

- Show events from a binary log between specific positions:

```
mysqlbinlog --start-position={{100}} --stop-position={{200}}
{{path/to/binlog}}
```

- Show binary log from a MySQL server on the given host:

```
mysqlbinlog --host={{hostname}} {{path/to/binlog}}
```

mysqlcheck

Check and repair MySQL tables.

More information: https://dev.mysql.com/doc/refman/en/mysqlcheck.html.

- Check a table:

```
mysqlcheck --check {{table}}
```

- Check a table and provide credentials to access it:

```
mysqlcheck --check {{table}} --user {{username}} --password {{password}}
```

- Repair a table:

```
mysqlcheck --repair {{table}}
```

- Optimize a table:

```
mysqlcheck --optimize {{table}}
```

mysqld

Start the MySQL database server.

More information: https://dev.mysql.com/doc/refman/en/mysqld.html.

- Start the MySQL database server:

```
mysqld
```

- Start the server, printing error messages to the console:

```
mysqld --console
```

- Start the server, saving logging output to a custom log file:

```
mysqld --log={{path/to/file.log}}
```

- Print the default arguments and their values and exit:

```
mysqld --print-defaults
```

- Start the server, reading arguments and values from a file:

```
mysqld --defaults-file={{path/to/file}}
```

- Start the server and listen on a custom port:

```
mysqld --port={{port}}
```

- Display help:

```
mysqld --verbose --help
```

mysqldump

Backups MySQL databases.

See also: **`mysql`** for restoring databases.

More information: https://dev.mysql.com/doc/refman/en/mysqldump.html.

- Create a backup (user will be prompted for a password):

```
mysqldump --user {{user}} --password {{database_name}} --
result-file={{path/to/file.sql}}
```

- Backup a specific table redirecting the output to a file (user will be prompted for a password):

```
mysqldump --user {{user}} --password {{database_name}}
{{table_name}} > {{path/to/file.sql}}
```

- Backup all databases redirecting the output to a file (user will be prompted for a password):

```
mysqldump --user {{user}} --password --all-databases >
{{path/to/file.sql}}
```

- Backup all databases from a remote host, redirecting the output to a file (user will be prompted for a password):

```
mysqldump --host={{ip_or_hostname}} --user {{user}} --
password --all-databases > {{path/to/file.sql}}
```

mysqlsh

Advanced client for MySQL, supporting SQL, JavaScript, and Python.

It offers features for managing InnoDB clusters and document store collections.

More information: https://manned.org/mysqlsh.

- Start MySQL Shell in interactive mode:

```
mysqlsh
```

- Connect to a MySQL server:

```
mysqlsh --user {{username}} --host {{hostname}} --port {{port}}
```

- Execute an SQL statement on the server and exit:

```
mysqlsh --user {{username}} --execute '{{sql_statement}}'
```

- Start MySQL Shell in JavaScript mode:

```
mysqlsh --js
```

- Start MySQL Shell in Python mode:

```
mysqlsh --py
```

- Import JSON documents into a MySQL collection:

```
mysqlsh --import {{path/to/file.json}} --schema {{schema_name}} --collection {{collection_name}}
```

- Enable verbose output:

```
mysqlsh --verbose
```

mytop

Display MySQL server performance info like **`top`**.

More information: https://jeremy.zawodny.com/mysql/mytop/mytop.html.

- Start `mytop`:

```
mytop
```

- Connect with a specified username and password:

```
mytop {{[-u|-user]}} {{user}} {{[-p|-password]}} {{password}}
```

- Connect with a specified username (the user will be prompted for a password):

```
mytop {{[-u|-user]}} {{user}} -prompt
```

- Do not show any idle (sleeping) threads:

```
mytop {{[-u|-user]}} {{user}} {{[-p|-password]}} {{password}} --noidle
```

n

Manage multiple node versions.

More information: https://github.com/tj/n.

- Install a given version of node. If the version is already installed, it will be activated:

```
n {{version}}
```

- Display installed versions and interactively activate one of them:

```
n
```

- Remove a version:

```
n rm {{version}}
```

- Execute a file with a given version:

```
n use {{version}} {{file.js}}
```

- Output binary path for a version:

```
n bin {{version}}
```

naabu

A fast port scanner written in Go with a focus on reliability and simplicity.

Note: Some features are only activated when **naabu** is run with root privileges such as SYN scan.

More information: https://docs.projectdiscovery.io/tools/naabu/running.

- Run a SYN scan against default (top 100) ports of remote host:

```
sudo naabu -host {{host}}
```

- Display available network interfaces and public IP address of the local host:

```
naabu {{[-il|-interface-list]}}
```

- Scan all ports of the remote host (CONNECT scan without `sudo`):

```
naabu {{[-p|-port]}} - -host {{host}}
```

- Scan the top 1000 ports of the remote host:

```
naabu {{[-tp|-top-ports]}} 1000 -host {{host}}
```

- Scan TCP ports 80, 443 and UDP port 53 of the remote host:

```
naabu {{[-p|-port]}} 80,443,u:53 -host {{host}}
```

- Show CDN type the remote host is using, if any:

```
naabu {{[-p|-port]}} 80,443 -cdn -host {{host}}
```

- Run `nmap` from `naabu` for additional functionalities (`nmap` must be installed):

```
sudo naabu {{[-v|-verbose]}} -host {{host}} -nmap-cli 'nmap
{{-v -T5 -sC}}'
```

nagios

Legacy host/service/networking monitoring program.

Largely deprecated by **`nagios4`**.

See also: **`nagios2`**, **`nagios3`**, **`nagios4`**.

More information: https://manned.org/nagios.

- Start `nagios`:

```
nagios /etc/nagios/nagios.cfg
```

- Start `nagios` in daemon mode:

```
nagios -d
```

- Start `nagios`, print service check scheduling information to `stdout`, then shutdown:

```
nagios -s
```

- Verify configuration file:

```
nagios -v
```

nagios2

Legacy host/service/networking monitoring program.

Largely deprecated by **`nagios4`**.

See also: **`nagios`**, **`nagios3`**, **`nagios4`**.

More information: https://manned.org/nagios.

- Start `nagios2`:

  ```
  nagios2 /etc/nagios2/nagios.cfg
  ```

- Start `nagios2` in daemon mode:

  ```
  nagios2 -d
  ```

- Start `nagios2`, print service check scheduling information to `stdout`, then shutdown:

  ```
  nagios2 -s
  ```

- Verify configuration file:

  ```
  nagios2 -v
  ```

nagios3

Legacy host/service/networking monitoring program.

Largely deprecated by **`nagios4`**.

See also: **`nagios`**, **`nagios2`**, **`nagios4`**.

More information: https://manned.org/nagios.

- Start `nagios3`:

```
nagios3 /etc/nagios3/nagios.cfg
```

- Start `nagios3` in daemon mode:

```
nagios3 -d
```

- Start `nagios3`, print service check scheduling information to `stdout`, then shutdown:

```
nagios3 -s
```

- Verify configuration file:

```
nagios3 -v
```

nagios4

Legacy host/service/networking monitoring program.

See also: **`nagios`**, **`nagios2`**, **`nagios3`**.

More information: https://manned.org/nagios.

- Start `nagios4`:

```
nagios4 /etc/nagios4/nagios.cfg
```

- Start `nagios4` in daemon mode:

```
nagios4 -d
```

- Start `nagios4`, print service check scheduling information to `stdout`, then shutdown:

```
nagios4 -s
```

- Verify configuration file:

```
nagios4 -v
```

nano

Text editor. An enhanced **pico** clone.

See also: **pico**, **rnano**.

More information: https://nano-editor.org/dist/latest/nano.html.

- Open specific files, moving to the next file after closing the previous one:

```
nano {{path/to/file1 path/to/file2 ...}}
```

- Start the editor without using configuration files:

```
nano {{[-I|--ignorercfiles]}}
```

- Open a file and position the cursor at a specific line and column:

```
nano +{{line}},{{column}} {{path/to/file}}
```

- Open a file and enable soft wrapping:

```
nano {{[-S|--softwrap]}} {{path/to/file}}
```

- Open a file and indent new lines to the previous line's indentation:

```
nano {{[-i|--autoindent]}} {{path/to/file}}
```

- Open a file and create a backup file (`path/to/file~`) on save:

```
nano {{[-B|--backup]}} {{path/to/file}}
```

- Open a file in restricted mode (i.e. don't read/write to files not specified on the command-line):

```
nano {{[-R|--restricted]}} {{path/to/file}}
```

- Exit nano:

```
<Ctrl x>
```

nasm

The Netwide Assembler, a portable 80x86 assembler.

More information: https://www.nasm.us/xdoc/2.16.03/html/nasmdoc2.html.

- Assemble `source.asm` into a binary file `source`, in the (default) raw binary format:

```
nasm {{source.asm}}
```

- Assemble `source.asm` into a binary file `output_file`, in the specified format:

```
nasm -f {{format}} {{source.asm}} -o {{output_file}}
```

- List valid output formats (along with basic nasm help):

```
nasm -hf
```

- Assemble and generate an assembly listing file:

```
nasm -l {{list_file}} {{source.asm}}
```

- Add a directory (must be written with trailing slash) to the include file search path before assembling:

```
nasm -i {{path/to/include_directory}}/ {{source.asm}}
```

nativefier

Create a desktop app for any web site with minimal configuration.

More information: https://github.com/jiahaog/nativefier.

- Make a desktop app for a website:

```
nativefier {{url}}
```

- Create a desktop app with a custom name:

```
nativefier --name {{name}} {{url}}
```

- Use a custom icon, should be a PNG:

```
nativefier --icon {{path/to/icon.png}} {{url}}
```

nb

Note-taking, bookmarking, and archiving tool.

Supports encryption, tagging, wiki-style links, Git syncing, Pandoc conversion, and more.

More information: https://github.com/xwmx/nb#-help.

- Create a new note in your `$EDITOR`:

```
nb {{[a|add]}} "{{note_title}}"
```

- Edit a note in your `$EDITOR`:

```
nb {{[e|edit]}} {{note_id}}
```

- List all notes in the current notebook:

```
nb list
```

- Add a todo:

```
nb {{[to|todos]}} {{[a|add]}} {{title}}
```

- Import a file:

```
nb import ({{path/to/file|url}})
```

- Search for notes containing a keyword:

```
nb {{[q|search]}} "{{keyword}}"
```

- Sync notes via Git:

```
nb sync
```

- Display help:

```
nb help
```

nbtscan

Scan networks for NetBIOS name information.

More information: https://github.com/resurrecting-open-source-projects/nbtscan.

- Scan a network for NetBIOS names:

```
nbtscan {{192.168.0.1/24}}
```

- Scan a single IP address:

```
nbtscan {{192.168.0.1}}
```

- Display verbose output:

```
nbtscan -v {{192.168.0.1/24}}
```

- Display output in `/etc/hosts` format:

```
nbtscan -e {{192.168.0.1/24}}
```

- Read IP addresses/networks to scan from a file:

```
nbtscan -f {{path/to/file.txt}}
```

nc

Redirect I/O into a network stream through this versatile tool.

More information: https://manned.org/nc.

- Start a [l]istener on the specified TCP [p]ort and send a file into it:

```
nc < {{filename}} -l -p {{port}}
```

- Connect to a target listener on the specified port and receive a file from it:

```
nc {{host}} {{port}} > {{received_filename}}
```

- Scan the open TCP ports of a specified host:

```
nc -v -z -w {{timeout_in_seconds}} {{host}} {{start_port}}-{{end_port}}
```

- Start a [l]istener on the specified TCP [p]ort and provide your local shell access to the connected party (this is dangerous and can be abused):

```
nc -l -p {{port}} -e {{shell_executable}}
```

- Connect to a target listener and provide your local shell access to the remote party (this is dangerous and can be abused):

```
nc {{host}} {{port}} -e {{shell_executable}}
```

- Act as a proxy and forward data from a local TCP port to the given remote host:

```
nc -l -p {{local_port}} | nc {{host}} {{remote_port}}
```

- Send an HTTP GET request:

```
echo -e "GET / HTTP/1.1\nHost: {{host}}\n\n" | nc {{host}} 80
```

ncc

Compile a Node.js application into a single file.

Supports TypeScript, binary addons and dynamic requires.

More information: https://github.com/vercel/ncc.

- Bundle a Node.js application:

```
ncc build {{path/to/file.js}}
```

- Bundle and minify a Node.js application:

```
ncc build {{[-m|--minify]}} {{path/to/file.js}}
```

- Bundle and minify a Node.js application and generate source maps:

```
ncc build {{[-s|--source-map]}} {{path/to/file.js}}
```

- Automatically recompile on changes to source files:

```
ncc build {{[-w|--watch]}} {{path/to/file.js}}
```

- Bundle a Node.js application into a temporary directory and run it for testing:

```
ncc run {{path/to/file.js}}
```

- Clean the `ncc` cache:

```
ncc clean cache
```

ncdu

Disk usage analyzer with an ncurses interface.

More information: https://dev.yorhel.nl/ncdu/man.

- Analyze the current working directory:

```
ncdu
```

- Colorize output:

```
ncdu --color {{dark|off}}
```

- Analyze a given directory:

```
ncdu {{path/to/directory}}
```

- Save results to a file:

```
ncdu -o {{path/to/file}}
```

- Exclude files that match a pattern, argument can be given multiple times to add more patterns:

```
ncdu --exclude '{{*.txt}}'
```

ncmpcpp

A music player client for the Music Player Daemon.

See also: **`mpd`**, **`mpc`**, **`qmmp`**, **`termusic`**.

More information: https://rybczak.net/ncmpcpp.

- Connect to a music player daemon on a given host and port:

```
ncmpcpp --host {{ip}} --port {{port}}
```

- Display metadata of the current song to console:

```
ncmpcpp --current-song
```

- Use a specified configuration file:

```
ncmpcpp --config {{file}}
```

- Use a different set of key bindings from a file:

```
ncmpcpp --bindings {{file}}
```

ncspot

Cross-platform TUI Spotify client.

More information: https://github.com/hrkfdn/ncspot/blob/main/doc/users.md.

- Start `ncspot`:

  ```
  ncspot
  ```

- Enable logging to the specified file:

  ```
  ncspot {{[-d|--debug]}} {{path/to/file}}
  ```

- Display paths to the configuration and cache:

  ```
  ncspot info
  ```

- Use a different configuration file (in the directory reported by `ncspot info`; default: `config.toml`):

  ```
  ncspot {{[-c|--config]}} {{file_name}}
  ```

- Display a list of keybindings:

  ```
  <?>
  ```

ncu

Find newer versions of package dependencies and check outdated npm packages locally or globally.

`ncu` only updates dependency versions in **`package.json`**. To install the new versions, run **`npm install`** afterwards.

More information: https://github.com/raineorshine/npm-check-updates.

- List outdated dependencies in the current directory:

  `ncu`

- List outdated global `npm` packages:

  `ncu --global`

- Upgrade all dependencies in the current directory:

  `ncu --upgrade`

- Interactively upgrade dependencies in the current directory:

  `ncu --interactive`

- List outdated dependencies up to the highest minor version:

  `ncu --target {{minor}}`

- List outdated dependencies that match a keyword or `regex`:

  `ncu --filter {{keyword|/regex/}}`

- List only a specific section of outdated dependencies:

  `ncu --dep {{dev|optional|peer|prod|packageManager}}`

- Display help:

  `ncu --help`

ndc

Name daemon control service for name servers.

If a command isn't provided, NDC will prompt for one until EOF.

More information: https://manned.org/ndc.

- Set the [c]ontrol channel rendezvous point:

```
ndc -c {{channel}} {{command}}
```

- Bind the client side to a specific [l]ocalsock address:

```
ndc -l {{localsock}} {{command}}
```

- Set path to [p]idfile for UNIX signal control:

```
ndc -p {{path/to/pidfile}} {{command}}
```

- Enable [d]ebugging:

```
ndc -d {{command}}
```

- Enable [q]uiet mode:

```
ndc -q {{command}}
```

- Enable nonfatal error [s]uppression:

```
ndc -s {{command}}
```

- Enable [t]racing for protocol and system debugging:

```
ndc -t {{command}}
```

- List built-in commands:

```
ndc /help
```

neato

Render an image of a **`linear undirected`** network graph from a **`graphviz`** file.

Layouts: **`dot`**, **`neato`**, **`twopi`**, **`circo`**, **`fdp`**, **`sfdp`**, **`osage`** & **`patchwork`**.

More information: https://graphviz.org/doc/info/command.html.

- Render a PNG image with a filename based on the input filename and output format (uppercase -O):

```
neato -T {{png}} -O {{path/to/input.gv}}
```

- Render a SVG image with the specified output filename (lowercase -o):

```
neato -T {{svg}} -o {{path/to/image.svg}} {{path/to/input.gv}}
```

- Render the output in PS, PDF, SVG, Fig, PNG, GIF, JPEG, JSON, or DOT format:

```
neato -T {{format}} -O {{path/to/input.gv}}
```

- Render a GIF image using `stdin` and `stdout`:

```
echo "{{graph {this -- that} }}" | neato -T {{gif}} > {{path/to/image.gif}}
```

- Display help:

```
neato -?
```

ned

Like **grep** but with powerful replace capabilities.

Unlike **sed**, as it isn't restricted to line oriented editing.

More information: https://github.com/nevdelap/ned.

- Recursively search starting in the current directory, ignoring case:

```
ned --ignore-case --recursive '{{^[dl]og}}' {{.}}
```

- Search always showing colored output:

```
ned --colors '{{^[dl]og}}' {{.}}
```

- Search never showing colored output:

```
ned --colors=never '{{^[dl]og}}' {{.}}
```

- Search ignoring certain files:

```
ned --recursive --exclude '{{*.htm}}' '{{^[dl]og}}' {{.}}
```

- Simple replace:

```
ned '{{dog}}' --replace '{{cat}}' {{.}}
```

- Replace using numbered group references:

```
ned '{{the ([a-z]+) dog and the ([a-z]+) dog}}' --replace
'{{the $2 dog and the $1 dog}}' {{.}}
```

- Replace changing case:

```
ned '{{([a-z]+) dog}}' --case-replacements --replace
'{{\U$1\E! dog}}' --stdout {{.}}
```

- Preview results of a find and replace without updating the target files:

```
ned '{{^[sb]ad}}' --replace '{{happy}}' --stdout {{.}}
```

neo4j-admin

Manage and administer a Neo4j DBMS (Database Management System).

See also: **`cypher-shell`**, **`mysqld`**.

More information: https://neo4j.com/docs/operations-manual/current/neo4j-admin-neo4j-cli/.

- Start the DBMS:

```
neo4j-admin server start
```

- Stop the DBMS:

```
neo4j-admin server stop
```

- Set the initial password of the default `neo4j` user (prerequisite for the first start of the DBMS):

```
neo4j-admin dbms set-initial-password {{database_name}}
```

- Create an archive (dump) of an offline database to a file named `database_name.dump`:

```
neo4j-admin database dump --to-path={{path/to/directory}}
{{database_name}}
```

- Load a database from an archive named `database_name.dump`:

```
neo4j-admin database load --from-path={{path/to/directory}}
{{database_name}} --overwrite-destination=true
```

- Load a database from a specified archive file through `stdin`:

```
neo4j-admin < {{path/to/file.dump}} database load --from-
stdin {{database_name}} --overwrite-destination=true
```

- Display help:

```
neo4j-admin --help
```

neofetch

Display information about the operating system, software and hardware.

Note: **`neofetch`** is no longer maintained.

See also: **`fastfetch`**.

More information: https://github.com/dylanaraps/neofetch.

- Return the default config, and create it if it's the first time the program runs:

```
neofetch
```

- Trigger an info line from appearing in the output, where `infoname` is the function name in the configuration file, e.g. memory:

```
neofetch --{{enable|disable}} {{infoname}}
```

- Hide/Show OS architecture:

```
neofetch --os_arch {{on|off}}
```

- Enable/Disable CPU brand in output:

```
neofetch --cpu_brand {{on|off}}
```

neomutt

NeoMutt email client.

More information: https://neomutt.org/guide/reference.html.

- Open the specified mailbox:

```
neomutt -f {{path/to/mailbox}}
```

- Start writing an email and specify a subject and a `cc` recipient:

```
neomutt -s "{{subject}}" -c {{cc@example.com}}
{{recipient@example.com}}
```

- Send an email with files attached:

```
neomutt -a {{path/to/file1 path/to/file2 ...}} --
{{recipient@example.com}}
```

- Specify a file to include as the message body:

```
neomutt -i {{path/to/file}} {{recipient@example.com}}
```

- Specify a draft file containing the header and the body of the message, in RFC 5322 format:

```
neomutt -H {{path/to/file}} {{recipient@example.com}}
```

neotoppm

Convert an Atari Neochrome NEO file into a PPM image.

More information: https://netpbm.sourceforge.net/doc/neotoppm.html.

- Generate the PPM image as output for an Atari Neochrome NEO file as input:

```
neotoppm {{path/to/file.neo}}
```

- Display version:

```
neotoppm {{[-v|-version]}}
```

nest

Initialize, develop, and maintain Nest applications.

More information: https://docs.nestjs.com/cli/overview.

- Display information about installed nest version:

  ```
  nest {{[i|info]}}
  ```

- Create a new NestJS project in a directory of the same name:

  ```
  nest {{[n|new]}} {{project_name}}
  ```

- Build a specific NestJS project:

  ```
  nest build {{project_name}}
  ```

- Run a specific NestJS project:

  ```
  nest start {{project_name}}
  ```

- Import a library into the current NestJS project:

  ```
  nest add {{library_name}}
  ```

- Generate a new component:

  ```
  nest {{[g|generate]}} {{module|controller|service|...}}
  {{component_name}}
  ```

- Display help:

  ```
  nest
  ```

netcat

This command is an alias of **nc**.

- View documentation for the original command:

  ```
  tldr nc
  ```

netdiscover

Network scanner used to find live hosts on a network.

More information: https://github.com/netdiscover-scanner/netdiscover.

- Scan the IP range on the network interface for active hosts:

```
netdiscover -r {{172.16.6.0/23}} -i {{ens244}}
```

netexec

This command is an alias of **`nxc`**.

- View documentation for the original command:

```
tldr nxc
```

netlify

Deploy sites and configure continuous deployment to the Netlify platform.

More information: https://cli.netlify.com.

- Log in to the Netlify account:

  ```
  netlify login
  ```

- Deploy the contents of a directory to Netlify:

  ```
  netlify deploy
  ```

- Configure continuous deployment for a new or an existing site:

  ```
  netlify init
  ```

- Start a local dev server:

  ```
  netlify dev
  ```

netperf

Client-side command for **`netperf`**, the benchmarking application that measures network throughput. Similar to **`iperf`**.

See also: **`netserver`** for the server-side command.

More information: https://hewlettpackard.github.io/netperf/doc/netperf.html#Global-Command_002dline-Options.

- Connect to server on a specific IP address via default port (12865):

```
netperf {{address}}
```

- Specify [p]ort:

```
netperf {{address}} -p {{port}}
```

- Specify the sampling [l]ength in seconds (default is 10):

```
netperf {{address}} -l {{seconds}}
```

- Force IPv[4] or IPv[6]:

```
netperf {{address}} -{{4|6}}
```

netserver

Server-side command for **`netperf`**, the benchmarking application that measures network throughput.

See also: **`netperf`** for the client-side command.

More information: https://manned.org/netserver.1.

- Start a server on the default port (12865) and fork to background:

```
netserver
```

- Start server in foreground and do not fork:

```
netserver -D
```

- Specify [p]ort:

```
netserver -p {{port}}
```

- Force IPv[4] or IPv[6]:

```
netserver -{{4|6}}
```

netstat

Display network-related information such as open connections, open socket ports, etc.

See also: **ss**.

More information: https://manned.org/netstat.

- List all ports:

```
netstat {{[-a|--all]}}
```

- List all listening ports:

```
netstat {{[-l|--listening]}}
```

- List listening TCP ports:

```
netstat {{[-t|--tcp]}}
```

- Display PID and program names:

```
netstat {{[-p|--program]}}
```

- List information continuously:

```
netstat {{[-c|--continuous]}}
```

- List routes and do not resolve IP addresses to hostnames:

```
netstat {{[-rn|--route --numeric]}}
```

- List listening TCP and UDP ports (+ user and process if you're root):

```
netstat {{[-tulpne|--tcp --udp --listening --program --
numeric --extend]}}
```

nettacker

Automate information gathering, vulnerability scanning and eventually generate a report.

More information: https://nettacker.readthedocs.io/en/latest/Home/.

- List all available modules:

```
nettacker --show-all-modules
```

- Run a port scan on targets:

```
nettacker {{[-m|--modules]}} port_scan {{[-i|--targets]}}
{{192.168.0.1/24,owasp.org,scanme.org,...}}
```

- Run a port scan on specific ports and targets listed in a file (newline separated):

```
nettacker {{[-m|--modules]}} port_scan {{[-g|--ports]}}
{{22,80,443,...}} {{[-l|--targets-list]}} {{path/to/
targets.txt}}
```

- Run ping test before scan and then run multiple scan types on target:

```
nettacker --ping-before-scan {{[-m|--modules]}}
{{port_scan,subdomain_scan,waf_scan,...}} {{[-g|--ports]}}
{{80,443}} {{[-i|--targets]}} {{owasp.org}}
```

newman

Collection runner for Postman.

More information: https://github.com/postmanlabs/newman.

- Run a collection (from a file):

```
newman run {{path/to/collection.json}}
```

- Run a collection (from a URL):

```
newman run {{https://www.getpostman.com/collections/631643-f695cab7-6878-eb55-7943-ad88e1ccfd65-JsLv}}
```

newsboat

An RSS/Atom feed reader for text terminals.

More information: https://newsboat.org/releases/2.40/docs/newsboat.html#_first_steps.

- First import feed URLs from an OPML file:

```
newsboat {{[-i|--import-from-opml]}} {{my-feeds.xml}}
```

- Alternatively, add feeds manually:

```
echo {{http://example.com/path/to/feed}} >> "$
{HOME}/.newsboat/urls"
```

- Start Newsboat and refresh all feeds on startup:

```
newsboat {{[-r|--refresh-on-start]}}
```

- Execute one or more commands in non-interactive mode:

```
newsboat {{[-x|--execute]}} {{reload print-unread ...}}
```

- See keyboard shortcuts (the most relevant are visible in the status line):

```
<?>
```

next

React framework that uses server-side rendering for building optimized web applications.

More information: https://nextjs.org/docs/app/api-reference/cli/next.

- Start the current application in development mode:

  ```
  next dev
  ```

- Start the current application and listen on a specific port:

  ```
  next dev {{[-p|--port]}} {{port}}
  ```

- Build the current application optimized for production:

  ```
  next build
  ```

- Start the compiled application in production mode:

  ```
  next start
  ```

- Start the compiled application and listen on a specific port:

  ```
  next start {{[-p|--port]}} {{port}}
  ```

- Export the current application to static HTML pages:

  ```
  next export
  ```

- Display the Next.js telemetry status:

  ```
  next telemetry
  ```

- Display help for a subcommand:

  ```
  next {{build|dev|export|start|telemetry}} {{[-h|--help]}}
  ```

nextclade

Bioinformatics tool for virus genome alignment, clade assignment and qc checks.

More information: https://docs.nextstrain.org/projects/nextclade/en/stable/user/nextclade-cli/reference.html.

- Align sequences to user provided reference, outputting the alignment to a file:

```
nextclade run {{path/to/sequences.fa}} {{[-r|--input-ref]}} {{path/to/reference.fa}} {{[-o|--output-fasta]}} {{path/to/alignment.fa}}
```

- Create a TSV report, auto-downloading the latest dataset:

```
nextclade run {{path/to/fasta}} {{[-d|--dataset-name]}} {{dataset_name}} {{[-t|--output-tsv]}} {{path/to/report.tsv}}
```

- List all available datasets:

```
nextclade dataset list
```

- Download the latest SARS-CoV-2 dataset:

```
nextclade dataset get {{[-n|--name]}} sars-cov-2 {{[-o|--output-dir]}} {{path/to/directory}}
```

- Use a downloaded dataset, producing all outputs:

```
nextclade run {{[-D|--input-dataset]}} {{path/to/dataset_directory}} {{[-O|--output-all]}} {{path/to/output_directory}} {{path/to/sequences.fasta}}
```

- Run on multiple files:

```
nextclade run {{[-d|--dataset-name]}} {{dataset_name}} {{[-t|--output-tsv]}} {{path/to/output_tsv}} -- {{path/to/input_fasta_1 path/to/input_fasta_2 ...}}
```

- Try reverse complement if sequence does not align:

```
nextclade run --retry-reverse-complement {{[-d|--dataset-name]}} {{dataset_name}} {{[-t|--output-tsv]}} {{path/to/output_tsv}} {{path/to/input_fasta}}
```

nextflow

Run computational pipelines. Mostly used for bioinformatics workflows.

More information: https://www.nextflow.io.

- Run a pipeline, use cached results from previous runs:

```
nextflow run {{main.nf}} -resume
```

- Run a specific release of a remote workflow from GitHub:

```
nextflow run {{user/repo}} -revision {{release_tag}}
```

- Run with a given work directory for intermediate files, save execution report:

```
nextflow run {{workflow}} -work-dir {{path/to/directory}} -with-report {{report.html}}
```

- Show details of previous runs in current directory:

```
nextflow log
```

- Remove cache and intermediate files for a specific run:

```
nextflow clean -force {{run_name}}
```

- List all downloaded projects:

```
nextflow list
```

- Pull the latest version of a remote workflow from Bitbucket:

```
nextflow pull {{user/repo}} -hub bitbucket
```

- Update Nextflow:

```
nextflow self-update
```

nf-core

The nf-core framework tools, to create, check and develop best-practice guidelines for Nextflow.

More information: https://nf-co.re/docs/nf-core-tools.

- List existing pipelines on nf-core:

  ```
  nf-core list
  ```

- Create a new pipeline skeleton:

  ```
  nf-core create
  ```

- Lint the pipeline code:

  ```
  nf-core lint {{path/to/directory}}
  ```

- Bump software versions in pipeline recipe:

  ```
  nf-core bump-version {{path/to/directory}} {{new_version}}
  ```

- Launch an nf-core pipeline:

  ```
  nf-core launch {{pipeline_name}}
  ```

- Download an nf-core pipeline for offline use:

  ```
  nf-core download {{pipeline_name}}
  ```

ng

Create and manage Angular applications.

More information: https://angular.io/cli.

- Create a new Angular application inside a directory:

  `ng new {{project_name}}`

- Add a new component to one's application:

  `ng generate component {{component_name}}`

- Add a new class to one's application:

  `ng generate class {{class_name}}`

- Add a new directive to one's application:

  `ng generate directive {{directive_name}}`

- Run the application with the following command in its root directory:

  `ng serve`

- Build the application:

  `ng build`

- Run unit tests:

  `ng test`

- Display the version of your current Angular installation:

  `ng version`

nginx

Nginx web server.

More information: https://nginx.org/docs/switches.html.

- Start the server with the default configuration file:

```
nginx
```

- Start the server with a custom configuration file:

```
nginx -c {{configuration_file}}
```

- Start the server with a prefix for all relative paths in the configuration file:

```
nginx -c {{configuration_file}} -p {{path/to/prefix}}
```

- Test the configuration without affecting the running server:

```
nginx -t
```

- Reload the configuration by sending a signal with no downtime:

```
nginx -s reload
```

ngrep

Filter network traffic packets using **regex**.

More information: https://github.com/jpr5/ngrep/blob/master/EXAMPLES.md.

- Capture traffic of all interfaces:

```
ngrep -d any
```

- Capture traffic of a specific interface:

```
ngrep -d {{eth0}}
```

- Capture traffic crossing port 22 of interface eth0:

```
ngrep -d {{eth0}} port {{22}}
```

- Capture traffic from or to a host:

```
ngrep host {{www.example.com}}
```

- Filter keyword 'User-Agent:' of interface eth0:

```
ngrep -d {{eth0}} '{{User-Agent:}}'
```

ngrok

Reverse proxy that creates a secure tunnel from a public endpoint to a locally running web service.

More information: https://ngrok.com/docs/agent/cli/.

- Expose a local HTTP service on a given port:

```
ngrok http {{80}}
```

- Expose a local HTTP service on a specific host:

```
ngrok http {{example.com}}:{{80}}
```

- Expose a local HTTPS server:

```
ngrok http https://localhost
```

- Expose TCP traffic on a given port:

```
ngrok tcp {{22}}
```

- Expose TLS traffic for a specific host and port:

```
ngrok tls -hostname={{example.com}} {{443}}
```

ngs

Scripting language created specifically for Ops.

More information: https://github.com/ngs-lang/ngs.

- Execute a code snippet:

```
ngs -e "{{echo('ngs is executed')}}"
```

- Execute a script:

```
ngs {{path/to/script.ngs}}
```

- Display version:

```
ngs --version
```

nhentai

Download doujinshis from nhentai.

More information: https://github.com/RicterZ/nhentai.

- Set cookies:

```
nhentai --cookie "csrftoken={{TOKEN}}; sessionid={{ID}}"
```

- Download a specific doujin:

```
nhentai --id {{number}}
```

- Download the first page of your favorites:

```
nhentai --favorites --download --delay 1
```

- Download specific pages of your favorites:

```
nhentai --favorites --pages {{start_page}}-{{end_page}} --
download --delay 1
```

nice

Execute a program with a custom scheduling priority (niceness).

Niceness values range from -20 (the highest priority) to 19 (the lowest).

More information: https://www.gnu.org/software/coreutils/manual/html_node/nice-invocation.html.

- Launch a program with altered priority:

```
nice -{{niceness_value}} {{command}}
```

- Define the priority with an explicit option:

```
nice {{[-n|--adjustment]}} {{niceness_value}} {{command}}
```

nicotine

Graphical client for the Soulseek peer-to-peer network.

More information: https://manned.org/nicotine.

- Start Nicotine+:

```
nicotine
```

- Start Nicotine+ in headless ([n]) mode:

```
nicotine {{[-n|--headless]}}
```

- Rescan shared files:

```
nicotine {{[-r|--rescan]}}
```

- Display help:

```
nicotine {{[-h|--help]}}
```

nikto

Web server scanner which performs tests against web servers for multiple items.

More information: https://cirt.net/Nikto2.

- Perform a basic Nikto scan against a target host:

```
perl nikto.pl {{[-h|-host]}} {{192.168.0.1}}
```

- Specify the port number when performing a basic scan:

```
perl nikto.pl {{[-h|-host]}} {{192.168.0.1}} {{[-p|-port]}} {{443}}
```

- Scan ports and protocols with full URL syntax:

```
perl nikto.pl {{[-h|-host]}} {{https://192.168.0.1:443/}}
```

- Scan multiple ports in the same scanning session:

```
perl nikto.pl {{[-h|-host]}} {{192.168.0.1}} {{[-p|-port]}} {{80,88,443}}
```

- Update to the latest plugins and databases:

```
perl nikto.pl {{[-u|-update]}}
```

nim

The Nim compiler.

Processes, compiles and links Nim language source files.

More information: https://nim-lang.org/docs/nimc.html.

- Compile a source file:

```
nim {{[c|compile]}} {{path/to/file.nim}}
```

- Compile and run a source file:

```
nim {{[c|compile]}} {{[-r|--run]}} {{path/to/file.nim}}
```

- Compile a source file with release optimizations enabled:

```
nim {{[c|compile]}} {{[-d|--define]}}:release {{path/to/
file.nim}}
```

- Build a release binary optimized for low file size:

```
nim {{[c|compile]}} {{[-d|--define]}}:release --opt:size
{{path/to/file.nim}}
```

- Generate HTML documentation for a module (output will be placed in the current directory):

```
nim doc {{path/to/file.nim}}
```

- Check a file for syntax and semantics:

```
nim check {{path/to/file.nim}}
```

nimble

Package manager for the Nim programming language.

Manage Nim projects and their dependencies.

More information: https://github.com/nim-lang/nimble.

- Search for packages:

```
nimble search {{search_string}}
```

- Install a package:

```
nimble install {{package}}
```

- List installed packages:

```
nimble list {{[-i|--installed]}}
```

- Create a new Nimble package in the current directory:

```
nimble init
```

- Build a Nimble package:

```
nimble build
```

- Install a Nimble package:

```
nimble install
```

ninja

A Build system designed to be fast.

More information: https://ninja-build.org/manual.html.

- Build in the current directory:

```
ninja
```

- Build in the current directory, executing 4 jobs at a time in parallel:

```
ninja -j {{4}}
```

- Build a program in a given directory:

```
ninja -C {{path/to/directory}}
```

- Show targets (e.g. `install` and `uninstall`):

```
ninja -t targets
```

- Display help:

```
ninja -h
```

nix-build

Build a Nix expression.

See also: **`nix build.3`**.

More information: https://nixos.org/manual/nix/stable/command-ref/nix-build.html.

- Build a Nix expression:

```
nix-build '<nixpkgs>' {{[-A|--attr]}} {{firefox}}
```

- Build a sandboxed Nix expression (on non-NixOS):

```
nix-build '<nixpkgs>' {{[-A|--attr]}} {{firefox}} --option sandbox true
```

nix build

Build a Nix expression (downloading from the cache when possible).

See also: **`nix-build`** for information about traditional Nix builds from expressions, **`nix flake`** for information about flakes.

More information: https://nixos.org/manual/nix/stable/command-ref/new-cli/nix3-build.html.

- Build a package from nixpkgs, symlinking the result to `./result`:

  ```
  nix build {{nixpkgs#pkg}}
  ```

- Build a package from a flake in the current directory, showing the build logs in the process:

  ```
  nix build -L {{.#pkg}}
  ```

- Build the default package from a flake in some directory:

  ```
  nix build {{path/to/directory}}
  ```

- Build a package without making the `result` symlink, instead printing the store path to the `stdout`:

  ```
  nix build --no-link --print-out-paths
  ```

nix-build

nix-build can refer to multiple commands with the same name.

- View documentation for the traditional builder:

  ```
  tldr nix-build.2
  ```

- View documentation for nix3 builder:

  ```
  tldr nix build.3
  ```

nix-channel

Manage **`nix`** update channels.

More information: https://nixos.wiki/wiki/Nix_channels.

- List current channels:

```
nix-channel --list
```

- Add a channel:

```
nix-channel --add {{https://nixos.org/channels/nixpkgs-unstable}}
```

- Update package list of all channels:

```
nix-channel --update
```

nix classic

A classic, stable interface to a powerful package manager that makes package management reliable, reproducible, and declarative.

Some Nix commands such as **`nix-build`**, **`nix-shell`**, **`nix-env`**, and **`nix-store`** have their own pages.

See also: **`nix`**.

More information: https://nixos.org.

- Search for a package in nixpkgs via its name:

```
nix-env {{[-qaP|--query --available --attr-path]}} {{search_term_regex}}
```

- Start a shell with the specified packages available:

```
nix-shell {{[-p|--packages]}} {{pkg1 pkg2 pkg3 ...}}
```

- Install some packages permanently:

```
nix-env {{[-iA|--install --attr]}} {{nixpkgs.pkg1 nixpkgs.pkg2 ...}}
```

- Show all dependencies of a store path (package), in a tree format:

```
nix-store {{[-q|--query]}} --tree /nix/store/{{checksum-package-version.ext}}
```

- Update the channels (repositories):

```
nix-channel --update
```

- Remove unused paths from Nix store:

```
nix-collect-garbage
```

nix-collect-garbage

Delete unused and unreachable nix store paths.

Generations can be listed using **`nix-env --list-generations`**.

More information: https://nixos.org/manual/nix/stable/command-ref/nix-collect-garbage.html.

- Delete all store paths unused by current generations of each profile:

```
nix-collect-garbage {{[-d|--delete-old]}}
```

- Simulate the deletion of old store paths:

```
nix-collect-garbage {{[-d|--delete-old]}} --dry-run
```

- Delete all store paths older than 30 days:

```
nix-collect-garbage --delete-older-than 30d
```

nix develop

Run a Bash shell that provides the build environment of a derivation.

More information: https://nixos.org/manual/nix/stable/command-ref/new-cli/nix3-develop.html.

- Start a shell with all dependencies of a package from nixpkgs available:

  ```
  nix develop {{nixpkgs#pkg}}
  ```

- Start a development shell for the default package in a flake in the current directory:

  ```
  nix develop
  ```

- In that shell, configure and build the sources:

  ```
  configurePhase; buildPhase
  ```

nix edit

Open the Nix expression of a Nix package in $EDITOR.

More information: https://nixos.org/manual/nix/stable/command-ref/new-cli/nix3-edit.html.

- Open the source of the Nix expression of a package from nixpkgs in your `$EDITOR`:

```
nix edit {{nixpkgs#pkg}}
```

- Dump the source of a package to `stdout`:

```
EDITOR=cat nix edit {{nixpkgs#pkg}}
```

nix-env

Manipulate or query Nix user environments.

More information: https://nixos.org/manual/nix/stable/#sec-nix-env.

- List all installed packages:

```
nix-env {{[-q|--query]}}
```

- Query installed packages:

```
nix-env {{[-q|--query]}} {{search_term}}
```

- Query available packages:

```
nix-env {{[-qa|--query --available]}} {{search_term}}
```

- Install package:

```
nix-env {{[-iA|--install --attr]}} nixpkgs.{{pkg_name}}
```

- Install a package from a URL:

```
nix-env {{[-i|--install]}} {{pkg_name}} {{[-f|--file]}}
{{example.com}}
```

- Uninstall package:

```
nix-env {{[-e|--uninstall]}} {{pkg_name}}
```

- Upgrade one package:

```
nix-env {{[-u|--upgrade]}} {{pkg_name}}
```

- Upgrade all packages:

```
nix-env {{[-u|--upgrade]}}
```

nix flake

Manage Nix flakes.

More information: https://nixos.org/manual/nix/stable/command-ref/new-cli/nix3-flake.html.

- Create a new flake (just the `flake.nix` file) from the default template, in the current directory:

```
nix flake init
```

- Update all inputs (dependencies) of the flake in the current directory:

```
nix flake update
```

- Update a specific input (dependency) of the flake in the current directory:

```
nix flake update {{input}}
```

- Show all the outputs of a flake on github:

```
nix flake show {{github:owner/repo}}
```

- Display help:

```
nix flake --help
```

nix profile

Install, update and remove packages from Nix profiles.

More information: https://nixos.org/manual/nix/stable/command-ref/new-cli/nix3-profile.html.

- Install some packages from nixpkgs into the default profile:

```
nix profile install {{nixpkgs#pkg1 nixpkgs#pkg2 ...}}
```

- Install a package from a flake on GitHub into a custom profile:

```
nix profile install {{github:owner/repo/pkg}} --profile {{path/to/directory}}
```

- List packages currently installed in the default profile:

```
nix profile list
```

- Remove a package installed from nixpkgs from the default profile, by name:

```
nix profile remove {{legacyPackages.x86_64-linux.pkg}}
```

- Upgrade packages in the default to the latest available versions:

```
nix profile upgrade
```

- Rollback (cancel) the latest action on the default profile:

```
nix profile rollback
```

nix registry

Manage a Nix flake registry.

See also: **`nix flake`** for information about flakes.

More information: https://nixos.org/manual/nix/stable/command-ref/new-cli/nix3-registry.html.

- Pin the `nixpkgs` revision to the current version of the upstream repository:

```
nix registry pin {{nixpkgs}}
```

- Pin an entry to the latest version of the branch, or a particular reivision of a GitHub repository:

```
nix registry pin {{entry}} {{github:owner/repo/branch_or_revision}}
```

- Add a new entry that always points to the latest version of a GitHub repository, updating automatically:

```
nix registry add {{entry}} {{github:owner/repo}}
```

- Remove a registry entry:

```
nix registry remove {{entry}}
```

- See documentation about what Nix flake registries are:

```
nix registry --help
```

nix repl

Start an interactive environment for evaluating Nix expressions.

See https://nixos.org/manual/nix/stable/language/index.html for a description of the Nix expression language.

More information: https://nixos.org/manual/nix/stable/command-ref/new-cli/nix3-repl.html.

- Start an interactive environment for evaluating Nix expressions:

  ```
  nix repl
  ```

- Load all packages from a flake (e.g. `nixpkgs`) into scope:

  ```
  :lf {{nixpkgs}}
  ```

- Build a package from an expression:

  ```
  :b {{expression}}
  ```

- Start a shell with package from the expression available:

  ```
  :u {{expression}}
  ```

- Start a shell with dependencies of the package from the expression available:

  ```
  :s {{expression}}
  ```

nix run

Run an application from a Nix flake.

See also: **`nix flake`** for information about flakes.

More information: https://nixos.org/manual/nix/stable/command-ref/new-cli/nix3-run.html.

- Run the default application in the flake in the current directory:

```
nix run
```

- Run a command whose name matches the package name from nixpkgs (if you want a different command from that package, see `tldr nix shell`):

```
nix run nixpkgs#{{pkg}}
```

- Run a command with provided arguments:

```
nix run nixpkgs#{{vim}} -- {{path/to/file}}
```

- Run from a remote repository:

```
nix run {{remote_name}}:{{owner}}/{{repo}}
```

- Run from a remote repository using a specific tag, revision or branch:

```
nix run {{remote_name}}:{{owner}}/{{repo}}/{{reference}}
```

- Run from a remote repository specifying a subdirectory and a program:

```
nix run "{{remote_name}}:{{owner}}/{{repo}}?dir={{dir_name}}
#{{app}}"
```

- Run the flake of a GitHub pull request:

```
nix run github:{{owner}}/{{repo}}/pull/{{number}}/head
```

nix search

Search for packages in a Nix flake.

See also: **`nix flake`** for information about flakes.

More information: https://nixos.org/manual/nix/stable/command-ref/new-cli/nix3-search.html.

- Search `nixpkgs` for a package based on its name or description:

```
nix search {{nixpkgs}} {{search_term}}
```

- Show description of a package from nixpkgs:

```
nix search {{nixpkgs#pkg}}
```

- Show all packages available from a flake on github:

```
nix search {{github:owner/repo}}
```

nix-shell

Start an interactive shell based on a Nix expression.

See also: **`nix shell.3`**.

More information: https://nixos.org/manual/nix/stable/command-ref/nix-shell.html.

- Start with nix expression in `shell.nix` or `default.nix` in the current directory:

```
nix-shell
```

- Run shell command in non-interactive shell and exit:

```
nix-shell --run "{{command}} {{argument1 argument2 ...}}"
```

- Start with expression in `default.nix` in the current directory:

```
nix-shell {{default.nix}}
```

- Start with packages loaded from nixpkgs:

```
nix-shell {{[-p|--packages]}} {{package1 package2 ...}}
```

- Start with packages loaded from specific nixpkgs revision:

```
nix-shell {{[-p|--packages]}} {{package1 package2 ...}} {{[-I|--include]}} nixpkgs={{https://github.com/NixOS/nixpkgs/archive/nixpkgs_revision.tar.gz}}
```

- Evaluate rest of file in specific interpreter, for use in `#!-scripts` (see https://nixos.org/manual/nix/stable/#use-as-a-interpreter):

```
nix-shell -i {{interpreter}} {{[-p|--packages]}} {{package1 package2 ...}}
```

nix shell

Start a shell in which the specified packages are available.

See also: **`nix-shell`** for setting up development environments, **`nix flake`** for information about flakes.

More information: https://manned.org/nix3-shell.

- Start an interactive shell with some packages from `nixpkgs`:

  ```
  nix shell {{nixpkgs#pkg1 nixpkgs#packageSet.pkg2 ...}}
  ```

- Start a shell providing a package from an older version of `nixpkgs` (21.05):

  ```
  nix shell {{nixpkgs/nixos-21.05#pkg}}
  ```

- Start a shell with the "default package" from a flake in the current directory, printing build logs if any builds happen:

  ```
  nix shell -L
  ```

- Start a shell with a package from a flake on GitHub:

  ```
  nix shell {{github:owner/repo#pkg}}
  ```

- Run a command in a shell with a package:

  ```
  nix shell {{nixpkgs#pkg}} -c {{some-cmd --someflag 'Some other arguments'}}
  ```

nix-shell

nix-shell can refer to multiple commands with the same name.

- View documentation for the traditional shell:

  `tldr nix-shell.2`

- View documentation for nix3 shell:

  `tldr nix shell.3`

nix-store

Manipulate or query the Nix store.

See also: **`nix store.3`**.

More information: https://nixos.org/manual/nix/stable/command-ref/nix-store.html.

- Collect garbage, such as removing unused paths:

```
nix-store --gc
```

- Hard-link identical files together to reduce space usage:

```
nix-store --optimise
```

- Delete a specific store path (must be unused):

```
nix-store --delete /nix/store/{{checksum-package-version.ext}}
```

- Show all dependencies of a store path (package), in a tree format:

```
nix-store {{[-q|--query]}} --tree /nix/store/{{checksum-package-version.ext}}
```

- Calculate the total size of a certain store path with all the dependencies:

```
du {{[-cLsh|--total --dereference --summarize --human-readable]}} $(nix-store {{[-q|--query]}} --references /nix/store/{{checksum-package-version.ext}})
```

- Show all dependents of a particular store path:

```
nix-store {{[-q|--query]}} --referrers /nix/store/{{checksum-package-version.ext}}
```

nix store

Manipulate the Nix store.

See also: **`nix-store`**.

More information: https://nixos.org/manual/nix/stable/command-ref/new-cli/nix3-store.html.

- Collect garbage, i.e. remove unused paths to reduce space usage:

```
nix store gc
```

- Hard-link identical files together to reduce space usage:

```
nix store optimise
```

- Delete a specific store path (most be unused):

```
nix store delete /nix/store/{{checksum-package-version.ext}}
```

- List a contents of the store path, on a remote store:

```
nix store --store {{https://cache.nixos.org}} ls /nix/store/{{checksum-package-version.ext}}
```

- Show the differences in versions between two store paths, with their respective dependencies:

```
nix store diff-closures /nix/store/{{checksum-package-version.ext}} /nix/store/{{checksum-package-version.ext}}
```

nix-store

nix-store can refer to multiple commands with the same name.

- View documentation for the traditional store:

  `tldr nix-store.2`

- View documentation for nix3 store:

  `tldr nix store.3`

nix why-depends

Show why a package depends on another package.

More information: https://nixos.org/manual/nix/stable/command-ref/new-cli/nix3-why-depends.html.

- Show why the currently running NixOS system requires a certain store path:

```
nix why-depends {{/run/current-system}} /nix/store/
{{checksum-package-version.ext}}
```

- Show why a package from nixpkgs requires another package as a build-time dependency:

```
nix why-depends --derivation {{nixpkgs#dependent}}
{{nixpkgs#dependency}}
```

nix

A powerful package manager that makes package management reliable, reproducible, and declarative.

`nix` is experimental and requires enabling experimental features.

See also: **`nix classic`** for a classic, stable interface.

Some subcommands such as **`build`**, **`develop`**, **`flake`**, **`registry`**, **`profile`**, **`search`**, **`repl`**, **`store`**, **`edit`**, **`why-depends`**, etc. have their own usage documentation.

More information: https://nix.dev/manual/nix/stable/command-ref/new-cli/nix.

- Enable the `nix` command:

```
mkdir {{[-p|--parents]}} ~/.config/nix; echo 'experimental-features = nix-command flakes' > ~/.config/nix/nix.conf
```

- Search for a package in nixpkgs via its name or description:

```
nix search nixpkgs {{search_term}}
```

- Start a shell with the specified packages from nixpkgs available:

```
nix shell {{nixpkgs#pkg1 nixpkgs#pkg2 nixpkgs#pkg3 ...}}
```

- Install some packages from nixpkgs permanently:

```
nix profile install {{nixpkgs#pkg1 nixpkgs#pkg2 nixpkgs#pkg3 ...}}
```

- Remove unused paths from Nix store to free up space:

```
nix store gc
```

- Start an interactive environment for evaluating Nix expressions:

```
nix repl
```

- Display help for a specific subcommand:

```
nix help {{subcommand}}
```

nixpkgs-review

Review pull requests in the NixOS packages repository (nixpkgs).

After a successful build, a **`nix-shell`** with all built packages is started.

More information: https://github.com/Mic92/nixpkgs-review#usage.

- Build changed packages in the specified pull request:

  ```
  nixpkgs-review pr {{pr_number|pr_url}}
  ```

- Build changed packages and post a comment with a report (requires setting up a token in `hub`, `gh`, or the `GITHUB_TOKEN` environment variable):

  ```
  nixpkgs-review pr --post-result {{pr_number|pr_url}}
  ```

- Build changed packages and print a report:

  ```
  nixpkgs-review pr --print-result {{pr_number|pr_url}}
  ```

- Build changed packages in a local commit:

  ```
  nixpkgs-review rev {{HEAD}}
  ```

- Build changed packages that haven't been committed yet:

  ```
  nixpkgs-review wip
  ```

- Build changed packages that have been staged:

  ```
  nixpkgs-review wip --staged
  ```

nkf

Network kanji filter: convert kanji code from one encoding to another.

More information: https://manned.org/nkf.

- Convert to UTF-8 encoding:

```
nkf -w {{path/to/file.txt}}
```

- Convert to SHIFT_JIS encoding:

```
nkf -s {{path/to/file.txt}}
```

- Convert to UTF-8 encoding and overwrite the file:

```
nkf -w --overwrite {{path/to/file.txt}}
```

- Use LF as the new line code and overwrite (UNIX type):

```
nkf -d --overwrite {{path/to/file.txt}}
```

- Use CRLF as the new line code and overwrite (windows type):

```
nkf -c --overwrite {{path/to/file.txt}}
```

- Decrypt mime file and overwrite:

```
nkf -m --overwrite {{path/to/file.txt}}
```

nl

Number lines from a file or from **`stdin`**.

More information: https://manned.org/nl.1p.

- Number non-blank lines in a file:

```
nl {{path/to/file}}
```

- Read from `stdin`:

```
{{command}} | nl -
```

- Number [a]ll [b]ody lines including blank lines or do [n]ot number [b]ody lines:

```
nl -b {{a|n}} {{path/to/file}}
```

- Number only the [b]ody lines that match a basic `regex` (BRE) [p]attern:

```
nl -b p'FooBar[0-9]' {{path/to/file}}
```

- Use a specific [i]ncrement for line numbering:

```
nl -i {{increment}} {{path/to/file}}
```

- Specify the line numbering format to [r]ight or [l]eft justified, keeping leading [z]eros or [n]ot:

```
nl -n {{rz|ln|rn}}
```

- Specify the line numbering's [w]idth (6 by default):

```
nl -w {{col_width}} {{path/to/file}}
```

- Use a specific string to [s]eparate the line numbers from the lines (TAB by default):

```
nl -s {{separator}} {{path/to/file}}
```

nload

Visualize network usage in the terminal.

More information: https://github.com/rolandriegel/nload.

- View all network traffic (use the `<ArrowKeys>` to switch interfaces):

```
nload
```

- View network traffic on specific interfaces (use the `<ArrowKeys>` to switch interfaces):

```
nload devices {{interface1 interface2 ...}}
```

nm-classic

This command is an alias of **nm**.

- View documentation for the original command:

```
tldr nm
```

nm

List symbol names in object files.

More information: https://manned.org/nm.

- List global (extern) functions in a file (prefixed with T):

```
nm {{[-g|--extern-only]}} {{path/to/file.o}}
```

- List only undefined symbols in a file:

```
nm {{[-u|--undefined-only]}} {{path/to/file.o}}
```

- List all symbols, even debugging symbols:

```
nm {{[-a|--debug-syms]}} {{path/to/file.o}}
```

- Demangle C++ symbols (make them readable):

```
nm {{[-C|--demangle]}} {{path/to/file.o}}
```

nmap

Network exploration tool and security/port scanner.

Some features (e.g. SYN scan) activate only when **`nmap`** is run with root privileges.

More information: https://nmap.org/book/man.html.

- Scan the top 1000 ports of a remote host with various [v]erbosity levels:

```
nmap -v{{1|2|3}} {{ip_or_hostname}}
```

- Run a ping sweep over an entire [s]ub[n]et or individual hosts very aggressively:

```
nmap -T5 -sn {{192.168.0.0/24|
ip_or_hostname1,ip_or_hostname2,...}}
```

- Enable OS detection, version detection, script scanning, and traceroute of hosts from a file:

```
sudo nmap -A -iL {{path/to/file.txt}}
```

- Scan a specific list of [p]orts (use `-p-` for all ports from 1 to 65535):

```
nmap -p {{port1,port2,...}} {{ip_or_host1,ip_or_host2,...}}
```

- Perform service and version detection of the top 1000 ports using default NSE scripts, writing results (`-oA`) to output files:

```
nmap -sC -sV -oA {{top-1000-ports}}
{{ip_or_host1,ip_or_host2,...}}
```

- Scan target(s) carefully using `default and safe` NSE scripts:

```
nmap --script "default and
safe" {{ip_or_host1,ip_or_host2,...}}
```

- Scan for web servers running on standard [p]orts 80 and 443 using all available `http-*` NSE scripts:

```
nmap --script "http-*" {{ip_or_host1,ip_or_host2,...}} -p
80,443
```

- Attempt evading IDS/IPS detection by using an extremely slow scan (`-T0`), [D]ecoy source addresses, [f]ragmented packets, random data and other methods:

```
sudo nmap -T0 -D {{decoy_ip1,decoy_ip2,...}} --source-port
{{53}} -f --data-length {{16}} -Pn {{ip_or_host}}
```

nmblookup

Discover SMB shares.

More information: https://www.samba.org/samba/docs/current/man-html/nmblookup.1.html.

- Find hosts in the local network with SMB shares:

```
nmblookup {{[-S|--status]}} '*'
```

- Find hosts in the local network with SMB shares run by SAMBA:

```
nmblookup {{[-S|--status]}} __SAMBA__
```

nms

Tool that recreates the famous data decryption effect seen in the 1992 movie Sneakers from **`stdin`**.

More information: https://github.com/bartobri/no-more-secrets.

- Decrypt text after a keystroke:

```
echo "{{Hello, World!}}" | nms
```

- Decrypt output immediately, without waiting for a keystroke:

```
{{ls -la}} | nms -a
```

- Decrypt the content of a file, with a custom output color:

```
cat {{path/to/file}} | nms -a -f {{blue|white|yellow|black|magenta|green|red}}
```

- Clear the screen before decrypting:

```
{{command}} | nms -a -c
```

nnn

Interactive terminal file manager and disk usage analyzer.

More information: https://github.com/jarun/nnn/wiki/Usage#program-options.

- Open the current directory (or specify one as the first argument):

```
nnn
```

- Start in [d]etailed mode:

```
nnn -d
```

- Show [H]idden files:

```
nnn -H
```

- Open an existing [b]ookmark (defined in the `NNN_BMS` environment variable):

```
nnn -b {{bookmark_name}}
```

- Sort files on [a]pparent disk usage / [d]isk usage / [e]xtension / [r]everse / [s]ize / [t]ime / [v]ersion:

```
nnn -T {{a|d|e|r|s|t|v}}
```

- Open a file you have selected. Select the file then press `<o>`, and type a program to open the file in:

```
nnn -o
```

node

Server-side JavaScript platform (Node.js).

More information: https://nodejs.org.

- Run a JavaScript file:

```
node {{path/to/file}}
```

- Start a REPL (interactive shell):

```
node
```

- Execute the specified file restarting the process when an imported file is changed (requires Node.js version 18.11+):

```
node --watch {{path/to/file}}
```

- Evaluate JavaScript code by passing it as an argument:

```
node {{[-e|--eval]}} "{{code}}"
```

- Evaluate and print the result, useful to print node's dependencies versions:

```
node {{[-p|--print]}} "process.versions"
```

- Activate inspector, pausing execution until a debugger is connected once source code is fully parsed:

```
node --no-lazy --inspect-brk {{path/to/file}}
```

nodemon

Watch files and automatically restart a node application when changes are detected.

More information: https://github.com/remy/nodemon/tree/main/doc/cli.

- Execute the specified file and watch a specific file for changes:

```
nodemon {{path/to/file.js}}
```

- Manually restart nodemon (note nodemon must already be active for this to work):

```
rs
```

- Ignore specific files:

```
nodemon {{[-i|--ignore]}} {{path/to/file_or_directory}}
```

- Pass arguments to the node application:

```
nodemon {{path/to/file.js}} {{arguments}}
```

- Pass arguments to node itself if they're not nodemon arguments already (e.g. `--inspect`):

```
nodemon {{arguments}} {{path/to/file.js}}
```

- Run an arbitrary non-node script:

```
nodemon {{[-x|--exec]}} "{{command_to_run_script}} {{options}}" {{path/to/script}}
```

- Run a Python script:

```
nodemon {{[-x|--exec]}} "python {{options}}" {{path/to/file.py}}
```

nodenv

Manage Node.js versions.

More information: https://github.com/nodenv/nodenv.

- Install a specific version of Node.js:

```
nodenv install {{version}}
```

- Display a list of available versions:

```
nodenv install --list
```

- Use a specific version of Node.js across the whole system:

```
nodenv global {{version}}
```

- Use a specific version of Node.js with a directory:

```
nodenv local {{version}}
```

- Display the Node.js version for the current directory:

```
nodenv version
```

- Display the location of a Node.js installed command (e.g. `npm`):

```
nodenv which {{command}}
```

nohup

Allows for a process to live when the terminal gets killed.

More information: https://www.gnu.org/software/coreutils/manual/html_node/nohup-invocation.html.

- Run a process that can live beyond the terminal:

```
nohup {{command}} {{argument1 argument2 ...}}
```

- Launch `nohup` in background mode:

```
nohup {{command}} {{argument1 argument2 ...}} &
```

- Run a shell script that can live beyond the terminal:

```
nohup {{path/to/script.sh}} &
```

- Run a process and write the output to a specific file:

```
nohup {{command}} {{argument1 argument2 ...}} > {{path/to/output_file}} &
```

nokogiri

An HTML, XML, SAX and Reader parser.

More information: https://manned.org/nokogiri.

- Parse the contents of a URL or file:

```
nokogiri {{url|path/to/file}}
```

- Parse as a specific type:

```
nokogiri {{url|path/to/file}} --type {{xml|html}}
```

- Load a specific initialization file before parsing:

```
nokogiri {{url|path/to/file}} -C {{path/to/config_file}}
```

- Parse using a specific encoding:

```
nokogiri {{url|path/to/file}} {{[-E|--encoding]}}
{{encoding}}
```

- Validate using a RELAX NG file:

```
nokogiri {{url|path/to/file}} --rng {{url|path/to/file}}
```

nomad

Distributed, highly available, datacenter-aware scheduler.

More information: https://www.nomadproject.io/docs/commands/.

- Show the status of nodes in the cluster:

```
nomad node status
```

- Validate a job file:

```
nomad job validate {{path/to/file.nomad}}
```

- Plan a job for execution on the cluster:

```
nomad job plan {{path/to/file.nomad}}
```

- Run a job on the cluster:

```
nomad job run {{path/to/file.nomad}}
```

- Show the status of jobs currently running on the cluster:

```
nomad job status
```

- Show the detailed status information about a specific job:

```
nomad job status {{job_name}}
```

- Follow the logs of a specific allocation:

```
nomad alloc logs {{alloc_id}}
```

- Show the status of storage volumes:

```
nomad volume status
```

nop

Check validity and pretty-print graphs in canonical format.

Graphviz filters: **`acyclic`**, **`bcomps`**, **`comps`**, **`edgepaint`**, **`gvcolor`**, **`gvpack`**, **`mingle`**, **`nop`**, **`sccmap`**, **`tred`**, & **`unflatten`**.

More information: https://www.graphviz.org/pdf/nop.1.pdf.

- Pretty-print one or more graphs in canonical format:

```
nop {{path/to/input1.gv path/to/input2.gv ...}} > {{path/to/output.gv}}
```

- Check one or more graphs for validity, producing no output graph:

```
nop -p {{path/to/input1.gv path/to/input2.gv ...}}
```

- Display help:

```
nop -?
```

noti

Monitor a process and trigger a banner notification.

More information: https://github.com/variadico/noti.

- Display a notification when tar finishes compressing files:

```
noti {{tar -cjf example.tar.bz2 example/}}
```

- Display a notification even when you put it after the command to watch:

```
{{command_to_watch}}; noti
```

- Monitor a process by PID and trigger a notification when the PID disappears:

```
noti {{[-w|--pwatch]}} {{process_id}}
```

notify-send

Use the current desktop environment's notification system to create a notification.

More information: https://manned.org/notify-send.

- Show a notification with the title "Test" and the content "This is a test":

```
notify-send "{{Test}}" "{{This is a test}}"
```

- Show a notification with a custom icon:

```
notify-send {{[-i|--icon]}} {{icon.png}} "{{Test}}" "{{This is a test}}"
```

- Show a notification for 5 seconds:

```
notify-send {{[-t|--expire-time]}} 5000 "{{Test}}" "{{This is a test}}"
```

- Show a notification with the specified urgency level (default: normal):

```
notify-send {{[-u|--urgency]}} {{low|normal|critical}} "{{Test}}" "{{This is a test}}"
```

- Show a notification with an app's icon and name:

```
notify-send "{{Test}}" {{[-i|--icon]}} {{google-chrome}} {{[-a|--app-name]}} "{{Google Chrome}}"
```

notmuch

Index, search, read, and tag large collections of email messages.

More information: https://notmuchmail.org/manpages/.

- Configure for first use:

```
notmuch setup
```

- Add a tag for all messages matching a search term:

```
notmuch tag +{{custom_tag}} "{{search_term}}"
```

- Remove a tag for all messages matching a search term:

```
notmuch tag -{{custom_tag}} "{{search_term}}"
```

- Count messages matching the given search term:

```
notmuch count --output={{messages|threads}} "{{search_term}}"
```

- Search for messages matching the given search term:

```
notmuch search --format={{json|text}} --output={{summary|
threads|messages|files|tags}} "{{search_term}}"
```

- Limit the number of search results to X:

```
notmuch search --format={{json|text}} --output={{summary|
threads|messages|files|tags}} --limit={{X}} "{{search_term}}"
```

- Create a reply template for a set of messages:

```
notmuch reply --format={{default|headers-only}} --reply-
to={{sender|all}} "{{search_term}}"
```

now

Cloud platform for serverless deployment.

This command is deprecated. See **`vercel`**, the updated version of this tool.

More information: https://zeit.co/now.

- Deploy the current directory:

  ```
  now
  ```

- Display a list of deployments:

  ```
  now list
  ```

- Display information related to a deployment:

  ```
  now inspect {{deployment_url}}
  ```

- Remove a deployment:

  ```
  now remove {{deployment_id}}
  ```

- Log in into an account or create a new one:

  ```
  now login
  ```

- Initialize an example project (a new directory will be created):

  ```
  now init
  ```

nping

Network packet generation tool/ping utility.

More information: https://nmap.org/nping/.

- Ping a specified host using ICMP if the user is allowed to, otherwise using TCP:

```
nping {{example.com}}
```

- Ping a specified host using ICMP assuming that the user is allowed to do so:

```
nping --icmp --privileged {{example.com}}
```

- Ping a specified host using UDP:

```
nping --udp {{example.com}}
```

- Ping a specified host on a given port using TCP:

```
nping --tcp --dest-port {{443}} {{example.com}}
```

- Ping a certain number of times:

```
nping --count {{10}} {{example.com}}
```

- Wait a certain amount of time between each ping:

```
nping --delay {{5s}} {{example.com}}
```

- Send the request over a specified interface:

```
nping --interface {{eth0}} {{example.com}}
```

- Ping an IP range:

```
nping {{10.0.0.1-10}}
```

npm access

Set access level on published packages.

More information: https://docs.npmjs.com/cli/npm-access.

- List packages for a user or scope:

```
npm access list packages {{user|scope|scope:team}}
{{package_name}}
```

- List collaborators on a package:

```
npm access list collaborators {{package_name}} {{username}}
```

- Get status of a package:

```
npm access get status {{package_name}}
```

- Set package status (public or private):

```
npm access set status {{public|private}} {{package_name}}
```

- Grant access to a package:

```
npm access grant {{read-only|read-write}} {{scope:team}}
{{package_name}}
```

- Revoke access to a package:

```
npm access revoke {{scope:team}} {{package_name}}
```

- Configure two-factor authentication requirement:

```
npm access set mfa {{none|publish|automation}}
{{package_name}}
```

npm adduser

Add a registry user account.

More information: https://docs.npmjs.com/cli/npm-adduser.

- Create a new user in the specified registry and save credentials to `.npmrc`:

```
npm adduser --registry {{registry_url}}
```

- Log in to a private registry with a specific scope:

```
npm login --scope {{@organization}} --registry {{https://registry.example.com}}
```

- Log out from a specific scope and remove the auth token:

```
npm logout --scope {{@organization}}
```

- Create a scoped package during initialization:

```
npm init --scope {{@organization}} {{[-y|--yes]}}
```

npm audit

Scan for known vulnerabilities in project dependencies.

Reports vulnerabilities and suggests remediation.

More information: https://docs.npmjs.com/cli/npm-audit.

- Scan the project's dependencies for known vulnerabilities:

```
npm audit
```

- Automatically fix vulnerabilities in the project's dependencies:

```
npm audit fix
```

- Force an automatic fix to dependencies with vulnerabilities:

```
npm audit fix {{[-f|--force]}}
```

- Update the lock file without modifying the `node_modules` directory:

```
npm audit fix --package-lock-only
```

- Perform a dry run. Simulate the fix process without making any changes:

```
npm audit fix --dry-run
```

- Output audit results in JSON format:

```
npm audit --json
```

- Configure the audit to only fail on vulnerabilities above a specified severity:

```
npm audit --audit-level {{info|low|moderate|high|critical}}
```

npm author

This command is an alias of **`npm owner`**.

- View documentation for the original command:

  ```
  tldr npm owner
  ```

npm bugs

Report bugs for a package in a web browser.

Attempts to open the package's bug tracker URL or support email.

More information: https://docs.npmjs.com/cli/npm-bugs.

- Report bugs for a specific package by opening the bug tracker for the specified package:

  ```
  npm bugs {{package_name}}
  ```

- Open the bug tracker for the current package by searching for a `package.json` file and using its name:

  ```
  npm bugs
  ```

- Configure the browser used to open URLs by setting your preferred browser for `npm` commands:

  ```
  npm {{[c|config]}} set browser {{browser_name}}
  ```

- Control URL opening: set `browser` to `true` for the system URL opener, or `false` to print URLs in the terminal:

  ```
  npm {{[c|config]}} set browser {{true|false}}
  ```

npm cache

Manage the npm package cache.

More information: https://docs.npmjs.com/cli/npm-cache.

- Add a specific package to the cache:

```
npm cache add {{package_name}}
```

- Remove a specific package from the cache:

```
npm cache remove {{package_name}}
```

- Clear a specific cached item by key:

```
npm cache clean {{key}}
```

- Clear the entire npm cache:

```
npm cache clean --force
```

- List the contents of the npm cache:

```
npm cache ls
```

- Verify the integrity of the npm cache:

```
npm cache verify
```

- Show the cache path:

```
npm {{[c|config]}} get cache
```

- Change the cache path:

```
npm {{[c|config]}} set cache {{path/to/directory}}
```

npm-check

Check for outdated, incorrect, and unused npm package dependencies.

More information: https://github.com/dylang/npm-check.

- Display a report of outdated, incorrect, and unused dependencies:

```
npm-check
```

- Interactively update out-of-date packages:

```
npm-check {{[-u|--update]}}
```

- Update everything without prompting:

```
npm-check {{[-y|--update-all]}}
```

- Don't check for unused packages:

```
npm-check {{[-s|--skip-unused]}}
```

npm ci

Clean install of **`npm`** project dependencies for automated environments.

Installs packages based on **`package-lock.json`** or **`npm-shrinkwrap.json`**.

More information: https://docs.npmjs.com/cli/npm-ci.

- Install project dependencies from `package-lock.json` or `npm-shrinkwrap.json`:

  ```
  npm ci
  ```

- Install project dependencies but skip the specified dependency type:

  ```
  npm ci --omit {{dev|optional|peer}}
  ```

- Install project dependencies without running any pre-/post-scripts defined in `package.json`:

  ```
  npm ci --ignore-scripts
  ```

npm completion

Enable or set up tab-completion for npm commands.

More information: https://docs.npmjs.com/cli/npm-completion.

- Enable tab-completion for the current shell session:

```
npm completion
```

- Add completion to your shell profile (Bash):

```
npm completion >> ~/.bashrc
```

- Add completion to your shell profile (Zsh):

```
npm completion >> ~/.zshrc
```

- Write completions script to a specific file:

```
npm completion > {{path/to/file}}
```

npm config

Manage the **npm** configuration settings.

More information: https://docs.npmjs.com/cli/npm-config.

- Show all configuration settings:

```
npm {{[c|config]}} list
```

- List all configuration settings as JSON:

```
npm {{[c|config]}} list --json
```

- Get the value of a specific configuration key:

```
npm {{[c|config]}} get {{key}}
```

- Set a configuration key to a specific value:

```
npm {{[c|config]}} set {{key}} {{value}}
```

- Delete a configuration key:

```
npm {{[c|config]}} delete {{key}}
```

- Edit the global npm configuration file in the default editor:

```
npm {{[c|config]}} edit
```

- Attempt to repair invalid configuration items:

```
npm {{[c|config]}} fix
```

npm dedupe

Reduce duplication in the **`node_modules`** directory.

More information: https://docs.npmjs.com/cli/npm-dedupe.

- Deduplicate packages in `node_modules`:

  ```
  npm {{[ddp|dedupe]}}
  ```

- Follow `package-lock.json` or `npm-shrinkwrap.json` during deduplication:

  ```
  npm {{[ddp|dedupe]}} --lock
  ```

- Run deduplication in strict mode:

  ```
  npm {{[ddp|dedupe]}} --strict
  ```

- Skip optional/peer dependencies during deduplication:

  ```
  npm {{[ddp|dedupe]}} --omit {{optional|peer}}
  ```

- Enable detailed logging for troubleshooting:

  ```
  npm {{[ddp|dedupe]}} --loglevel verbose
  ```

- Limit deduplication to a specific package:

  ```
  npm {{[ddp|dedupe]}} {{package_name}}
  ```

npm deprecate

Mark a version or range of versions of an **npm** package as deprecated.

More information: https://docs.npmjs.com/cli/npm-deprecate/.

- Deprecate a specific version of a package:

```
npm deprecate {{package_name}}@{{version}}
"{{deprecation_message}}"
```

- Deprecate a range of versions of a package:

```
npm deprecate {{package_name}}@"<{{version_range}}"
"{{deprecation_message}}"
```

- Un-deprecate a specific version of a package:

```
npm deprecate {{package_name}}@{{version}} ""
```

npm diff

Compare package versions from the **`npm`** registry and show differences.

Similar to **`git diff`**.

More information: https://docs.npmjs.com/cli/npm-diff.

- Compare two specific package versions:

```
npm diff --diff {{package_name}}@{{version1}} --diff
{{package_name}}@{{version2}}
```

- Compare current local packages with latest published version:

```
npm diff
```

- Compare current local package with a specific version:

```
npm diff --diff {{package_name}}@{{version}}
```

- Compare a package in the current directory with the registry version:

```
npm diff --diff {{package_name}}
```

- Show only filenames that differ:

```
npm diff --diff-name-only --diff {{package_name}}
@{{version1}} --diff {{package_name}}@{{version2}}
```

- Compare specific files or directories only:

```
npm diff {{path/to/file_or_directory}} --diff
{{package_name}}@{{version1}} --diff {{package_name}}
@{{version2}}
```

- Ignore whitespace when comparing:

```
npm diff --diff-ignore-all-space --diff {{package_name}}
@{{version1}} --diff {{package_name}}@{{version2}}
```

npm dist-tag

Manage distribution tags on packages.

More information: https://docs.npmjs.com/cli/npm-dist-tag.

- List all distribution tags for a package:

```
npm dist-tag ls {{package_name}}
```

- List all distribution tags for the current package:

```
npm dist-tag ls
```

- Add a distribution tag to a specific package version:

```
npm dist-tag add {{package_name}}@{{version}} {{tag}}
```

- Remove a distribution tag from a package:

```
npm dist-tag rm {{package_name}} {{tag}}
```

- Add a tag using the configured tag from npm config:

```
npm dist-tag add {{package_name}}@{{version}}
```

- Add a tag with two-factor authentication:

```
npm dist-tag add {{package_name}}@{{version}} {{tag}} --otp
{{one_time_password}}
```

npm docs

Open the documentation for one or more packages in the default web browser.

More information: https://docs.npmjs.com/cli/npm-docs.

- Open a specific package's documentation page:

```
npm docs {{package}}
```

- Open documentation for multiple packages at once:

```
npm docs {{package1 package2 ...}}
```

npm doctor

Check the health of the npm environment.

More information: https://docs.npmjs.com/cli/npm-doctor.

- Run all default health checks for `npm`:

```
npm doctor
```

- Check the connection to the `npm` registry:

```
npm doctor connection
```

- Check the versions of Node.js and `npm` in use:

```
npm doctor versions
```

- Check for permissions issues with `npm` directories and cache:

```
npm doctor permissions
```

- Validate the cached package files and checksums:

```
npm doctor cache
```

npm edit

Select a dependency in the current project and open the package folder in the default editor (**`$EDITOR`**).

After editing, the package is rebuilt to pick up any changes in compiled packages.

More information: https://docs.npmjs.com/cli/npm-edit.

- Edit a locally installed package in the default editor:

```
npm edit {{package_name}}
```

npm exec

Execute binaries from **`npm`** packages.

More information: https://docs.npmjs.com/cli/npm-exec.

- Execute the command from a local or remote `npm` package:

```
npm {{[x|exec]}} {{command}} {{argument1 argument2 ...}}
```

- Specify the package explicitly (useful if multiple commands with the same name exist):

```
npm {{[x|exec]}} --package {{package}} {{command}}
```

- Run a command if it exists in the current path or in `node_modules/.bin`:

```
npm {{[x|exec]}} --no-install {{command}} {{argument1
argument2 ...}}
```

- Execute a specific command, suppressing any output from `npm` itself:

```
npm {{[x|exec]}} --quiet {{command}} {{argument1
argument2 ...}}
```

- Display help:

```
npm {{[x|exec]}} --help
```

npm explain

Explain how a package is installed, detailing its dependencies and reasons for inclusion.

More information: https://docs.npmjs.com/cli/explain.

- Explain why a specific package is installed:

```
npm explain {{package_name}}
```

- Show explanation in JSON format:

```
npm explain {{package_name}} --json
```

- Include peer dependencies in the explanation:

```
npm explain {{package_name}} --include peer
```

- Limit explanation depth to 2 levels deep:

```
npm explain {{package_name}} --depth 2
```

npm explore

Browse an installed npm package.

More information: https://docs.npmjs.com/cli/npm-explore.

- Browse an installed package:

```
npm explore {{package_name}}
```

- Browse a package and run a command inside it:

```
npm explore {{package_name}} -- {{command}}
```

npm find-dupes

Identify duplicate dependencies in **`node_modules`**.

More information: https://docs.npmjs.com/cli/npm-find-dupes.

- List all duplicate packages within `node_modules`:

  `npm find-dupes`

- Include `devDependencies` in duplicate detection:

  `npm find-dupes --include dev`

- List all duplicate instances of a specific package in `node-modules`:

  `npm find-dupes {{package_name}}`

- Exclude optional dependencies from duplicate detection:

  `npm find-dupes --omit optional`

- Set the logging level for output:

  `npm find-dupes --loglevel {{silent|error|warn|info|verbose}}`

- Output duplicate information in JSON format:

  `npm find-dupes --json`

- Limit duplicate search to specific scopes:

  `npm find-dupes --scope {{@scope1,@scope2}}`

- Exclude specific scopes from duplicate detection:

  `npm find-dupes --omit-scope {{@scope1,@scope2}}`

npm fund

Retrieve funding information from packages.

More information: https://docs.npmjs.com/cli/npm-fund.

- List dependencies with funding URL for the project in the current directory:

```
npm fund
```

- Open the funding URL for a specific package in the default web browser:

```
npm fund {{package}}
```

- List dependencies with a funding URL for a specific workspace for the project in the current directory:

```
npm fund {{[-w|--workspace]}} {{workspace}}
```

npm help-search

Search npm help documentation using a keyword.

More information: https://docs.npmjs.com/cli/npm-help-search.

- Search for commands or topics related to a keyword:

```
npm help-search {{keyword}}
```

- Search with multiple words:

```
npm help-search {{keyword1 keyword2 ...}}
```

npm help

Display an overview of **`npm`** syntax, frequently used commands, and a short description of each command.

More information: https://docs.npmjs.com/cli/npm-help.

- Display the documentation for a specific command:

```
npm help {{command}}
```

npm-home

Open the **npm** page, Yarn page, or GitHub repository of a package in the web browser.

More information: https://github.com/sindresorhus/npm-home.

- Open the npm page of a specific package in the web browser:

```
npm-home {{package}}
```

- Open the GitHub repository of a specific package in the web browser:

```
npm-home {{[-g|--github]}} {{package}}
```

- Open the Yarn page of a specific package in the web browser:

```
npm-home {{[-y|--yarn]}} {{package}}
```

npm hook

Manage **npm** registry hooks for packages.

Note: This command has been deprecated.

More information: https://docs.npmjs.com/cli/v10/hook.

- List all active hooks:

```
npm hook ls
```

- Add a new hook for a package:

```
npm hook add {{package_name}} {{event}} {{target_url}}
```

- Remove a specific hook by its ID:

```
npm hook rm {{hook_id}}
```

- Update a hook with new information:

```
npm hook update {{hook_id}} {{target_url}}
```

npm init

Create a **`package.json`** file.

More information: https://docs.npmjs.com/cli/npm-init.

- Initialize a new package with prompts:

  ```
  npm init
  ```

- Initialize a new package with default values:

  ```
  npm init {{[-y|--yes]}}
  ```

- Initialize a new package using a specific initializer:

  ```
  npm init {{create-react-app}} {{my-app}}
  ```

npm install-ci-test

Install project dependencies and run tests in a single command, mainly used in continuous integration (CI) environments.

Equivalent to running **`npm ci && npm test`**.

More information: https://docs.npmjs.com/cli/install-ci-test.

- Install dependencies and run tests:

  ```
  npm {{[cit|install-ci-test]}}
  ```

- Display detailed logs during installation and testing:

  ```
  npm {{[cit|install-ci-test]}} --verbose
  ```

npm install-test

Equivalent to running **`npm install`** followed by **`npm test`**.

Note: **`it`** can be used as shorthand for **`install-test`**.

More information: https://docs.npmjs.com/cli/v8/commands/npm-install-test.

- Install all dependencies and then run tests:

  ```
  npm install-test
  ```

- Install a specific package and then run tests:

  ```
  npm install-test {{package_name}}
  ```

- Install a package and save it as a dependency before running tests:

  ```
  npm install-test {{package_name}} {{[-S|--save]}}
  ```

- Install dependencies globally and then run tests:

  ```
  npm install-test {{[-g|--global]}}
  ```

npm install

Install Node packages.

More information: https://docs.npmjs.com/cli/npm-install.

- Install dependencies listed in `package.json`:

  ```
  npm {{[i|install]}}
  ```

- Download a specific version of a package and add it to the list of dependencies in `package.json`:

  ```
  npm {{[i|install]}} {{package_name}}@{{version}}
  ```

- Download the latest version of a package and add it to the list of dev dependencies in `package.json`:

  ```
  npm {{[i|install]}} {{[-D|--save-dev]}} {{package_name}}
  ```

- Download the latest version of a package and install it globally:

  ```
  npm {{[i|install]}} {{[-g|--global]}} {{package_name}}
  ```

npm it

This command is an alias of **`npm install-test`**.

- View documentation for the original command:

  ```
  tldr npm install-test
  ```

npm link

Symlink a local package into the global **`node_modules`** or another project for development.

More information: https://docs.npmjs.com/cli/npm-link.

- Symlink the current package globally:

  ```
  npm link
  ```

- Link a globally linked package into another project's `node_modules`:

  ```
  npm link {{package_name}}
  ```

- Unlink a package from the current project:

  ```
  npm unlink {{package_name}}
  ```

npm list

This command is an alias of **`npm ls`**.

- View documentation for the original command:

  ```
  tldr npm ls
  ```

npm login

Log in to a registry user account.

See also: **`npm logout`** for logging out.

More information: https://docs.npmjs.com/cli/npm-login.

- Log in to a registry user account and save the credentials to the `.npmrc` file:

```
npm login
```

- Log in using a custom registry:

```
npm login --registry {{registry_url}}
```

- Log in using a specific authentication strategy:

```
npm login --auth-type {{legacy|web}}
```

npm logout

Log out of the registry user account.

See also: **`npm login`** for logging in.

More information: https://docs.npmjs.com/cli/npm-logout.

- Log out of the registry user account:

```
npm logout
```

- Log out using a custom registry:

```
npm logout --registry {{registry_url}}
```

npm ls

Print installed packages to **`stdout`**.

More information: https://docs.npmjs.com/cli/npm-ls.

- Print all versions of direct dependencies in the current project to `stdout`:

```
npm {{[ls|list]}}
```

- Print all installed packages including peer dependencies:

```
npm {{[ls|list]}} {{[-a|--all]}}
```

- Print all globally installed packages:

```
npm {{[ls|list]}} {{[-g|--global]}}
```

- Print dependencies with extended information:

```
npm {{[ls|list]}} {{[-l|--long]}}
```

- Print dependencies in parseable format:

```
npm {{[ls|list]}} {{[-p|--parseable]}}
```

- Print dependencies in JSON format:

```
npm {{[ls|list]}} --json
```

npm-name

Check whether a package or organization name is available on npm.

More information: https://github.com/sindresorhus/npm-name-cli.

- Check if a specific package name is available in the `npm` registry:

```
npm-name {{package}}
```

- Find similar package names in the `npm` registry:

```
npm-name --similar {{package}}
```

npm org

Manage organizations.

More information: https://docs.npmjs.com/cli/npm-org.

- Add a new user to an organization:

```
npm org set {{organization_name}} {{username}}
```

- Change a user's role in an organization:

```
npm org set {{organization_name}} {{username}} {{developer|admin|owner}}
```

- Remove a user from an organization:

```
npm org rm {{organization_name}} {{username}}
```

- List all users in an organization:

```
npm org ls {{organization_name}}
```

- List all users in an organization, output in JSON format:

```
npm org ls {{organization_name}} --json
```

- Display a user's role in an organization:

```
npm org ls {{organization_name}} {{username}}
```

npm outdated

Check for outdated npm package dependencies.

More information: https://docs.npmjs.com/cli/npm-outdated.

- Find packages that are outdated in a project:

```
npm outdated
```

- Find packages that are outdated regardless of the current project:

```
npm outdated {{[-a|--all]}}
```

npm owner

Manage ownership of published packages.

More information: https://docs.npmjs.com/cli/npm-owner.

- Add a new user as a maintainer of a package:

```
npm owner add {{username}} {{package_name}}
```

- Remove a user from a package's owner list:

```
npm owner rm {{username}} {{package_name}}
```

- List all owners of a package:

```
npm owner ls {{package_name}}
```

npm pack

Create a tarball from a package.

More information: https://docs.npmjs.com/cli/pack.

- Create a tarball from the current package in the current directory:

```
npm pack
```

- Create a tarball from a specific package folder:

```
npm pack {{path/to/package_directory}}
```

- Run a dry run to preview the tarball contents without creating it:

```
npm pack --dry-run
```

- Create a tarball without running lifecycle scripts:

```
npm pack --ignore-scripts
```

- Specify a custom registry to fetch package metadata from:

```
npm pack --registry {{https://registry.npmjs.org/}}
```

npm ping

Ping **`npm`** registry.

More information: https://docs.npmjs.com/cli/npm-ping.

- Ping the default `npm` registry:

```
npm ping
```

- Ping custom `npm` registry:

```
npm ping --registry {{custom_registry_url}}
```

npm pkg

Show or modify **`package.json`** properties.

More information: https://docs.npmjs.com/cli/npm-pkg.

- Get the value of a specific property:

```
npm pkg get {{name}}
```

- Get multiple properties at once:

```
npm pkg get {{name version ...}}
```

- Get multiple values across all workspaces:

```
npm pkg get {{name}} {{version}} {{[--ws|--workspaces]}}
```

- Get a nested or array property value:

```
npm pkg get {{contributors[0].email}}
```

- Set a property to a specific value:

```
npm pkg set {{property}}={{value}}
```

- Set multiple properties at once:

```
npm pkg set {{property1}}={{value1}} {{property2}}={{value2}}
```

- Delete a property from `package.json`:

```
npm pkg delete {{scripts.build}}
```

- Auto-fix common errors in `package.json`:

```
npm pkg fix
```

npm prefix

Print the local or global prefix to **`stdout`**.

More information: https://docs.npmjs.com/cli/npm-prefix.

- Print the local prefix:

```
npm prefix
```

- Print the global prefix:

```
npm prefix {{[-g|--global]}}
```

npm profile

Manage the npm profile and related settings.

Note: This command is unaware of workspaces.

More information: https://docs.npmjs.com/cli/npm-profile.

- View the npm profile details:

  ```
  npm profile get
  ```

- View a specific property of the profile:

  ```
  npm profile get {{property}}
  ```

- Set or update a profile property:

  ```
  npm profile set {{property}} {{value}}
  ```

- Set the public email address:

  ```
  npm profile set email {{email}}
  ```

- Set the public name:

  ```
  npm profile set fullname {{name}}
  ```

- Set a new password interactively:

  ```
  npm profile set password
  ```

- Enable two-factor authentication (2FA) (defaults to `auth-and-writes`):

  ```
  npm profile enable-2fa {{auth-only|auth-and-writes}}
  ```

- Disable two-factor authentication (2FA):

  ```
  npm profile disable-2fa
  ```

npm prune

Remove extraneous packages from **`node_modules`**.

Note: Extraneous packages are those present in the node_modules folder that are not listed as any package's dependency list.

More information: https://docs.npmjs.com/cli/npm-prune.

- Remove all extraneous packages not listed in dependencies:

  ```
  npm prune
  ```

- Remove extraneous packages and devDependencies (useful for production builds):

  ```
  npm prune --production
  ```

- Show what would be removed without making any changes:

  ```
  npm prune --dry-run
  ```

- Output the changes as JSON:

  ```
  npm prune --json
  ```

- Remove specific packages by name:

  ```
  npm prune {{package_name}}
  ```

npm publish

Publish a package to the npm registry.

More information: https://docs.npmjs.com/cli/publish.

- Publish the current package to the default npm registry:

```
npm publish
```

- Publish a package from a specific directory:

```
npm publish {{path/to/package_directory}}
```

- Publish a scoped package with public access:

```
npm publish --access public
```

- Publish a scoped package with restricted (private) access:

```
npm publish --access restricted
```

- Publish a package to a custom registry:

```
npm publish --registry {{https://registry.npmjs.org/}}
```

- Run a dry run to see what would be published without uploading:

```
npm publish --dry-run
```

- Publish a package with a specific distribution tag (e.g., beta):

```
npm publish --tag {{beta}}
```

- Publish with a one-time password for 2FA-enabled accounts:

```
npm publish --otp {{one_time_password}}
```

npm query

Print an array of dependency objects using CSS-like selectors.

More information: https://docs.npmjs.com/cli/npm-query.

- Print direct dependencies:

```
npm query ':root > *'
```

- Print all direct production/development dependencies:

```
npm query ':root > .{{prod|dev}}'
```

- Print dependencies with a specific name:

```
npm query '#{{package}}'
```

- Print dependencies with a specific name and within a semantic versioning range:

```
npm query '#{{package}}@{{semantic_version}}'
```

- Print dependencies which have no dependencies:

```
npm query ':empty'
```

- Find all dependencies with postinstall scripts and uninstall them:

```
npm query ":attr(scripts, [postinstall])" | jq 'map(.name) | join("\n")' {{[-r|--raw-output]}} | xargs -I _ npm uninstall _
```

- Find all Git dependencies and print which application requires them:

```
npm query ":type(git)" | jq 'map(.name)' | xargs -I _ npm why _
```

npm-rb

This command is an alias of **`npm-rebuild`**.

- View documentation for the original command:

  ```
  tldr npm-rebuild
  ```

npm rebuild

Rebuild native Node.js packages after Node or dependency changes.

More information: https://docs.npmjs.com/cli/npm-rebuild.

- Rebuild a specific package:

  ```
  npm {{[rb|rebuild]}} {{package}}
  ```

- Rebuild all installed packages:

  ```
  npm {{[rb|rebuild]}}
  ```

- Rebuild with verbose output:

  ```
  npm {{[rb|rebuild]}} --verbose
  ```

- Rebuild a package in a specific directory:

  ```
  npm {{[rb|rebuild]}} --prefix {{path/to/dir}} {{package}}
  ```

- Rebuild without using the npm cache:

  ```
  npm {{[rb|rebuild]}} --no-cache
  ```

- Rebuild in global mode:

  ```
  npm {{[rb|rebuild]}} {{[-g|--global]}}
  ```

npm repo

Open the repository page of a package in the browser.

More information: https://docs.npmjs.com/cli/npm-repo.

- Open the repository page of the current project (based on `package.json`):

  ```
  npm repo
  ```

- Open the repository page of a specific package from the registry:

  ```
  npm repo {{package_name}}
  ```

- Open repository pages for multiple packages:

  ```
  npm repo {{package_name1 package_name2 ...}}
  ```

- Print the repository URL instead of opening it in the browser:

  ```
  npm repo --browser false
  ```

- Open the repository page for a package in a specific browser:

  ```
  npm repo --browser {{browser_name}}
  ```

- Open the repository page of a package in a specific workspace:

  ```
  npm repo --workspace {{workspace_name}}
  ```

npm restart

This command is an alias of **`npm run restart`**.

- View documentation for the original command:

  ```
  tldr npm run
  ```

npm root

Display path to **`node_modules`** directory.

More information: https://docs.npmjs.com/cli/npm-root.

- Display path to the local `node_modules` directory:

```
npm root
```

- Display path to the global `node_modules` directory:

```
npm root {{[-g|--global]}}
```

npm run-script

This command is an alias of **`npm run`**.

- View documentation for the original command:

```
tldr npm run
```

npm run

Run a script.

More information: https://docs.npmjs.com/cli/npm-run.

- Run a script:

```
npm run {{script_name}}
```

- Pass arguments to a script:

```
npm run {{script_name}} -- {{argument}} {{--option}}
```

- Run a script named `start`:

```
npm start
```

- Run a script named `stop`:

```
npm stop
```

- Run a script named `restart`:

```
npm restart
```

- Run a script named `test`:

```
npm {{[t|test]}}
```

npm sbom

Generate a Software Bill of Materials (SBOM) for your Node.js project.

More information: https://docs.npmjs.com/cli/npm-sbom.

- Output a list of all dependencies in your project:

```
npm sbom
```

- Exclude both `dev` and `optional` dependencies:

```
npm sbom --omit dev --omit optional
```

- Generate an SBOM based only on the `package-lock.json`:

```
npm sbom --package-lock-only
```

npm search

Search for packages in the **npm** registry.

More information: https://docs.npmjs.com/cli/npm-search.

- Search for a package by name:

```
npm {{[s|search]}} {{package}}
```

- Search for packages by a specific keyword:

```
npm {{[s|search]}} {{keyword}}
```

- Search for packages, including detailed information (e.g., description, author, version):

```
npm {{[s|search]}} {{package}} --long
```

- Search for packages maintained by a specific author:

```
npm {{[s|search]}} --author {{author}}
```

- Search for packages with a specific organization:

```
npm {{[s|search]}} --scope {{organization}}
```

- Search for packages with a specific combination of terms:

```
npm {{[s|search]}} {{term1 term2 ...}}
```

npm shrinkwrap

Lock down the dependencies of a package, creating a **`npm-shrinkwrap.json`** file.

Similar to **`package-lock.json`** but intended for published packages.

More information: https://docs.npmjs.com/cli/shrinkwrap.

- Generate a `npm-shrinkwrap.json` file from the current `package-lock.json`:

  ```
  npm shrinkwrap
  ```

- Run in production mode (excludes devDependencies):

  ```
  npm shrinkwrap --production
  ```

- Force recreate the shrinkwrap file even if it already exists:

  ```
  npm shrinkwrap --force
  ```

npm star

Mark a package as favorite.

More information: https://docs.npmjs.com/cli/npm-star.

- Star a public package from the default registry:

```
npm star {{package_name}}
```

- Star a package within a specific scope:

```
npm star @{{scope}}/{{package_name}}
```

- Star a package from a specific registry:

```
npm star {{package_name}} --registry {{registry_url}}
```

- Star a private package that requires authentication:

```
npm star {{package_name}} --auth-type {{legacy|oauth|web|saml}}
```

- Star a package by providing an OTP for two-factor authentication:

```
npm star {{package_name}} --otp {{otp}}
```

- Star a package with detailed logging:

```
npm star {{package_name}} --loglevel verbose
```

- List all your starred packages:

```
npm star --list
```

- List your starred packages from a specific registry:

```
npm star --list --registry {{registry_url}}
```

npm start

This command is an alias of **`npm run start`**.

- View documentation for the original command:

  ```
  tldr npm run
  ```

npm stop

This command is an alias of **`npm run stop`**.

- View documentation for the original command:

  ```
  tldr npm run
  ```

npm team

Manage teams in an organization on the **npm** registry.

More information: https://docs.npmjs.com/cli/npm-team.

- Add a user to a team in an organization:

  ```
  npm team add {{organization:team}} {{username}}
  ```

- Remove a user from a team:

  ```
  npm team rm {{organization:team}} {{username}}
  ```

- Create a new team in an organization:

  ```
  npm team create {{organization:team}}
  ```

- Delete a team from an organization:

  ```
  npm team destroy {{organization:team}}
  ```

- List all teams in an organization:

  ```
  npm team ls {{organization}}
  ```

- List all users in a specific team:

  ```
  npm team ls {{organization:team}}
  ```

npm test

This command is an alias of **`npm run test`**.

- View documentation for the original command:

  ```
  tldr npm run
  ```

npm token

Manage and generate authentication tokens for the npm registry.

More information: https://docs.npmjs.com/cli/npm-token.

- Create a new authentication token:

  ```
  npm token create
  ```

- List all tokens associated with an account:

  ```
  npm token list
  ```

- Delete a specific token using its token ID:

  ```
  npm token revoke {{token_id}}
  ```

- Create a token with read-only access:

  ```
  npm token create --read-only
  ```

- Create a token with publish access:

  ```
  npm token create --publish
  ```

- Automatically configure an npm token in your global `.npmrc` file when you log in:

  ```
  npm login
  ```

- Remove a token from the global configuration:

  ```
  npm token revoke {{token_id}}
  ```

npm uninstall

Remove a package.

More information: https://docs.npmjs.com/cli/npm-uninstall.

- Remove a package from the current project:

```
npm {{[r|uninstall]}} {{package_name}}
```

- Remove a package globally:

```
npm {{[r|uninstall]}} {{[-g|--global]}} {{package_name}}
```

- Remove multiple packages at once:

```
npm {{[r|uninstall]}} {{package_name1 package_name2 ...}}
```

npm unpublish

Remove a package from the npm registry.

More information: https://docs.npmjs.com/cli/npm-unpublish.

- Unpublish a specific package version:

```
npm unpublish {{package_name}}@{{version}}
```

- Unpublish the entire package:

```
npm unpublish {{package_name}} {{[-f|--force]}}
```

- Unpublish a package that is scoped:

```
npm unpublish @{{scope}}/{{package_name}}
```

- Specify a timeout period before unpublishing:

```
npm unpublish {{package_name}} --timeout
{{time_in_milliseconds}}
```

- To prevent accidental unpublishing, use the `--dry-run` flag to see what would be unpublished:

```
npm unpublish {{package_name}} --dry-run
```

npm unstar

Remove the favorite/star mark from a package.

More information: https://docs.npmjs.com/cli/npm-unstar.

- Unstar a public package from the default registry:

```
npm unstar {{package_name}}
```

- Unstar a package within a specific scope:

```
npm unstar @{{scope}}/{{package_name}}
```

- Unstar a package from a specific registry:

```
npm unstar {{package_name}} --registry {{registry_url}}
```

- Unstar a private package that requires authentication:

```
npm unstar {{package_name}} --auth-type {{legacy|oauth|web|saml}}
```

- Unstar a package by providing an OTP for two-factor authentication:

```
npm unstar {{package_name}} --otp {{otp}}
```

- Unstar a package with a specific logging level:

```
npm unstar {{package_name}} --loglevel {{silent|error|warn|notice|http|timing|info|verbose|silly}}
```

npm update

Update packages in the current project.

More information: https://docs.npmjs.com/cli/npm-update.

- Update all packages in the current project:

```
npm {{[up|update]}}
```

- Update a specific package in the current project:

```
npm {{[up|update]}} {{package}}
```

- Update a package globally:

```
npm {{[up|update]}} {{[-g|--global]}} {{package}}
```

- Update multiple packages at once:

```
npm {{[up|update]}} {{package1 package2 ...}}
```

npm version

Bump a node package version.

More information: https://docs.npmjs.com/cli/npm-version.

- Display version:

```
npm version
```

- Bump the minor version:

```
npm version minor
```

- Set a specific version:

```
npm version {{version}}
```

- Bump the patch version without creating a Git tag:

```
npm version patch --no-git-tag-version
```

- Bump the major version with a custom commit message:

```
npm version major {{[-m|--message]}} "{{Upgrade to %s for reasons}}"
```

npm view

View registry information about a package.

More information: https://docs.npmjs.com/cli/npm-view.

- View information about the latest version of a package:

```
npm {{[v|view]}} {{package}}
```

- View information about a specific version of a package:

```
npm {{[v|view]}} {{package}}@{{version}}
```

- View all available versions of a package:

```
npm {{[v|view]}} {{package}} versions
```

- View the description of a package:

```
npm {{[v|view]}} {{package}} description
```

- View the dependencies of the latest version of a package:

```
npm {{[v|view]}} {{package}} dependencies
```

- View the repository URL of a package:

```
npm {{[v|view]}} {{package}} repository
```

- View the maintainers of a package:

```
npm {{[v|view]}} {{package}} maintainers
```

npm whoami

Display npm username.

More information: https://docs.npmjs.com/cli/whoami.

- Display username of the currently logged-in user:

```
npm whoami
```

- Display username of the current user in the specific registry:

```
npm whoami --registry {{registry_url}}
```

npm-why

Identifies why an npm package is installed.

More information: https://github.com/amio/npm-why.

- Show why an `npm` package is installed:

```
npm-why {{package}}
```

npm

JavaScript and Node.js package manager.

Manage Node.js projects and their module dependencies.

More information: https://docs.npmjs.com/cli/npm.

- Create a `package.json` file with default values (omit `--yes` to do it interactively):

```
npm init {{[-y|--yes]}}
```

- Download all the packages listed as dependencies in `package.json`:

```
npm {{[i|install]}}
```

- Download a specific version of a package and add it to the list of dependencies in `package.json`:

```
npm {{[i|install]}} {{package_name}}@{{version}}
```

- Download the latest version of a package and add it to the list of dev dependencies in `package.json`:

```
npm {{[i|install]}} {{package_name}} {{[-D|--save-dev]}}
```

- Download the latest version of a package and install it globally:

```
npm {{[i|install]}} {{package_name}} {{[-g|--global]}}
```

- Uninstall a package and remove it from the list of dependencies in `package.json`:

```
npm {{[r|uninstall]}} {{package_name}}
```

- List all locally installed dependencies:

```
npm {{[ls|list]}}
```

- List all top-level globally installed packages:

```
npm {{[ls|list]}} {{[-g|--global]}} --depth {{0}}
```

nproc

Print the number of processing units (normally CPUs) available.

More information: https://www.gnu.org/software/coreutils/manual/html_node/nproc-invocation.html.

- Display the number of available processing units:

```
nproc
```

- Display the number of installed processing units, including any inactive ones:

```
nproc --all
```

- If possible, subtract a given number of units from the returned value:

```
nproc --ignore {{count}}
```

npx

This command is an alias of **`npm exec`**.

- View documentation for the original command:

```
tldr npm exec
```

nrm

npm registry manager.

Helps to easily switch between different npm registries.

More information: https://github.com/Pana/nrm.

- List all registries:

```
nrm ls
```

- Change to a particular registry:

```
nrm use {{registry}}
```

- Show the response time for all registries:

```
nrm test
```

- Add a custom registry:

```
nrm add {{registry}} {{url}}
```

- Delete a registry:

```
nrm del {{registry}}
```

nslookup

Query name servers for various domain records.

More information: https://manned.org/nslookup.

- Query your system's default name server for an IP address (A record) of the domain:

```
nslookup {{example.com}}
```

- Query a given name server for a NS record of the domain:

```
nslookup -type=NS {{example.com}} {{8.8.8.8}}
```

- Query for a reverse lookup (PTR record) of an IP address:

```
nslookup -type=PTR {{54.240.162.118}}
```

- Query for ANY available records using TCP protocol:

```
nslookup -vc -type=ANY {{example.com}}
```

- Query a given name server for the whole zone file (zone transfer) of the domain using TCP protocol:

```
nslookup -vc -type=AXFR {{example.com}} {{name_server}}
```

- Query for a mail server (MX record) of the domain, showing details of the transaction:

```
nslookup -type=MX -debug {{example.com}}
```

- Query a given name server on a specific port number for a TXT record of the domain:

```
nslookup -port={{port_number}} -type=TXT {{example.com}}
{{name_server}}
```

ntfs-read.py

A read-only NTFS explorer for accessing and extracting files from NTFS volumes.

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- Open an NTFS volume for exploration (e.g., `C:\.\` or `/dev/disk1s1`):

```
ntfs-read.py {{volume}}
```

- Extract a specific file from an NTFS volume (e.g., `\windows\system32\config\sam`):

```
ntfs-read.py -extract {{\windows\system32\config\sam}} {{volume}}
```

- Enable debug output:

```
ntfs-read.py -debug {{volume}}
```

- Display help:

```
ntfs-read.py --help
```

ntfy

Send and receive HTTP POST notifications.

More information: https://github.com/binwiederhier/ntfy.

- Send a message to the `security` topic:

```
ntfy pub security "{{Front door has been opened.}}"
```

- Send with a title, priority and tags:

```
ntfy publish --title="{{Someone bought your item}}" --
priority={{high}} --tags={{duck}} {{ebay}} "{{Someone just
bought your item: Platypus Sculpture}}"
```

- Send at 8:30am:

```
ntfy pub --at=8:30am {{delayed_topic}} "{{Time for school,
sleepyhead...}}"
```

- Trigger a webhook:

```
ntfy trigger {{my_webhook}}
```

- Subscribe to a topic (`<Ctrl c>` to stop listening):

```
ntfy sub {{home_automation}}
```

- Display help:

```
ntfy --help
```

ntfyme

A notification tool to track and notify you about your long-running termination process.

Send notifications with success/error messages with Gmail, Telegram, and more.

More information: https://github.com/AnirudhG07/ntfyme.

- Directly run your command:

```
ntfyme exec {{[-c|--cmd]}} {{command}}
```

- Pipe your command and run:

```
echo {{command}} | ntfyme exec
```

- Run multiple commands by enclosing them in quotes:

```
echo "{{command1; command2; command3}}" | ntfyme exec
```

- Track and terminate the process after prolonged suspension:

```
ntfyme exec {{[-t|--track-process]}} {{[-c|--cmd]}} {{command}}
```

- Setup the tool configurations interactively:

```
ntfyme setup
```

- Encrypt your password:

```
ntfyme enc
```

- See the log history:

```
ntfyme log
```

- Open and edit the configuration file:

```
ntfyme config
```

nth

Name That Hash - Instantly name the type of any hash.

More information: https://github.com/hashpals/name-that-hash.

- Name a hash:

```
nth {{[-t|--text]}} {{5f4dcc3b5aa765d61d8327deb882cf99}}
```

- Name hashes in a file:

```
nth {{[-f|--file]}} {{path/to/hashes}}
```

- Print in JSON format:

```
nth {{[-t|--text]}} {{5f4dcc3b5aa765d61d8327deb882cf99}} {{[-g|--greppable]}}
```

- Decode hash in Base64 before naming it:

```
nth {{[-t|--text]}} {{NWY0ZGNjM2I1YWE3NjVkNjFkODMyN2RlYjg4MmNmOTkK}} {{[-b64|--base64]}}
```

ntl

This command is an alias of **`netlify`**.

- View documentation for the original command:

```
tldr netlify
```

ntp-ctl

Management client for the **`ntpd-rs`** daemon.

More information: https://docs.ntpd-rs.pendulum-project.org/man/ntp-ctl.8.

- Display information about the current state of the NTP daemon:

```
ntp-ctl status
```

- Check if the specified configuration file (default: `/etc/ntpd-rs/ntp.toml`) is valid:

```
ntp-ctl {{[-c|--config]}} {{path/to/config}} validate
```

- Interactively run a single synchronization of the clock:

```
sudo ntp-ctl force-sync
```

ntpctl

Display information about the running instance of OpenNTPD.

More information: https://man.openbsd.org/ntpctl.

- Show all data:

```
ntpctl -s {{[a|all]}}
```

- Show information about each peer:

```
ntpctl -s {{[p|peers]}}
```

- Show the status of peers and sensors, and whether the system clock is synced:

```
ntpctl -s {{[s|status]}}
```

- Show information about each sensor:

```
ntpctl -s {{[S|Sensors]}}
```

nu

Nushell ("a new type of shell") takes a modern, structured approach to your command-line.

See also: **`elvish`**.

More information: https://www.nushell.sh.

- Start an interactive shell session:

```
nu
```

- Execute specific commands:

```
nu --commands "{{echo 'nu is executed'}}"
```

- Execute a specific script:

```
nu {{path/to/script.nu}}
```

- Execute a specific script with logging:

```
nu --log-level {{error|warn|info|debug|trace}} {{path/to/
script.nu}}
```

nuclei

Fast and customizable vulnerability scanner using a simple YAML-based DSL.

More information: https://docs.projectdiscovery.io/tools/nuclei/running.

- Update `nuclei` templates to the latest released version (downloaded to `~/nuclei-templates` on macOS/Linux or `%USERPROFILE%\nuclei-templates` on Windows):

```
nuclei {{[-ut|-update-templates]}}
```

- [l]ist all [t]emplates by specific [p]rotocol [t]ype:

```
nuclei -tl {{[-pt|-type]}} {{dns|file|http|headless|tcp|workflow|ssl|websocket|whois|code|javascript}}
```

- Run an automatic web scan using Wappalyzer technology detection for a specific target [u]RL/host:

```
nuclei {{[-as|-automatic-scan]}} {{[-u|-target]}} {{example.com}}
```

- Run HTTP [p]rotocol [t]ype templates of specific severity, exporting results to markdown files inside a specific directory:

```
nuclei {{[-s|-severity]}} {{high,critical,...}} {{[-pt|-type]}} http {{[-u|-target]}} {{https://example.com}} {{[-me|-markdown-export]}} {{path/to/directory}}
```

- Run all templates with a custom rate limit, maximum bulk size, and silent output (only findings shown):

```
nuclei {{[-rl|-rate-limit]}} {{150}} {{[-bs|-bulk-size]}} {{25}} {{[-c|-concurrency]}} {{25}} -silent {{[-u|-target]}} {{https://example.com}}
```

- Run a specific nuclei-bundled workflow against a target:

```
nuclei {{[-w|-workflows]}} {{workflows/wordpress-workflow.yaml}} {{[-u|-target]}} {{https://example.com}}
```

- Run one or more specific templates or directory with templates with verbose output in `stderr` and output detected issues/vulnerabilities to a file:

```
nuclei {{[-t|-templates]}} {{path/to/nuclei-templates/http}}
{{[-u|-target]}} {{https://example.com}} {{[-v|-verbose]}}
{{[-o|-output]}} {{path/to/results}}
```

- Run a scan based on one or more template conditions:

```
nuclei {{[-tc|-template-condition]}} "{{contains(tags, 'xss')
&& contains(tags, 'cve')}}" {{[-u|-target]}} {{https://
example.com}}
```

nudoku

Sudoku game in terminal.

More information: https://jubalh.github.io/nudoku/.

- Start a sudoku game:

  ```
  nudoku
  ```

- Choose the difficulty of the game:

  ```
  nudoku -d {{easy|normal|hard}}
  ```

- Navigate the board:

  ```
  {{<h>|<j>|<k>|<l>|<ArrowKeys>}}
  ```

- Delete a number:

  ```
  {{<Backspace>|<x>}}
  ```

- Get a hint:

  ```
  <H>
  ```

- See the complete solution:

  ```
  <S>
  ```

- Create a new puzzle:

  ```
  <N>
  ```

- Quit the game:

  ```
  <Q>
  ```

numfmt

Convert numbers to and from human-readable strings.

More information: https://www.gnu.org/software/coreutils/manual/html_node/numfmt-invocation.html.

- Convert 1.5K (SI Units) to 1500:

```
numfmt --from si 1.5K
```

- Convert 5th field (1-indexed) to IEC Units without converting header:

```
ls -l | numfmt --header=1 --field 5 --to iec
```

- Convert to IEC units, pad with 5 characters, left aligned:

```
du {{[-s|--summarize]}} * | numfmt --to iec --format "%-5f"
```

nvcc

The NVIDIA CUDA Compiler Driver.

More information: https://docs.nvidia.com/cuda/cuda-compiler-driver-nvcc.

- Compile a CUDA program:

```
nvcc {{path/to/source.cu}} {{[-o|--output-file]}} {{path/to/executable}}
```

- Generate debug information:

```
nvcc {{path/to/source.cu}} {{[-o|--output-file]}} {{path/to/executable}} {{[-g|--debug]}} {{[-G|--device-debug]}}
```

- Include libraries from a different path:

```
nvcc {{path/to/source.cu}} {{[-o|--output-file]}} {{path/to/executable}} {{[-I|--include-path]}} {{path/to/includes}} {{[-L|--library-path]}} {{path/to/library}} {{[-l|--library]}} {{library_name}}
```

- Specify the compute capability for a specific GPU architecture:

```
nvcc {{path/to/source.cu}} {{[-o|--output-file]}} {{path/to/executable}} {{[-gencode|--generate-code]}} arch={{arch_name}},code={{gpu_code_name}}
```

nvidia-smi mig

Manage Nvidia multi-instance GPUs.

More information: https://docs.nvidia.com/datacenter/tesla/mig-user-guide/index.html.

- Create a compute instance from device 0:

```
nvidia-smi mig {{[-cgi|--create-gpu-instance]}} {{0}} {{[-C|--default-compute-instance]}}
```

- List GPU instances:

```
nvidia-smi mig {{[-lgi|--list-gpu-instances]}}
```

- Display help:

```
nvidia-smi mig {{[-h|--help]}}
```

nvidia-smi

Aid the management and monitoring of NVIDIA GPU devices.

More information: https://developer.nvidia.com/nvidia-system-management-interface.

- Display information on all available GPUs and processes using them:

```
nvidia-smi
```

- Display more detailed GPU information:

```
nvidia-smi {{[-q|--query]}}
```

- Monitor overall GPU usage with 1-second update interval:

```
nvidia-smi dmon
```

nvim

Neovim, a programmer's text editor based on Vim, provides several modes for different kinds of text manipulation.

Pressing **`<i>`** in normal mode enters insert mode. **`<Esc>`** or **`<Ctrl c>`** goes back to normal mode, which doesn't allow regular text insertion.

See also: **`vim`**, **`vimtutor`**, **`vimdiff`**.

More information: https://neovim.io.

- Open a file:

  ```
  nvim {{path/to/file}}
  ```

- Enter text editing mode (insert mode):

  ```
  <Esc><i>
  ```

- Copy ("yank") or cut ("delete") the current line (paste it with `<p>`):

  ```
  <Esc>{{<y><y>|<d><d>}}
  ```

- Enter normal mode and undo the last operation:

  ```
  <Esc><u>
  ```

- Search for a pattern in the file (press `<n>`/`<N>` to go to next/previous match):

  ```
  <Esc></>{{search_pattern}}<Enter>
  ```

- Perform a `regex` substitution in the whole file:

  ```
  <Esc><:>%s/{{regex}}/{{replacement}}/g<Enter>
  ```

- Enter normal mode and save (write) the file, and quit:

  ```
  {{<Esc><Z><Z>|<Esc><:>x<Enter>|<Esc><:>wq<Enter>}}
  ```

- Quit without saving:

  ```
  <Esc><:>q!<Enter>
  ```

nvm

Install, uninstall, or switch between Node.js versions under the fish shell.

Supports version numbers like "12.8" or "v16.13.1", and labels like "stable", "system", etc.

More information: https://github.com/jorgebucaran/nvm.fish.

- Install a specific version of Node.js:

```
nvm install {{node_version}}
```

- Use a specific version of Node.js in the current shell:

```
nvm use {{node_version}}
```

- Set the default Node.js version:

```
set nvm_default_version {{node_version}}
```

- List all available Node.js versions and highlight the default one:

```
nvm list
```

- Uninstall a given Node.js version:

```
nvm uninstall {{node_version}}
```

nvm

Install, uninstall or switch between Node.js versions.

Supports version numbers like "12.8" or "v16.13.1", and labels like "stable", "system", etc.

See also: **`asdf`**.

More information: https://github.com/creationix/nvm.

- Install a specific version of Node.js:

```
nvm install {{node_version}}
```

- Use a specific version of Node.js in the current shell:

```
nvm use {{node_version}}
```

- Set the default Node.js version:

```
nvm alias default {{node_version}}
```

- List all available Node.js versions and highlight the default one:

```
nvm list
```

- Uninstall a given Node.js version:

```
nvm uninstall {{node_version}}
```

- Launch the REPL of a specific version of Node.js:

```
nvm run {{node_version}} --version
```

- Execute a script in a specific version of Node.js:

```
nvm exec {{node_version}} node {{app.js}}
```

nvme

NVMe storage user space utility.

More information: https://github.com/linux-nvme/nvme-cli.

- List all nvme devices:

```
sudo nvme list
```

- Show device information:

```
sudo nvme smart-log {{device}}
```

nx

Manage **nx** workspaces.

More information: https://nx.dev/l/r/getting-started/nx-cli.

- Build a specific project:

```
nx build {{project}}
```

- Test a specific project:

```
nx test {{project}}
```

- Execute a target on a specific project:

```
nx run {{project}}:{{target}}
```

- Execute a target on multiple projects:

```
nx run-many --target {{target}} --projects {{project1}},
{{project2}}
```

- Execute a target on all projects in the workspace:

```
nx run-many --target {{target}} --all
```

- Execute a target only on projects that have been changed:

```
nx affected --target {{target}}
```

nxc ftp

Pentest and exploit FTP servers.

More information: https://www.netexec.wiki/ftp-protocol.

- Search for valid credentials by trying out every combination in the specified lists of usernames and passwords:

```
nxc ftp {{192.168.178.2}} {{[-u|--username]}} {{path/to/
usernames.txt}} {{[-p|--password]}} {{path/to/passwords.txt}}
```

- Continue searching for valid credentials even after valid credentials have been found:

```
nxc ftp {{192.168.178.2}} {{[-u|--username]}} {{path/to/
usernames.txt}} {{[-p|--password]}} {{path/to/passwords.txt}}
--continue-on-success
```

- Perform directory listings on each FTP server the supplied credentials are valid on:

```
nxc ftp {{192.168.178.0/24}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --ls
```

- Download the specified file from the target server:

```
nxc ftp {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --get {{path/to/file}}
```

- Upload the specified file to the target server at the specified location:

```
nxc ftp {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --put {{path/to/local_file}}
{{path/to/remote_location}}
```

nxc ldap

Pentest and exploit Windows Active Directory Domains via LDAP.

More information: https://www.netexec.wiki/ldap-protocol.

- Search for valid domain credentials by trying out every combination in the specified lists of usernames and passwords:

```
nxc ldap {{192.168.178.2}} {{[-u|--username]}} {{path/to/
usernames.txt}} {{[-p|--password]}} {{path/to/passwords.txt}}
```

- Enumerate active domain users:

```
nxc ldap {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --active-users
```

- Collect data about the targeted domain and automatically import these data into BloodHound:

```
nxc ldap {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --bloodhound {{[-c|--
collection]}} {{All}}
```

- Attempt to collect AS_REP messages for the specified user in order to perform an ASREPRoasting attack:

```
nxc ldap {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} '' --asreproast {{path/to/output.txt}}
```

- Attempt to extract the passwords of group managed service accounts on the domain:

```
nxc ldap {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --gmsa
```

nxc mssql

Pentest and exploit Microsoft SQL servers.

More information: https://www.netexec.wiki/mssql-protocol.

- Search for valid credentials by trying out every combination in the specified lists of usernames and passwords:

```
nxc mssql {{192.168.178.2}} {{[-u|--username]}} {{path/to/
usernames.txt}} {{[-p|--password]}} {{path/to/passwords.txt}}
```

- Execute the specified SQL query on the target server:

```
nxc mssql {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} {{[-q|--query]}} '{{SELECT *
FROM sys.databases;}}'
```

- Execute the specified shell command on the target server through MSSQL:

```
nxc mssql {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} -x {{whoami}}
```

- Execute the specified PowerShell command on the target server through MSSQL without retrieving output:

```
nxc mssql {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} -X {{whoami}} --no-output
```

- Download a remote file from the target server and store it in the specified location:

```
nxc mssql {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --get-file {{C:
\path\to\remote_file}} {{path/to/local_file}}
```

- Upload a local file to the specified location on the target server:

```
nxc mssql {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --put-file {{path/to/
local_file}} {{C:\path\to\remote_file}}
```

nxc nfs

Pentest and exploit NFS servers. Currently supports anonymous mode only.

More information: https://www.netexec.wiki/nfs-protocol.

- Detect the version of a remote NFS server:

```
nxc nfs {{192.168.178.0/24}}
```

- List the available NFS shares:

```
nxc nfs {{192.168.178.2}} --shares
```

- Enumerate the exposed shares recursively to the specified depth:

```
nxc nfs {{192.168.178.2}} --enum-shares {{5}}
```

- Download the specified remote file:

```
nxc nfs {{192.168.178.2}} --get-file {{path/to/remote_file}}
{{path/to/local_file}}
```

- Upload the specified local file to the remote share:

```
nxc nfs {{192.168.178.2}} --put-file {{path/to/local_file}}
{{path/to/remote_file}}
```

nxc rdp

Pentest and exploit RDP servers.

More information: https://www.netexec.wiki/rdp-protocol.

- Search for valid credentials by trying out every combination in the specified lists of usernames and passwords:

```
nxc rdp {{192.168.178.2}} {{[-u|--username]}} {{path/to/
usernames.txt}} {{[-p|--password]}} {{path/to/passwords.txt}}
```

- Take a screenshot after waiting the for specified number of seconds:

```
nxc rdp {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --screenshot --screentime
{{10}}
```

- Take a screenshot in the specified resolution:

```
nxc rdp {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --screenshot --res
{{1024x768}}
```

- Take a screenshot of the RDP login prompt if Network Level Authentication is disabled:

```
nxc rdp {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --nla-screenshot
```

nxc smb

Pentest and exploit SMB servers.

More information: https://www.netexec.wiki/smb-protocol.

- Search for valid domain credentials by trying out every combination in the specified lists of usernames and passwords:

```
nxc smb {{192.168.178.2}} {{[-u|--username]}} {{path/to/
usernames.txt}} {{[-p|--password]}} {{path/to/passwords.txt}}
```

- Search for valid credentials for local accounts instead of domain accounts:

```
nxc smb {{192.168.178.2}} {{[-u|--username]}} {{path/to/
usernames.txt}} {{[-p|--password]}} {{path/to/passwords.txt}}
--local-auth
```

- Enumerate SMB shares and the specified users' access rights to them on the target hosts:

```
nxc smb {{192.168.178.0/24}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --shares
```

- Enumerate network interfaces on the target hosts, performing authentication via pass-the-hash:

```
nxc smb {{192.168.178.30-45}} {{[-u|--username]}}
{{username}} {{[-H|--hash]}} {{NTLM_hash}} --interfaces
```

- Scan the target hosts for common vulnerabilities:

```
nxc smb {{path/to/target_list.txt}} {{[-u|--username]}}
'' {{[-p|--password]}} '' {{[-M|--module]}} zerologon {{[-
M|--module]}} petitpotam
```

- Attempt to execute a command on the target hosts:

```
nxc smb {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} -x {{command}}
```

nxc ssh

Pentest and exploit SSH servers.

See also: **`hydra`**.

More information: https://www.netexec.wiki/ssh-protocol.

- Spray the specified password against a list of usernames on the specified target:

```
nxc ssh {{192.168.178.2}} {{[-u|--username]}} {{path/to/
usernames.txt}} {{[-p|--password]}} {{password}}
```

- Search for valid credentials by trying out every combination in the specified lists of usernames and passwords:

```
nxc ssh {{192.168.178.2}} {{[-u|--username]}} {{path/to/
usernames.txt}} {{[-p|--password]}} {{path/to/passwords.txt}}
```

- Use the specified private key for authentication, using the supplied password as the key's passphrase:

```
nxc ssh {{192.186.178.2}} {{[-u|--username]}} {{path/to/
usernames.txt}} {{[-p|--password]}} {{password}} --key-file
{{path/to/id_rsa}}
```

- Try a combination of username and password on a number of targets:

```
nxc ssh {{192.168.178.0/24}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}}
```

- Check for `sudo` privileges on a successful login:

```
nxc ssh {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{path/to/passwords.txt}} --sudo-check
```

nxc vnc

Pentest and exploit VNC servers.

More information: https://www.netexec.wiki/getting-started/selecting-and-using-a-protocol.

- Search for valid credentials by trying out every combination in the specified lists of usernames and passwords:

```
nxc vnc {{192.168.178.2}} {{[-u|--username]}} {{path/to/usernames.txt}} {{[-p|--password]}} {{path/to/passwords.txt}}
```

- Avoid rate limiting through VNC-sleep:

```
nxc vnc {{192.168.178.2}} {{[-u|--username]}} {{path/to/usernames.txt}} {{[-p|--password]}} {{path/to/passwords.txt}} --vnc-sleep {{10}}
```

- Take a screenshot on the remote system after waiting the specified amount of time:

```
nxc vnc {{192.168.178.2}} {{[-u|--username]}} {{username}} {{[-p|--password]}} {{password}} --screenshot --screentime {{10}}
```

nxc winrm

Pentest and exploit Windows Remote Management (winrm).

More information: https://www.netexec.wiki/winrm-protocol.

- Search for valid credentials by trying out every combination in the specified lists of usernames and passwords:

```
nxc winrm {{192.168.178.2}} {{[-u|--username]}} {{path/to/
usernames.txt}} {{[-p|--password]}} {{path/to/passwords.txt}}
```

- Specify the domain to authenticate to (avoids an initial SMB connection):

```
nxc winrm {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} -d {{domain_name}}
```

- Execute the specified command on the host:

```
nxc winrm {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} -x {{whoami}}
```

- Execute the specified PowerShell command on the host as administrator using LAPS:

```
nxc winrm {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --laps -X {{whoami}}
```

nxc wmi

Pentest and exploit the Windows Management Instrumentation (WMI).

More information: https://www.netexec.wiki/wmi-protocol.

- Search for valid credentials by trying out every combination in the specified lists of usernames and passwords:

```
nxc wmi {{192.168.178.2}} {{[-u|--username]}} {{path/to/
usernames.txt}} {{[-p|--password]}} {{path/to/passwords.txt}}
```

- Authenticate via local authentication (as opposed to authenticating to the domain):

```
nxc wmi {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --local-auth
```

- Issue the specified WMI query:

```
nxc wmi {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} --wmi {{wmi_query}}
```

- Execute the specified command on the targeted host:

```
nxc wmi {{192.168.178.2}} {{[-u|--username]}} {{username}}
{{[-p|--password]}} {{password}} -x {{command}}
```

nxc

Network service enumeration and exploitation tool.

Some subcommands such as **`smb`** have their own usage documentation.

More information: https://www.netexec.wiki/getting-started/selecting-and-using-a-protocol.

- List available modules for the specified protocol:

```
nxc {{smb|ssh|ldap|ftp|wmi|winrm|rdp|vnc|mssql}} {{[-L|--list-modules]}}
```

- List the options available for the specified module:

```
nxc {{smb|ssh|ldap|ftp|wmi|winrm|rdp|vnc|mssql}} {{[-M|--module]}} {{module_name}} --options
```

- Specify an [o]ption for a module:

```
nxc {{smb|ssh|ldap|ftp|wmi|winrm|rdp|vnc|mssql}} {{[-M|--module]}} {{module_name}} -o {{OPTION_NAME}}={{option_value}}
```

- View the options available for the specified protocol:

```
nxc {{smb|ssh|ldap|ftp|wmi|winrm|rdp|vnc|mssql}} {{[-h|--help]}}
```

nxcdb

Interact with the NetExec database.

More information: https://www.netexec.wiki/getting-started/database-general-usage.

- Enter an interactive database session:

```
nxcdb
```

- Display the currently active workspace:

```
nxcdb --get-workspace
```

- Create a new workspace:

```
nxcdb --create-workspace {{workspace_name}}
```

- Activate the specified workspace:

```
nxcdb --set-workspace {{workspace_name}}
```

nyxt

A keyboard-driven web browser for power users.

More information: https://nyxt.atlas.engineer.

- List all profiles:

```
nyxt --list-data-profiles
```

- Set the `init.lisp` file path:

```
nyxt --init {{path/to/file}}
```

- Change the path to the auto-config file:

```
nyxt --auto-config {{path/to/file}}
```

- Print system information:

```
nyxt --system-information
```

o

Orbiton, a simple configuration-free text editor.

More information: https://github.com/xyproto/orbiton.

- Open a file in editor:

```
o {{path/to/file}}
```

- Open a file as read-only:

```
o {{[-m|-monitor]}} {{path/to/file}}
```

- Save the file:

```
<Ctrl s>
```

- Quit Orbiton:

```
<Ctrl q>
```

- Display help:

```
o {{[-h|--help]}}
```

oathtool

OATH one-time password tool.

More information: https://www.nongnu.org/oath-toolkit/oathtool.1.html.

- Generate TOTP token (behaves like Google Authenticator):

```
oathtool --totp {{[-b|--base32]}} "{{secret}}"
```

- Generate a TOTP token for a specific time:

```
oathtool --totp {{[-N|--now]}} "{{2004-02-29 16:21:42}}" {{[-b|--base32]}} "{{secret}}"
```

- Validate a TOTP token:

```
oathtool --totp {{[-b|--base32]}} "{{secret}}" "{{token}}"
```

oauth2c

Interact with OAuth 2.0 authorization servers.

More information: https://github.com/cloudentity/oauth2c#usage.

- Fetch an access token using client credentials:

```
oauth2c {{issuer_url}} --client-id {{client_id}} --client-secret {{client_secret}}
```

- Fetch a token using authorization code flow:

```
oauth2c {{issuer_url}} --client-id {{client_id}} --response-types code
```

- Fetch a token using authorization code with PKCE:

```
oauth2c {{issuer_url}} --client-id {{client_id}} --pkce
```

- Fetch a token using password credentials:

```
oauth2c {{issuer_url}} --client-id {{client_id}} --username {{username}} --password {{password}}
```

- Refresh an existing access token:

```
oauth2c {{issuer_url}} --client-id {{client_id}} --refresh-token {{refresh_token}}
```

- Fetch a token with specific scopes:

```
oauth2c {{issuer_url}} --client-id {{client_id}} --scopes {{scope1,scope2}}
```

- Use device authorization flow:

```
oauth2c {{issuer_url}} --client-id {{client_id}} --grant-type device_code
```

- Run in silent mode without browser:

```
oauth2c {{issuer_url}} --client-id {{client_id}} {{[-s|--silent]}} --no-browser
```

objdump

View information about object files.

More information: https://manned.org/objdump.

- Display the file header information:

```
objdump {{[-f|--file-headers]}} {{path/to/binary}}
```

- Display all header information:

```
objdump {{[-x|--all-headers]}} {{path/to/binary}}
```

- Display the disassembled output of executable sections:

```
objdump {{[-d|--disassemble]}} {{path/to/binary}}
```

- Display the disassembled executable sections in Intel syntax:

```
objdump {{[-M|--disassembler-options]}} intel {{[-d|--
disassemble]}} {{path/to/binary}}
```

- Display the symbol table:

```
objdump {{[-t|--syms]}} {{path/to/binary}}
```

- Display a complete binary hex dump of all sections:

```
objdump {{[-s|--full-contents]}} {{path/to/binary}}
```

obs

Open Broadcaster Software.

Video recording and livestreaming program.

More information: https://obsproject.com/kb/launch-parameters.

- Launch OBS:

```
obs
```

- Launch OBS in portable mode:

```
obs {{[-p|--portable]}}
```

- Automatically start recording a video on launch:

```
obs --startrecording
```

- Automatically start the replay buffer on launch:

```
obs --startreplaybuffer
```

- Automatically start streaming on launch:

```
obs --startstreaming
```

- Minimise to the system tray on launch:

```
obs --minimize-to-tray
```

- Make the log more verbose (for debugging):

```
obs --verbose
```

oc

The OpenShift Container Platform CLI.

Allows for application and container management.

More information: https://docs.openshift.com/container-platform/latest/cli_reference/get_started_cli.html.

- Log in to the OpenShift Container Platform server:

```
oc login
```

- Create a new project:

```
oc new-project {{project_name}}
```

- Switch to an existing project:

```
oc project {{project_name}}
```

- Add a new application to a project:

```
oc new-app {{repo_url}} --name {{application}}
```

- Open a remote shell session to a container:

```
oc rsh {{pod_name}}
```

- List pods in a project:

```
oc get pods
```

- Log out from the current session:

```
oc logout
```

ocaml

The OCaml repl (read-evaluate-print-loop).

Interprets Ocaml commands.

More information: https://manned.org/ocaml.

- Read OCaml commands from the user and execute them:

  ```
  ocaml
  ```

- Read OCaml commands from a file and execute them:

  ```
  ocaml {{path/to/file.ml}}
  ```

- Run OCaml script with modules:

  ```
  ocaml {{module1}} {{module2}} {{path/to/file.ml}}
  ```

ocamlc

The OCaml bytecode compiler.

Produces executables runnable by the OCaml interpreter.

More information: https://manned.org/ocamlc.

- Create a binary from a source file:

```
ocamlc {{path/to/source_file.ml}}
```

- Create a named binary from a source file:

```
ocamlc -o {{path/to/binary}} {{path/to/source_file.ml}}
```

- Automatically generate a module signature (interface) file:

```
ocamlc -i {{path/to/source_file.ml}}
```

ocamlfind

The findlib package manager for OCaml.

Simplifies linking executables with external libraries.

More information: https://manned.org/ocamlfind.

- Compile a source file to a native binary and link with packages:

```
ocamlfind ocamlopt -package {{package1,package2,...}} -
linkpkg -o {{path/to/executable}} {{path/to/source.ml}}
```

- Compile a source file to a bytecode binary and link with packages:

```
ocamlfind ocamlc -package {{package1,package2,...}} -linkpkg
-o {{path/to/executable}} {{path/to/source.ml}}
```

- Cross-compile for a different platform:

```
ocamlfind -toolchain {{cross-toolchain}} ocamlopt -o {{path/
to/executable}} {{path/to/source.ml}}
```

ocamlopt

The OCaml native code compiler.

Produces native executables, e.g. ELF on Linux.

More information: https://manned.org/ocamlopt.

- Compile a source file:

```
ocamlopt -o {{path/to/binary}} {{path/to/source_file.ml}}
```

- Compile with debugging enabled:

```
ocamlopt -g -o {{path/to/binary}} {{path/to/source_file.ml}}
```

ocrmypdf

Generate a searchable PDF or PDF/A from a scanned PDF or an image of text.

More information: https://ocrmypdf.readthedocs.io/en/latest/cookbook.html.

- Create a new searchable PDF/A file from a scanned PDF or image file:

```
ocrmypdf {{path/to/input}} {{path/to/output.pdf}}
```

- Skip pages of a mixed-format input PDF file that already contain text:

```
ocrmypdf --skip-text {{path/to/input.pdf}} {{path/to/
output.pdf}}
```

- Clean, de-skew, and rotate pages of a poor scan:

```
ocrmypdf --clean --deskew --rotate-pages {{path/to/
input.pdf}} {{path/to/output.pdf}}
```

- Perform lossy optimization on a PDF without performing any OCR:

```
ocrmypdf --tesseract-timeout 0 --optimize 2 --skip-text
{{path/to/input.pdf}} {{path/to/output.pdf}}
```

- Set the metadata of a searchable PDF file:

```
ocrmypdf --title "{{title}}" --author "{{author}}" --subject
"{{subject}}" --keywords "{{keyword; key
phrase; ...}}" {{path/to/input.pdf}} {{path/to/output.pdf}}
```

- Display help:

```
ocrmypdf --help
```

octave

A programming language for scientific computing.

More information: https://docs.octave.org/latest/Command-Line-Options.html.

- Start an interactive session:

```
octave
```

- Execute a specific script file:

```
octave {{path/to/script.m}}
```

- Execute a script file with specific arguments:

```
octave {{path/to/script.m}} {{argument1 argument2 ...}}
```

- Start an interactive session with a GUI:

```
octave --gui
```

- Display help:

```
octave --help
```

- Display version:

```
octave --version
```

octez-client

Interact with the Tezos blockchain.

More information: https://tezos.gitlab.io/introduction/howtouse.html#client.

- Configure the client with a connection to a Tezos RPC node such as https://rpc.ghostnet.teztnets.com:

```
octez-client -E {{endpoint}} config update
```

- Create an account and assign a local alias to it:

```
octez-client gen keys {{alias}}
```

- Get the balance of an account by alias or address:

```
octez-client get balance for {{alias_or_address}}
```

- Transfer tez to a different account:

```
octez-client transfer {{5}} from {{alias|address}} to
{{alias|address}}
```

- Originate (deploy) a smart contract, assign it a local alias, and set its initial storage as a Michelson-encoded value:

```
octez-client originate contract {{alias}} transferring {{0}}
from {{alias|address}} running {{path/to/source_file.tz}} --
init "{{initial_storage}}" --burn_cap {{1}}
```

- Call a smart contract by its alias or address and pass a Michelson-encoded parameter:

```
octez-client transfer {{0}} from {{alias|address}} to
{{contract}} --entrypoint "{{entrypoint}}" --arg
"{{parameter}}" --burn-cap {{1}}
```

- Display help:

```
octez-client man
```

od

Display file contents in octal, decimal or hexadecimal format.

Optionally display the byte offsets and/or printable representation for each line.

See also: **`hexyl`**, **`xxd`**, **`hexdump`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/od-invocation.html.

- Display file using default settings: octal format, 8 bytes per line, byte offsets in octal, and duplicate lines replaced with *:

```
od {{path/to/file}}
```

- Display file in verbose mode, i.e. without replacing duplicate lines with *:

```
od {{[-v|--output-duplicates]}} {{path/to/file}}
```

- Display file in hexadecimal format (2-byte units), with byte offsets in decimal format:

```
od {{[-t|--format]}} {{x}} {{[-A|--address-radix]}} {{d}} {{[-v|--output-duplicates]}} {{path/to/file}}
```

- Display file in hexadecimal format (1-byte units), and 4 bytes per line:

```
od {{[-t|--format]}} {{x1}} {{[-w|--width=]}}4 {{[-v|--output-duplicates]}} {{path/to/file}}
```

- Display file in hexadecimal format along with its character representation, and do not print byte offsets:

```
od {{[-t|--format]}} {{xz}} {{[-A|--address-radix]}} {{n}} {{[-v|--output-duplicates]}} {{path/to/file}}
```

- Read only 100 bytes of a file starting from the 500th byte:

```
od {{[-N|--read-bytes]}} 100 {{[-j|--skip-bytes]}} 500 {{[-v|--output-duplicates]}} {{path/to/file}}
```

odps auth

User authorities in ODPS (Open Data Processing Service).

See also: **odps**.

More information: https://www.alibabacloud.com/help/doc-detail/27971.htm.

- Add a user to the current project:

```
add user {{username}};
```

- Grant a set of authorities to a user:

```
grant {{action_list}} on {{object_type}} {{object_name}} to
user {{username}};
```

- Show authorities of a user:

```
show grants for {{username}};
```

- Create a user role:

```
create role {{role_name}};
```

- Grant a set of authorities to a role:

```
grant {{action_list}} on {{object_type}} {{object_name}} to
role {{role_name}};
```

- Describe authorities of a role:

```
desc role {{role_name}};
```

- Grant a role to a user:

```
grant {{role_name}} to {{username}};
```

odps func

Manage functions in ODPS (Open Data Processing Service).

See also: **odps**.

More information: https://www.alibabacloud.com/help/doc-detail/27971.htm.

- Show functions in the current project:

```
list functions;
```

- Create a Java function using a `.jar` resource:

```
create function {{func_name}} as {{path.to.package.Func}}
using '{{package.jar}}';
```

- Create a Python function using a `.py` resource:

```
create function {{func_name}} as {{script.Func}} using
'{{script.py}}';
```

- Delete a function:

```
drop function {{func_name}};
```

odps inst

Manage instances in ODPS (Open Data Processing Service).

See also: **odps**.

More information: https://www.alibabacloud.com/help/doc-detail/27971.htm.

- Show instances created by current user:

```
show instances;
```

- Describe the details of an instance:

```
desc instance {{instance_id}};
```

- Check the status of an instance:

```
status {{instance_id}};
```

- Wait on the termination of an instance, printing log and progress information until then:

```
wait {{instance_id}};
```

- Kill an instance:

```
kill {{instance_id}};
```

odps resource

Manage resources in ODPS (Open Data Processing Service).

See also: **odps**.

More information: https://www.alibabacloud.com/help/doc-detail/27971.htm.

- Show resources in the current project:

```
list resources;
```

- Add file resource:

```
add file {{filename}} as {{alias}};
```

- Add archive resource:

```
add archive {{archive.tar.gz}} as {{alias}};
```

- Add .jar resource:

```
add jar {{package.jar}};
```

- Add .py resource:

```
add py {{script.py}};
```

- Delete resource:

```
drop resource {{resource_name}};
```

odps table

Create and modify tables in ODPS (Open Data Processing Service).

See also: **odps**.

More information: https://www.alibabacloud.com/help/doc-detail/27971.htm.

- Create a table with partition and lifecycle:

```
create table {{table_name}} ({{col}} {{type}}) partitioned by
({{col}} {{type}}) lifecycle {{days}};
```

- Create a table based on the definition of another table:

```
create table {{table_name}} like {{another_table}};
```

- Add partition to a table:

```
alter table {{table_name}} add partition
({{partition_spec}});
```

- Delete partition from a table:

```
alter table {{table_name}} drop partition
({{partition_spec}});
```

- Delete table:

```
drop table {{table_name}};
```

odps tunnel

Data tunnel in ODPS (Open Data Processing Service).

See also: **odps**.

More information: https://www.alibabacloud.com/help/doc-detail/27971.htm.

- Download table to local file:

```
tunnel download {{table_name}} {{path/to/file}};
```

- Upload local file to a table partition:

```
tunnel upload {{path/to/file}} {{table_name}}/
{{partition_spec}};
```

- Upload table specifying field and record delimiters:

```
tunnel upload {{path/to/file}} {{table_name}} -fd
{{field_delim}} -rd {{record_delim}};
```

- Upload table using multiple threads:

```
tunnel upload {{path/to/file}} {{table_name}} -threads
{{num}};
```

odps

Aliyun ODPS (Open Data Processing Service) command-line tool.

Some subcommands such as **`inst`** have their own usage documentation.

More information: https://www.alibabacloud.com/help/doc-detail/27971.htm.

- Start the command-line with a custom configuration file:

```
odpscmd --config={{odps_config.ini}}
```

- Switch current project:

```
use {{project_name}};
```

- Show tables in the current project:

```
show tables;
```

- Describe a table:

```
desc {{table_name}};
```

- Show table partitions:

```
show partitions {{table_name}};
```

- Describe a partition:

```
desc {{table_name}} partition ({{partition_spec}});
```

offlineimap

Synchronize a remote IMAP server with local Maildir folders.

More information: https://www.offlineimap.org.

- Synchronize once, without enabling autorefresh:

```
offlineimap -o
```

- Synchronize a specific account:

```
offlineimap -a {{account}}
```

- Synchronize a specific folder:

```
offlineimap -f {{folder}}
```

ogr2ogr

Convert geospatial vector data between file formats.

More information: https://gdal.org/programs/ogr2ogr.html.

- Convert a Shapefile into a GeoPackage:

```
ogr2ogr -f GPKG {{path/to/output.gpkg}} {{path/to/input.shp}}
```

- Reduce a GeoJSON to features matching a condition:

```
ogr2ogr -where '{{myProperty > 42}}' -f {{GeoJSON}} {{path/to/output.geojson}} {{path/to/input.geojson}}
```

- Change coordinate reference system of a GeoPackage from `EPSG:4326` to `EPSG:3857`:

```
ogr2ogr -s_srs {{EPSG:4326}} -t_srs {{EPSG:3857}} -f GPKG {{path/to/output.gpkg}} {{path/to/input.gpkg}}
```

- Convert a CSV file into a GeoPackage, specifying the names of the coordinate columns and assigning a coordinate reference system:

```
ogr2ogr -f GPKG {{path/to/output.gpkg}} {{path/to/input.csv}} -oo X_POSSIBLE_NAMES={{longitude}} -oo Y_POSSIBLE_NAMES={{latitude}} -a_srs {{EPSG:4326}}
```

- Load a GeoPackage into a PostGIS database:

```
ogr2ogr -f PostgreSQL PG:dbname="{{database_name}}" {{path/to/input.gpkg}}
```

- Clip layers of a GeoPackage file to the given bounding box:

```
ogr2ogr -spat {{min_x}} {{min_y}} {{max_x}} {{max_y}} -f GPKG {{path/to/output.gpkg}} {{path/to/input.gpkg}}
```

ogrinfo

List information about an OGR-supported data source.

More information: https://gdal.org/programs/ogrinfo.html.

- List supported formats:

```
ogrinfo --formats
```

- List layers of a data source:

```
ogrinfo {{path/to/input.gpkg}}
```

- Get detailed information about a specific layer of a data source:

```
ogrinfo {{path/to/input.gpkg}} {{layer_name}}
```

- Show summary information about a specific layer of a data source:

```
ogrinfo -so {{path/to/input.gpkg}} {{layer_name}}
```

- Show summary of all layers of the data source:

```
ogrinfo -so -al {{path/to/input.gpkg}}
```

- Show detailed information of features matching a condition:

```
ogrinfo -where '{{attribute_name > 42}}' {{path/to/
input.gpkg}} {{layer_name}}
```

- Update a layer in the data source with SQL:

```
ogrinfo {{path/to/input.geojson}} -dialect SQLite -sql
"{{UPDATE input SET attribute_name = 'foo'}}"
```

ogrmerge.py

Merge several vector datasets into a single one.

More information: https://gdal.org/programs/ogrmerge.html.

- Create a GeoPackage with a layer for each input Shapefile:

```
ogrmerge.py -f {{GPKG}} -o {{path/to/output.gpkg}} {{path/to/
input1.shp path/to/input2.shp ...}}
```

- Create a virtual datasource (VRT) with a layer for each input GeoJSON:

```
ogrmerge.py -f {{VRT}} -o {{path/to/output.vrt}} {{path/to/
input1.geojson path/to/input2.geojson ...}}
```

- Concatenate two vector datasets and store source name of dataset in attribute 'source_name':

```
ogrmerge.py -single -f {{GeoJSON}} -o {{path/to/
output.geojson}} -src_layer_field_name country
{{source_name}} {{path/to/input1.shp path/to/input2.shp ...}}
```

oh-my-posh

Cross platform prompt engine for any shell.

More information: https://ohmyposh.dev/docs/installation/customize.

- Display current shell name:

```
oh-my-posh get shell
```

- Initialize for Bash:

```
eval "$(oh-my-posh init bash --config {{path/to/theme}})"
```

- Initialize for PowerShell:

```
oh-my-posh init pwsh --config {{path/to/theme}} | Invoke-Expression
```

- Initialize for Zsh:

```
eval "$(oh-my-posh init zsh --config {{path/to/theme}})"
```

- Enable live reloading:

```
oh-my-posh enable reload
```

- Disable live reloading:

```
oh-my-posh disable reload
```

- Print debug information for troubleshooting:

```
oh-my-posh debug
```

- Update to latest version:

```
oh-my-posh upgrade
```

ohdear-cli

An unofficial Oh Dear CLI written with Laravel Zero.

More information: https://github.com/nunomaduro/ohdear-cli.

- Display details about the currently authenticated user:

  ```
  ohdear-cli me
  ```

- Add a new site to Oh Dear:

  ```
  ohdear-cli sites:add {{url}}
  ```

- Display a list of sites and their current status:

  ```
  ohdear-cli sites:list
  ```

- Display details about a specific site:

  ```
  ohdear-cli sites:show {{site_id}}
  ```

okular

View documents.

More information: https://docs.kde.org/stable5/en/okular/okular/command-line-options.html.

- Launch document viewer:

```
okular
```

- Open specific documents:

```
okular {{path/to/file1 path/to/file2 ...}}
```

- Open a document at a specific page:

```
okular {{[-p|--page]}} {{page_number}} {{path/to/file}}
```

- Open a specific document in presentation mode:

```
okular --presentation {{path/to/file}}
```

- Open a specific document and start a print dialog:

```
okular --print {{path/to/file}}
```

- Open a document and search for a specific string:

```
okular --find {{search_string}} {{path/to/file}}
```

olevba

Parse OLE and OpenXML files (e.g., DOC, XLS, PPT, etc.) to extract VBA macros, deobfuscate, and analyze malicious code.

Part of the **`python-oletools`** suite.

More information: https://github.com/decalage2/oletools.

- Analyze a file, showing both macro code and analysis results:

```
olevba {{path/to/file}}
```

- Recursively analyze all supported files in a directory:

```
olevba -r {{path/to/directory}}
```

- Provide a password for encrypted Microsoft Office files (may be repeated):

```
olevba {{[-p|--password]}} {{password}} {{path/to/
encrypted_file}}
```

- Display only analysis results, without showing macro source code:

```
olevba {{[-a|--analysis]}} {{path/to/file}}
```

- Display only macro source code:

```
olevba {{[-c|--code]}} {{path/to/file}}
```

- Show obfuscated strings and their decoded content:

```
olevba --decode {{path/to/file}}
```

ollama

A large language model runner.

For a list of available models, see https://ollama.com/library.

More information: https://github.com/ollama/ollama.

- Start the daemon required to run other commands:

```
ollama serve
```

- Run a model and chat with it:

```
ollama run {{model}}
```

- Run a model with a single prompt:

```
ollama run {{model}} {{prompt}}
```

- List downloaded models:

```
ollama list
```

- Pull a specific model:

```
ollama pull {{model}}
```

- List running models:

```
ollama ps
```

- Delete a model:

```
ollama rm {{model}}
```

- Create a model from a `Modelfile`:

```
ollama create {{new_model_name}} {{[-f|--file]}} {{path/to/Modelfile}}
```

omf

Oh My Fish, the Fishshell Framework.

Install packages to extend and modify the fish shell.

More information: https://github.com/oh-my-fish/oh-my-fish.

- Install one or more packages:

```
omf install {{name}}
```

- List installed packages:

```
omf list
```

- List available themes:

```
omf theme
```

- Apply a theme:

```
omf theme {{name}}
```

- Remove a theme or package:

```
omf remove {{name}}
```

- Uninstall Oh My Fish:

```
omf destroy
```

omz

Oh My Zsh command-line tool.

More information: https://github.com/ohmyzsh/ohmyzsh.

- Update Oh My Zsh:

```
omz update
```

- Print the changes from the latest update of Oh My Zsh:

```
omz changelog
```

- Restart the current Zsh session and Oh My Zsh:

```
omz reload
```

- List all available plugins:

```
omz plugin list
```

- List all enabled plugins:

```
omz plugin list --enabled
```

- Enable/Disable an Oh My Zsh plugin:

```
omz plugin {{enable|disable}} {{plugin}}
```

- List all available themes:

```
omz theme list
```

- Set an Oh My Zsh theme in `~/.zshrc`:

```
omz theme set {{theme}}
```

on_ac_power

A simple utility which tests if a computer is running on line power.

Returns **0** if yes, and **1** if no.

More information: https://manned.org/on_ac_power.

- Test if a computer is running on line power:

```
on_ac_power
```

onefetch

Display project information and code statistics for a local Git repository.

More information: https://github.com/o2sh/onefetch/wiki/command-line-options.

- Display statistics for the Git repository in the current working directory:

```
onefetch
```

- Display statistics for the Git repository in the specified directory:

```
onefetch {{path/to/directory}}
```

- Ignore commits made by bots:

```
onefetch --no-bots
```

- Ignore merge commits:

```
onefetch --no-merges
```

- Don't print the ASCII art of the language logo:

```
onefetch --no-art
```

- Show n authors, languages, or file churns (default: 3, 6, and 3 respectively):

```
onefetch --number-of-{{authors|languages|file-churns}} {{n}}
```

- Ignore the specified files and directories:

```
onefetch {{[-e|--exclude]}} {{path/to/file_or_directory|
regex}}
```

- Only detect languages from the specified categories (default: programming and markup):

```
onefetch {{[-T|--type]}} {{programming|markup|prose|data}}
```

oneliner

Turn plain English into shell commands using OpenAI, Claude, or local LLMs, designed to teach, not replace your knowledge.

More information: https://github.com/dorochadev/oneliner#-usage-flags.

- Generate a shell command from plain English:

```
oneliner "{{find all jpg files larger than 10MB}}"
```

- Explain what a command does:

```
oneliner {{[-e|--explain]}} "{{delete node_modules recursively}}"
```

- Copy a generated command to the clipboard:

```
oneliner {{[-c|--clipboard]}} "{{compress all pdfs}}"
```

- Show a detailed, educational breakdown of a command:

```
oneliner {{[-b|--breakdown]}} "{{list all active network connections}}"
```

- Execute a generated command (use with caution):

```
oneliner {{[-r|--run]}} "{{list files larger than 1GB}}"
```

- Interactively confirm before executing a generated command:

```
oneliner {{[-i|--interactive]}} "{{delete temporary files in /tmp}}"
```

onionsearch

Scrape URLs on different **.onion** search engines.

Note: **onionsearch** requires a Tor proxy running on **localhost:9050**; a Tor enabled browser is needed to visit the **.onion** websites.

More information: https://github.com/megadose/OnionSearch.

- Request results from all the search engines:

```
onionsearch "{{string}}"
```

- Request search results from specific search engines:

```
onionsearch "{{string}}" --engines {{tor66 deeplink
phobos ...}}
```

- Exclude certain search engines when searching:

```
onionsearch "{{string}}" --exclude {{candle ahmia ...}}
```

- Limit the number of pages to load per engine:

```
onionsearch "{{stuxnet}}" --engines {{tor66 deeplink
phobos ...}} --limit {{3}}
```

- List all supported search engines:

```
onionsearch --help | grep {{[-A|--after-context]}} 1 {{[-i|--
ignore-case]}} "supported engines"
```

ooniprobe

Open Observatory of Network Interference (OONI).

Test the blocking of websites and apps. Measure the speed and performance of your network.

More information: https://ooni.org/support/ooni-probe-cli/.

- List all tests performed:

  ```
  ooniprobe list
  ```

- Show information about a specific test:

  ```
  ooniprobe list {{7}}
  ```

- Run all available tests:

  ```
  ooniprobe run all
  ```

- Perform a specific test:

  ```
  ooniprobe run {{performance}}
  ```

- Check the availability of a specific website:

  ```
  ooniprobe run websites --input {{https://ooni.org/}}
  ```

- Check the availability of all websites listed in a file:

  ```
  ooniprobe run websites --input-file {{path/to/my-websites.txt}}
  ```

- Display detailed information about a test in JSON format:

  ```
  ooniprobe show {{9}}
  ```

op

Official CLI for 1Password's desktop app.

More information: https://developer.1password.com/docs/cli/reference.

- Sign in to a 1Password account:

```
op signin
```

- List all vaults:

```
op vault list
```

- Print item details in JSON format:

```
op item get {{item_name}} --format json
```

- Create a new item with a category in the default vault:

```
op item create --category {{category_name}}
```

- Print a referenced secret to `stdout`:

```
op read {{secret_reference}}
```

- Pass secret references from exported environment variables to a command:

```
op run -- {{command}}
```

- Pass secret references from an environment file to a command:

```
op run --env-file {{path/to/env_file.env}} -- {{command}}
```

- Read secret references from a file and save plaintext secrets to a file:

```
op inject --in-file {{path/to/input_file}} --out-file {{path/to/output_file}}
```

opam

OCaml Package Manager.

Manage OCaml compilers, tools and libraries.

More information: https://opam.ocaml.org/doc/Usage.html.

- Initialize opam for first use:

```
opam init
```

- Search for packages:

```
opam search {{query}}
```

- Install a package and all of its dependencies:

```
opam install {{package}}
```

- Display detailed information about a package:

```
opam show {{package}}
```

- List all installed packages:

```
opam list
```

- Update the local package database:

```
opam update
```

- Upgrade all installed packages:

```
opam upgrade
```

- Display help:

```
opam help
```

open

Open files, directories, and URIs with default applications.

This command is available through fish on operating systems without the built-in **open** command (e.g. Haiku and macOS).

More information: https://fishshell.com/docs/current/cmds/open.html.

- Open a file with the associated application:

```
open {{path/to/file.ext}}
```

- Open all the files of a given extension in the current directory with the associated application:

```
open {{*.ext}}
```

- Open a directory using the default file manager:

```
open {{path/to/directory}}
```

- Open a website using the default web browser:

```
open {{https://example.com}}
```

- Open a specific URI using the default application that can handle it:

```
open {{tel:123}}
```

open

open can refer to multiple commands with the same name.

- View documentation for the command available in macOS:

```
tldr open {{[-p|--platform]}} osx
```

- View documentation for the command available through fish:

```
tldr open.fish
```

openai

CLI tool providing access to the OpenAI API.

More information: https://github.com/openai/openai-python.

- List models:

```
openai api models.list
```

- Create a completion:

```
openai api completions.create --model {{ada}} --prompt
"{{Hello world}}"
```

- Create a chat completion:

```
openai api chat_completions.create --model {{gpt-3.5-turbo}}
--message {{user "Hello world"}}
```

- Generate images via DALL·E API:

```
openai api image.create --prompt "{{two dogs playing chess,
cartoon}}" --num-images {{1}}
```

opencode

An AI coding agent.

More information: https://opencode.ai/docs/cli/.

- Start the interactive TUI:

```
opencode
```

- Run opencode in non-interactive mode by passing a prompt directly:

```
opencode run "{{message}}"
```

- Use a specific model and agent:

```
opencode run {{[-m|--model]}} {{provider}}/{{model}} --agent {{agent_name}} "{{message}}"
```

- List all available models from configured providers:

```
opencode models
```

- Manage credentials and login for providers:

```
opencode auth login
```

- Start a headless opencode server for API access:

```
opencode serve {{[-h|--hostname]}} {{hostname}} {{[-p|--port]}} {{port}}
```

- Manage AI assistant agents for OpenCode:

```
opencode agent {{command}}
```

- Create a new agent with custom configuration:

```
opencode agent create
```

openconnect

A VPN client, for Cisco AnyConnect VPNs and others.

More information: https://www.infradead.org/openconnect/manual.html.

- Connect to a server:

```
openconnect {{vpn.example.org}}
```

- Connect to a server, forking into the background:

```
openconnect --background {{vpn.example.org}}
```

- Terminate the connection that is running in the background:

```
killall -SIGINT openconnect
```

- Connect to a server, reading options from a configuration file:

```
openconnect --config={{path/to/file}} {{vpn.example.org}}
```

- Connect to a server and authenticate with a specific SSL client certificate:

```
openconnect --certificate={{path/to/file}}
{{vpn.example.org}}
```

openocd

Debug and program embedded systems with OpenOCD.

More information: https://manned.org/openocd.

- Attach OpenOCD session to a board with a configuration file:

```
openocd {{[-f|--file]}} {{config_file.cfg}}
```

- Attach OpenOCD session to a board with multiple configuration files:

```
openocd {{[-f|--file]}} {{config_file1.cfg}} {{[-f|--file]}} {{config_file2.cfg}}
```

- Attach OpenOCD session to a board with configuration files and a list of commands to be executed on server startup:

```
openocd {{[-f|--file]}} {{config_file.cfg}} {{[-c|--command]}} "{{command}}"
```

- Use configuration files in the specified path:

```
openocd {{[-s|--search]}} {{path/to/search}} {{[-f|--file]}} {{config_file.cfg}}
```

- After OpenOCD startup, connect GDB to OpenOCD default port 3333:

```
target extended-remote localhost
```

- List site-wide script library:

```
ls /usr/local/share/openocd/scripts
```

openrgb

Control RGB lighting.

More information: https://gitlab.com/OpenRGBDevelopers/OpenRGB-Wiki/-/blob/stable/User-Documentation/Using-OpenRGB.md.

- Start the OpenRGB GUI:

```
openrgb
```

- List devices supported by OpenRGB:

```
openrgb --noautoconnect {{[-l|--list-devices]}}
```

- Set the mode and color of a device:

```
openrgb {{[-m|--mode]}} {{off|static|breathing|rainbow|
flashing|...}} {{[-c|--color]}} {{random|red|00AAFF|...}}
```

- Display help:

```
openrgb {{[-h|--help]}}
```

openscad

Software for creating solid 3D CAD objects.

More information: https://manned.org/openscad.

- Open a file:

```
openscad {{path/to/button.scad}}
```

- Convert a file to STL:

```
openscad -o {{path/to/button.stl}} {{path/to/button.scad}}
```

- Render a file to PNG in a specific colorscheme:

```
openscad -o {{path/to/button.png}} --colorscheme {{Sunset}}
{{path/to/button.scad}}
```

openssl dgst

OpenSSL command to generate digest values and perform signature operations.

More information: https://www.openssl.org/docs/manmaster/man1/openssl-dgst.html.

- Calculate the SHA256 digest for a file, saving the result to a specific file:

```
openssl dgst -sha256 -binary -out {{output_file}} {{input_file}}
```

- Sign a file using an RSA key, saving the result to a specific file:

```
openssl dgst -sign {{private_key_file}} -sha256 -sigopt rsa_padding_mode:pss -out {{output_file}} {{input_file}}
```

- Verify an RSA signature:

```
openssl dgst -verify {{public_key_file}} -signature {{signature_file}} -sigopt rsa_padding_mode:pss {{signature_message_file}}
```

- Sign a file using and ECDSA key:

```
openssl dgst -sign {{private_key_file}} -sha256 -out {{output_file}} {{input_file}}
```

- Verify an ECDSA signature:

```
openssl dgst -verify {{public_key_file}} -signature {{signature_file}} {{signature_message_file}}
```

openssl genpkey

OpenSSL command to generate asymmetric key pairs.

More information: https://www.openssl.org/docs/manmaster/man1/openssl-genpkey.html.

- Generate an RSA private key of 2048 bits, saving it to a specific file:

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:{{2048}} -out {{filename.key}}
```

- Generate an elliptic curve private key using the curve `prime256v1`, saving it to a specific file:

```
openssl genpkey -algorithm EC -pkeyopt ec_paramgen_curve:{{prime256v1}} -out {{filename.key}}
```

- Generate an `ED25519` elliptic curve private key, saving it to a specific file:

```
openssl genpkey -algorithm {{ED25519}} -out {{filename.key}}
```

openssl genrsa

OpenSSL command to generate RSA private keys.

More information: https://www.openssl.org/docs/manmaster/man1/openssl-genrsa.html.

- Generate an RSA private key of 2048 bits to `stdout`:

```
openssl genrsa
```

- Save an RSA private key of an arbitrary number of bits to the output file:

```
openssl genrsa -out {{output_file.key}} {{1234}}
```

- Generate an RSA private key and encrypt it with AES256 (you will be prompted for a passphrase):

```
openssl genrsa {{-aes256}}
```

openssl prime

OpenSSL command to compute prime numbers.

More information: https://www.openssl.org/docs/manmaster/man1/openssl-prime.html.

- Generate a 2048bit prime number and display it in hexadecimal:

```
openssl prime -generate -bits 2048 -hex
```

- Check if a given number is prime:

```
openssl prime {{number}}
```

openssl req

OpenSSL command to manage PKCS#10 Certificate Signing Requests.

More information: https://www.openssl.org/docs/manmaster/man1/openssl-req.html.

- Generate a certificate signing request to be sent to a certificate authority:

```
openssl req -new -sha256 -key {{filename.key}} -out
{{filename.csr}}
```

- Generate a self-signed certificate and a corresponding key-pair, storing both in a file:

```
openssl req -new -x509 -newkey {{rsa}}:{{4096}} -keyout
{{filename.key}} -out {{filename.cert}} -subj "{{/C=XX/
CN=foobar}}" -days {{365}}
```

openssl s_client

OpenSSL command to create TLS client connections.

More information: https://www.openssl.org/docs/manmaster/man1/openssl-s_client.html.

- Display the start and expiry dates for a domain's certificate:

```
openssl s_client -connect {{host}}:{{port}} 2>/dev/null |
openssl x509 -noout -dates
```

- Display the certificate presented by an SSL/TLS server:

```
openssl < /dev/null s_client -connect {{host}}:{{port}}
```

- Set the Server Name Indicator (SNI) when connecting to the SSL/TLS server:

```
openssl s_client -connect {{host}}:{{port}} -servername
{{hostname}}
```

- Display the complete certificate chain of an HTTPS server:

```
openssl < /dev/null s_client -connect {{host}}:443 -showcerts
```

openssl ts

OpenSSL command to generate and verify timestamps.

More information: https://www.openssl.org/docs/manmaster/man1/openssl-ts.html.

- Generate a SHA-512 timestamp request of a specific file and output to `file.tsq`:

```
openssl ts -query -data {{path/to/file}} -sha512 -out {{path/to/file.tsq}}
```

- Check the date and metadata of a specific timestamp response file:

```
openssl ts -reply -in {{path/to/file.tsr}} -text
```

- Verify a timestamp request file and a timestamp response file from the server with an SSL certificate file:

```
openssl ts -verify -in {{path/to/file.tsr}} -queryfile {{path/to/file.tsq}} -partial_chain -CAfile {{path/to/cert.pem}}
```

- Create a timestamp response for request using key and signing certificate and output it to `file.tsr`:

```
openssl ts -reply -queryfile {{path/to/file.tsq}} -inkey {{path/to/tsakey.pem}} -signer tsacert.pem -out {{path/to/file.tsr}}
```

openssl x509

OpenSSL command to manage X.509 certificates.

More information: https://www.openssl.org/docs/manmaster/man1/openssl-x509.html.

- Display certificate information:

```
openssl x509 -in {{filename.crt}} -noout -text
```

- Display a certificate's expiration date:

```
openssl x509 -enddate -noout -in {{filename.pem}}
```

- Convert a certificate between binary DER encoding and textual PEM encoding:

```
openssl x509 -inform {{der}} -outform {{pem}} -in
{{original_certificate_file}} -out
{{converted_certificate_file}}
```

- Store a certificate's public key in a file:

```
openssl x509 -in {{certificate_file}} -noout -pubkey -out
{{output_file}}
```

openssl

OpenSSL cryptographic toolkit.

Some subcommands such as **req** have their own usage documentation.

More information: https://www.openssl.org.

- Display help:

```
openssl help
```

- Display help for a specific subcommand:

```
openssl help {{x509}}
```

- Display version:

```
openssl version
```

openstack flavor

Manage OpenStack instance flavors (virtual hardware templates).

More information: https://docs.openstack.org/python-openstackclient/latest/cli/command-objects/flavor.html.

- List all flavors:

```
openstack flavor list
```

- Show details of a flavor:

```
openstack flavor show {{flavor_id_or_name}}
```

- Create a new flavor with 2 vCPUs, 4GB RAM, and 20GB disk:

```
openstack flavor create {{flavor_name}} --vcpus 2 --ram 4096
--disk 20
```

- Delete a flavor:

```
openstack flavor delete {{flavor_id_or_name}}
```

- Create a flavor with 10GB ephemeral disk and 512MB swap space:

```
openstack flavor create {{flavor_name}} --vcpus 4 --ram 8192
--disk 40 --ephemeral 10 --swap 512
```

openstack help

Display help information about the openstackclient cli.

More information: https://docs.openstack.org/python-openstackclient/latest/cli/man/openstack.html.

- Display a description of a particular command:

```
openstack help {{command_name}}
```

- Get help for Identity v3:

```
openstack --os-identity-api-version 3 --help
```

- Display help:

```
openstack --help
```

openstack image

OpenStack Image service, aka OpenStack Glance, allows users to upload and discover data assets meant to be used with other services.

More information: https://docs.openstack.org/python-openstackclient/latest/cli/command-objects/image-v2.html.

- List available images:

```
openstack image list {{--private|--shared|--all}}
```

- Display image details:

```
openstack image show --human-readable {{image_id}}
```

- Create/upload an image:

```
openstack image create --file {{path/to/file}} {{--private|--shared|--all}} {{image_name}}
```

- Delete image(s):

```
openstack image delete {{image_id1 image_id2 ...}}
```

- Save an image locally:

```
openstack image save --file {{filename}} {{image_id}}
```

openstack network

Manage OpenStack network resources.

More information: https://docs.openstack.org/python-openstackclient/latest/cli/command-objects/network.html.

- List all networks:

  ```
  openstack network list
  ```

- Show details of a network:

  ```
  openstack network show {{network_id_or_name}}
  ```

- Create a new network with a given name:

  ```
  openstack network create {{network_name}}
  ```

- Delete a network:

  ```
  openstack network delete {{network_id_or_name}}
  ```

- Enable a network:

  ```
  openstack network set --enable {{network_id_or_name}}
  ```

- Disable a network:

  ```
  openstack network set --disable {{network_id_or_name}}
  ```

openstack port

Manage OpenStack network ports (virtual network interfaces).

More information: https://docs.openstack.org/python-openstackclient/latest/cli/command-objects/port.html.

- List all ports:

```
openstack port list
```

- Show detailed information about a specific port:

```
openstack port show {{port_id_or_name}}
```

- Create a port on a specific network:

```
openstack port create --network {{network_id_or_name}} {{port_name}}
```

- Create a port and assign it a fixed IP `192.168.1.50`:

```
openstack port create --network {{network_id}} --fixed-ip subnet={{subnet_id}},ip-address=192.168.1.50 {{port_name}}
```

- Delete a port:

```
openstack port delete {{port_id_or_name}}
```

openstack server

Manage OpenStack virtual machines.

OpenStack Compute service, aka OpenStack Nova, mainly hosts and manages cloud computing systems.

More information: https://docs.openstack.org/python-openstackclient/latest/cli/command-objects/server.html.

- List servers:

```
openstack server list
```

- Start server(s):

```
openstack server start {{instance_id1 instance_id2 ...}}
```

- Stop server:

```
openstack server stop {{instance_id1 instance_id2 ...}}
```

- Create new server:

```
openstack server create --image {{image_id}} --flavor
{{flavor_id}} --network {{network_id}} --wait {{server_name}}
```

- Delete server(s):

```
openstack server delete {{instance_id1 instance_id2 ...}}
```

- Migrate server to different host:

```
openstack server migrate --live {{host_hostname}} {{--shared-
migration|--block-migration}} --wait {{instance_id}}
```

- Perform a soft or hard reset to the server:

```
openstack server reboot {{--soft|--hard}} --wait
{{instance_id}}
```

openstack subnet

Manage OpenStack subnets (IP address blocks within a network).

More information: https://docs.openstack.org/python-openstackclient/latest/cli/command-objects/subnet.html.

- List all subnets:

```
openstack subnet list
```

- Show details of a specific subnet:

```
openstack subnet show {{subnet_id_or_name}}
```

- List subnets associated with a network:

```
openstack subnet list --network {{network_id_or_name}}
```

- Create a subnet with subnet range `192.168.0.0/24` in a given network:

```
openstack subnet create --network {{network_id_or_name}} --subnet-range 192.168.0.0/24 {{subnet_name}}
```

- Delete a subnet:

```
openstack subnet delete {{subnet_id_or_name}}
```

- Update a subnet with DNS `8.8.8.8` and set a new name:

```
openstack subnet set --dns-nameserver 8.8.8.8 --name {{new_subnet_name}} {{subnet_id}}
```

openstack volume

Manage OpenStack volumes.

OpenStack Block Storage service, aka OpenStack Cinder, provides volumes to Nova vm's, Ironic bare-metal hosts, containers, and others.

More information: https://docs.openstack.org/python-openstackclient/latest/cli/command-objects/volume.html.

- List volumes:

```
openstack volume list --all-projects
```

- Show volume details:

```
openstack volume show {{volume_id}}
```

- Create new volume:

```
openstack volume create --size {{size_in_GB}} --image
{{image_id}} --snapshot {{snapshot_id}} {{--bootable|--non-
bootable}} {{volume_name}}
```

- Delete volumes(s):

```
openstack volume delete {{volume_id1 volume_id2 ...}}
```

- Migrate volume to a new host:

```
openstack volume migrate --host {{host_hostname}}
{{instance_id}}
```

- Set volume properties:

```
openstack volume set --name {{volume_new_name}} --size
{{volume_new_size}} {{--attached|--detached}} {{--bootable|--
non-bootable}} {{volume_id}}
```

openttd

Open source clone of the Microprose game "Transport Tycoon Deluxe".

More information: https://wiki.openttd.org/en/Manual/Command%20line.

- Start a new game:

```
openttd -g
```

- Load save game at start:

```
openttd -g {{path/to/file}}
```

- Start with the specified window resolution:

```
openttd -r {{1920x1080}}
```

- Start with a custom configuration file:

```
openttd -c {{path/to/file}}
```

- Start with selected video, sound, and music drivers:

```
openttd -v {{video_driver}} -s {{sound_driver}} -m
{{music_driver}}
```

- Start a dedicated server, forked in the background:

```
openttd -f -D {{host}}:{{port}}
```

- Join a server with a password:

```
openttd -n {{host}}:{{port}}#{{player_name}} -p {{password}}
```

openvpn

OpenVPN client and daemon binary.

More information: https://openvpn.net/community-docs/community-articles/openvpn-2-6-manual.html.

- Connect to server using a configuration file:

```
sudo openvpn {{path/to/client.conf}}
```

- Try to set up an insecure peer-to-peer tunnel on bob.example.com host:

```
sudo openvpn --remote {{alice.example.com}} --dev {{tun1}} --ifconfig {{10.4.0.1}} {{10.4.0.2}}
```

- Connect to the awaiting bob.example.com host without encryption:

```
sudo openvpn --remote {{bob.example.com}} --dev {{tun1}} --ifconfig {{10.4.0.2}} {{10.4.0.1}}
```

- Create a cryptographic key and save it to file:

```
openvpn --genkey secret {{path/to/key}}
```

- Try to set up a peer-to-peer tunnel on bob.example.com host with a static key:

```
sudo openvpn --remote {{alice.example.com}} --dev {{tun1}} --ifconfig {{10.4.0.1}} {{10.4.0.2}} --secret {{path/to/key}}
```

- Connect to the awaiting bob.example.com host with the same static key as on bob.example.com:

```
sudo openvpn --remote {{bob.example.com}} --dev {{tun1}} --ifconfig {{10.4.0.2}} {{10.4.0.1}} --secret {{path/to/key}}
```

opera

This command is an alias of **`chromium`**.

More information: https://opera.com.

- View documentation for the original command:

```
tldr chromium
```

opt

Run optimizations and analyse LLVM source files.

More information: https://llvm.org/docs/CommandGuide/opt.html.

- Run an optimization or analysis on a bitcode file:

```
opt -{{passname}} {{path/to/file.bc}} -S -o {{file_opt.bc}}
```

- Output the Control Flow Graph of a function to a `.dot` file:

```
opt {{-dot-cfg}} -S {{path/to/file.bc}} -disable-output
```

- Optimize the program at level 2 and output the result to another file:

```
opt -O2 {{path/to/file.bc}} -S -o {{path/to/output_file.bc}}
```

optipng

PNG file optimization utility.

More information: https://optipng.sourceforge.net.

- Compress a PNG with default settings:

```
optipng {{path/to/file.png}}
```

- Compress a PNG with the best compression:

```
optipng -o {{7}} {{path/to/file.png}}
```

- Compress a PNG with the fastest compression:

```
optipng -o {{0}} {{path/to/file.png}}
```

- Compress a PNG and add interlacing:

```
optipng -i {{1}} {{path/to/file.png}}
```

- Compress a PNG and preserve all metadata (including file timestamps):

```
optipng -preserve {{path/to/file.png}}
```

- Compress a PNG and remove all metadata:

```
optipng -strip all {{path/to/file.png}}
```

opusenc

Convert WAV or FLAC audio to Opus.

More information: https://opus-codec.org/docs/opus-tools/opusenc.html.

- Convert WAV to Opus using default options:

```
opusenc {{path/to/input.wav}} {{path/to/output.opus}}
```

- Convert stereo audio at the highest quality level:

```
opusenc --bitrate {{512}} {{path/to/input.wav}} {{path/to/output.opus}}
```

- Convert 5.1 surround sound audio at the highest quality level:

```
opusenc --bitrate {{1536}} {{path/to/input.flac}} {{path/to/output.opus}}
```

- Convert speech audio at the lowest quality level:

```
opusenc {{path/to/input.wav}} --downmix-mono --bitrate {{6}} {{path/to/out.opus}}
```

orca-c

A C-port of the ORCA live programming environment.

ORCA is an esoteric programming language for creating procedural sequencers.

More information: https://github.com/hundredrabbits/Orca-c.

- Start ORCA with an empty workspace:

```
orca-c
```

- Start ORCA and open a specific file:

```
orca-c {{path/to/file.orca}}
```

- Start ORCA and set a specific tempo (defaults to 120):

```
orca-c --bpm {{beats_per_minute}}
```

- Start ORCA and set the size of the grid:

```
orca-c --initial-size {{columns}}x{{rows}}
```

- Start ORCA and set the maximum number of undo steps (defaults to 100):

```
orca-c --undo-limit {{limit}}
```

- Show the main menu inside of ORCA:

```
<F1>
```

- Show all shortcuts inside of ORCA:

```
<?>
```

- Show all ORCA operators inside of ORCA:

```
<Ctrl g>
```

osage

Render an image of a **clustered** network graph from a **graphviz** file.

Layouts: **dot**, **neato**, **twopi**, **circo**, **fdp**, **sfdp**, **osage** & **patchwork**.

More information: https://graphviz.org/doc/info/command.html.

- Render a PNG image with a filename based on the input filename and output format (uppercase -O):

```
osage -T {{png}} -O {{path/to/input.gv}}
```

- Render a SVG image with the specified output filename (lowercase -o):

```
osage -T {{svg}} -o {{path/to/image.svg}} {{path/to/input.gv}}
```

- Render the output in PS, PDF, SVG, Fig, PNG, GIF, JPEG, JSON, or DOT format:

```
osage -T {{format}} -O {{path/to/input.gv}}
```

- Render a GIF image using `stdin` and `stdout`:

```
echo "{{digraph {this -> that} }}" | osage -T {{gif}} > {{path/to/image.gif}}
```

- Display help:

```
osage -?
```

osmium

Multipurpose tool for handling OpenStreetMap (OSM) files.

More information: https://osmcode.org/osmium-tool/manual.

- Show file information:

```
osmium fileinfo {{path/to/input.osm}}
```

- Display contents:

```
osmium show {{path/to/input.osm}}
```

- Convert file format from PBF into XML:

```
osmium cat {{path/to/input.osm.pbf}} {{[-o|--output]}}
{{path/to/output.osm}}
```

- Extract a geographic region by the given [b]ounding box:

```
osmium extract {{[-b|--bbox]}} {{min_longitude}},
{{min_latitude}},{{max_longitude}},{{max_latitude}} {{path/
to/input.pbf}} {{[-o|--output]}} {{path/to/output.pbf}}
```

- Extract a geographic region by a GeoJSON file:

```
osmium extract {{[-p|--polygon]}} {{path/to/polygon.geojson}}
{{path/to/input.pbf}} {{[-o|--output]}} {{path/to/
output.pbf}}
```

- Filter all objects tagged as "restaurant":

```
osmium tags-filter {{path/to/input.pbf}} amenity=restaurant
{{[-o|--output]}} {{path/to/output.pbf}}
```

- Filter for "way" objects tagged as "highway":

```
osmium tags-filter {{path/to/input.pbf}} w/highway {{[-o|--
output]}} {{path/to/output.pbf}}
```

- Filter "way" and "relation" objects tagged as "building":

```
osmium tags-filter {{path/to/input.pbf}} wr/building {{[-o|--
output]}} {{path/to/output.pbf}}
```

osv-scanner

Scan various mediums for dependencies and matches them against the OSV database.

More information: https://osv.dev/about.

- Scan a Docker image:

```
osv-scanner -D {{docker_image_name}}
```

- Scan a package lockfile:

```
osv-scanner -L {{path/to/lockfile}}
```

- Scan an SBOM file:

```
osv-scanner -S {{path/to/sbom_file}}
```

- Scan multiple directories recursively:

```
osv-scanner -r {{directory1 directory2 ...}}
```

- Skip scanning Git repositories:

```
osv-scanner --skip-git {{-r|-D}} {{target}}
```

- Output result in JSON format:

```
osv-scanner --json {{-D|-L|-S|-r}} {{target}}
```

ouch

Utility for compressing and decompressing files and directories.

More information: https://crates.io/crates/ouch.

- Decompress a specific file:

```
ouch decompress {{path/to/archive.tar.xz}}
```

- Decompress a file to a specific location:

```
ouch decompress {{path/to/archive.tar.xz}} --dir {{path/to/directory}}
```

- Decompress multiple files:

```
ouch decompress {{path/to/archive1.tar path/to/archive2.tar.gz ...}}
```

- Compress files:

```
ouch compress {{path/to/file1 path/to/file2 ...}} {{path/to/archive.zip}}
```

oxipng

Losslessly improve compression of PNG files.

More information: https://github.com/shssoichiro/oxipng.

- Compress a PNG file (overwrites the file by default):

```
oxipng {{path/to/file.png}}
```

- Compress a PNG file and save the output to a new file:

```
oxipng --out {{path/to/output.png}} {{path/to/file.png}}
```

- Compress all PNG files in the current directory using multiple threads:

```
oxipng "*.png"
```

- Compress a file with a set optimization level (default is 2):

```
oxipng --opt {{0|1|2|3|4|5|6|max}} {{path/to/file.png}}
```

- Set the PNG interlacing type (`0` removes interlacing, `1` applies Adam7 interlacing, `keep` preserves existing interlacing; default is `0`):

```
oxipng --interlace {{0|1|keep}} {{path/to/file.png}}
```

- Perform additional optimization on images with an alpha channel:

```
oxipng --alpha {{path/to/file.png}}
```

- Use the much slower but stronger Zopfli compressor with max optimization:

```
oxipng --zopfli --opt max {{path/to/file.png}}
```

- Strip all non-critical metadata chunks:

```
oxipng --strip all {{path/to/file.png}}
```

p10k

Manage configurations for powerlevel10k.

More information: https://github.com/romkatv/powerlevel10k.

- Configure powerlevel10k interactively:

```
p10k configure
```

- Reload powerlevel10k:

```
p10k reload
```

- Display help:

```
p10k help
```

p4

Perforce Version Control System.

More information: https://ftp.perforce.com/perforce/r16.2/doc/manuals/cmdref/.

- Log in to the Perforce service:

```
p4 login -a
```

- Create a client:

```
p4 client
```

- Copy files from depot into the client workspace:

```
p4 sync
```

- Create or edit changelist description:

```
p4 change
```

- Open a file to edit:

```
p4 edit -c {{changelist_number}} {{path/to/file}}
```

- Open a new file to add it to the depot:

```
p4 add
```

- Display list of files modified by changelist:

```
p4 describe -c {{changelist_number}}
```

- Submit a changelist to the depot:

```
p4 submit -c {{changelist_number}}
```

p5

Template builder and sketch manager for p5.js.

More information: https://github.com/chiunhau/p5-manager.

- Create a new p5 collection:

  ```
  p5 new {{collection_name}}
  ```

- Generate a new p5 project (should be run from collection directory):

  ```
  p5 generate {{project_name}}
  ```

- Run the p5 manager server:

  ```
  p5 server
  ```

- Update libraries to their latest versions:

  ```
  p5 update
  ```

p7zip

Wrapper of 7-Zip file archiver with high compression ratio.

Internally executes either 7za or 7zr command.

More information: https://p7zip.sourceforge.net.

- Archive a file, replacing it with a 7zipped compressed version:

```
p7zip {{path/to/file}}
```

- Archive a file keeping the input file:

```
p7zip {{[-k|--keep]}} {{path/to/file}}
```

- Decompress a file, replacing it with the original uncompressed version:

```
p7zip {{[-d|--decompress]}} {{compressed.ext}}.7z
```

- Decompress a file keeping the input file:

```
p7zip {{[-d|--decompress]}} {{[-k|--keep]}}
{{compressed.ext}}.7z
```

- Skip some checks and force compression or decompression:

```
p7zip {{[-f|--force]}} {{path/to/file}}
```

paci

A package manager for Bash scripts.

More information: https://github.com/tradebyte/paci.

- Update the list of available packages and versions (it's recommended to run this before other `paci` commands):

```
paci refresh
```

- Configure its behaviour:

```
paci configure
```

- Search for a given package:

```
paci search {{package}}
```

- Install a package:

```
paci install {{package}}
```

- Update a package:

```
paci update {{package}}
```

packer

Build automated machine images.

More information: https://developer.hashicorp.com/packer/docs/commands.

- Build an image:

```
packer build {{path/to/config.json}}
```

- Check the syntax of a Packer image config:

```
packer validate {{path/to/config.json}}
```

- Format a Packer image config:

```
packer fmt {{path/to/config.pkr.hcl}}
```

packtpub

Download freely offered books from packtpub.com.

More information: https://github.com/vladimyr/packtpub-cli.

- Download the daily offer book into the current directory with the specified book format (defaults to `pdf`):

```
packtpub download {{[-t|--type]}} {{pdf|ebup|mobi}}
```

- Download the daily offer book into the specified directory:

```
packtpub download {{[-d|--dir]}} {{path/to/directory}}
```

- Start an interactive login to packtpub.com:

```
packtpub login
```

- Log out from packtpub.com:

```
packtpub logout
```

- Display the daily offer:

```
packtpub view-offer
```

- Open the daily offer in the default web browser:

```
packtpub view-offer
```

- Display the currently logged-in user:

```
packtpub whoami
```

packwiz

Create, edit and manage Minecraft modpacks.

More information: https://packwiz.infra.link/reference/commands/packwiz/.

- Interactively create a new modpack in the current directory:

  ```
  packwiz init
  ```

- Add a mod from Modrinth or Curseforge:

  ```
  packwiz {{modrinth|curseforge}} add {{url|slug|search_term}}
  ```

- List all mods in the modpack:

  ```
  packwiz list
  ```

- Update `index.toml` after manually editing files:

  ```
  packwiz refresh
  ```

- Export as a Modrinth (`.mrpack`) or Curseforge (Zip) file:

  ```
  packwiz {{modrinth|curseforge}} export
  ```

pactl

Control a running PulseAudio sound server.

More information: https://manned.org/pactl.

- Show information about the sound server:

```
pactl info
```

- List all sinks (or other types - sinks are outputs and sink-inputs are active audio streams):

```
pactl list {{sinks}} short
```

- Change the default sink (output) to 1 (the number can be retrieved via the `list` subcommand):

```
pactl set-default-sink {{1}}
```

- Move sink-input 627 to sink 1:

```
pactl move-sink-input {{627}} {{1}}
```

- Set the volume of sink 1 to 75%:

```
pactl set-sink-volume {{1}} {{0.75}}
```

- Toggle mute on the default sink (using the special name `@DEFAULT_SINK@`):

```
pactl set-sink-mute {{@DEFAULT_SINK@}} toggle
```

pageres

Capture screenshots of websites in various resolutions.

More information: https://github.com/sindresorhus/pageres-cli.

- Take multiple screenshots of multiple URLs at different resolutions:

```
pageres {{https://example.com/}} {{https://example2.com/}} {{1366x768}} {{1600x900}}
```

- Provide specific options for a URL, overriding global options:

```
pageres [{{https://example.com/}} {{1366x768}} --no-crop] [{{https://example2.com/}} {{1024x768}}] {{[-c|--crop]}}
```

- Provide a custom filename template:

```
pageres {{https://example.com/}} {{1024x768}} --filename='{{<%= date %> - <%= url %>}}'
```

- Capture a specific element on a page:

```
pageres {{https://example.com/}} {{1366x768}} --selector='{{.page-header}}'
```

- Hide a specific element:

```
pageres {{https://example.com/}} {{1366x768}} --hide='{{.page-header}}'
```

- Capture a screenshot of a local file:

```
pageres {{path/to/local_file.html}} {{1366x768}}
```

pake

Turn any webpage into a desktop app with Rust/Tauri.

More information: https://github.com/tw93/Pake.

- Package a web page:

```
pake {{https://www.example.com/}}
```

- Package a web page with a specific window size:

```
pake --width {{800}} --height {{600}} {{https://
www.example.com/}}
```

- Package a web page with a custom application name and icon:

```
pake --name {{application_name}} --icon {{path/to/icon.ico}}
{{https://www.example.com/}}
```

- Package a web page with a non-resizable window:

```
pake --no-resizable {{https://www.example.com/}}
```

- Package a web page with fullscreen mode:

```
pake --fullscreen {{https://www.example.com/}}
```

- Package a web page with a transparent title bar:

```
pake --transparent {{https://www.example.com/}}
```

palmtopnm

Convert a Palm bitmap file to a PNM image.

More information: https://netpbm.sourceforge.net/doc/palmtopnm.html.

- Convert a Palm bitmap to a PNM image:

```
palmtopnm {{path/to/file.palm}} > {{path/to/file.pnm}}
```

- Display information about the input file:

```
palmtopnm {{[-verb|-verbose]}} {{path/to/file.palm}} >
{{path/to/file.pnm}}
```

- Convert the n'th rendition of the image contained in the input file:

```
palmtopnm {{[-r|-rendition]}} {{n}} {{path/to/file.palm}} >
{{path/to/file.pnm}}
```

- Write a histogram of the colors in the input file to `stdout`:

```
palmtopnm {{[-s|-showhist]}} {{path/to/file.palm}} > {{path/
to/file.pnm}}
```

- Output the transparent color of the input image if set:

```
palmtopnm {{[-t|-transparent]}} {{path/to/file.palm}}
```

pamarith

Apply a binary function on two Netpbm images.

See also: **pamfunc**.

More information: https://netpbm.sourceforge.net/doc/pamarith.html.

- Apply the specified binary function pixel-wise on the two specified images (which must be of the same size):

```
pamarith -{{add|subtract|multiply|divide|difference|minimum|maximum|...}} {{path/to/image1.pam|pbm|pgm|ppm}} {{path/to/image2.pam|pbm|pgm|ppm}}
```

pambackground

Create a mask of the background in a PAM image.

See also: **pbmmask**.

More information: https://netpbm.sourceforge.net/doc/pambackground.html.

- Create a mask of the background in a PAM image:

```
pambackground {{path/to/image.pam}} > {{path/to/output.pam}}
```

pambrighten

Change a PAM image's saturation and value.

More information: https://netpbm.sourceforge.net/doc/pambrighten.html.

- Increase the saturation of each pixel by the specified percentage:

```
pambrighten {{[-s|-saturation]}} {{value_percent}} {{path/to/image.pam}} > {{path/to/output.pam}}
```

- Increase the value (from the HSV color space) of each pixel by the specified percentage:

```
pambrighten {{[-va|-value]}} {{value_percent}} {{path/to/image.pam}} > {{path/to/output.pam}}
```

pamcomp

Overlay two PAM images.

More information: https://netpbm.sourceforge.net/doc/pamcomp.html.

- Overlay two images such with the overlay blocking parts of the underlay:

```
pamcomp {{path/to/overlay.pam}} {{path/to/underlay.pam}} > {{path/to/output.pam}}
```

- Set the horizontal alignment of the overlay:

```
pamcomp {{[-ali|-align]}} {{left|center|right|beyondleft|beyondright}} {{[-x|-xoff]}} {{x_offset}} {{path/to/overlay.pam}} {{path/to/underlay.pam}} > {{path/to/output.pam}}
```

- Set the vertical alignment of the overlay:

```
pamcomp {{[-va|-valign]}} {{top|middle|bottom|above|below}} {{[-y|-yoff]}} {{y_offset}} {{path/to/overlay.pam}} {{path/to/underlay.pam}} > {{path/to/output.pam}}
```

- Set the opacity of the overlay:

```
pamcomp {{[-o|-opacity]}} {{0.7}} {{path/to/overlay.pam}} {{path/to/underlay.pam}} > {{path/to/output.pam}}
```

pamcrater

Create a PAM image of cratered terrain.

See also: **pamshadedrelief**, **ppmrelief**.

More information: https://netpbm.sourceforge.net/doc/pamcrater.html.

- Create an image of cratered terrain with the specified dimensions:

```
pamcrater {{[-h|-height]}} {{height}} {{[-w|-width]}}
{{width}} > {{path/to/output.pam}}
```

- Create an image containing the specified number of craters:

```
pamcrater {{[-n|-number]}} {{n_craters}} > {{path/to/
output.pam}}
```

pamcut

Cut out a rectangular region from a Netpbm image.

See also: **pamcrop**, **pamdice**, **pamcomp**.

More information: https://netpbm.sourceforge.net/doc/pamcut.html.

- Discard the specified number of columns/rows on each side of the image:

```
pamcut {{[-cropl|-cropleft]}} {{value}} {{[-cropr|-
cropright]}} {{value}} {{[-cropt|-croptop]}} {{value}} {{[-
cropb|-cropbottom]}} {{value}} {{path/to/image.ppm}} >
{{path/to/output.ppm}}
```

- Keep only the columns between the specified columns (inclusively):

```
pamcut {{[-l|-left]}} {{value}} {{[-ri|-right]}} {{value}}
{{path/to/image.ppm}} > {{path/to/output.ppm}}
```

- Fill missing areas with black pixels if the specified rectangle does not entirely lie within the input image:

```
pamcut {{[-t|-top]}} {{value}} {{[-b|-bottom]}} {{value}} -
pad {{path/to/image.ppm}} > {{path/to/output.ppm}}
```

pamdeinterlace

Remove every other row in a Netpbm image.

See also: **pammixinterlace**.

More information: https://netpbm.sourceforge.net/doc/pamdeinterlace.html.

- Produce an image consisting of the input's even-numbered rows:

```
pamdeinterlace {{path/to/image.ppm}} > {{path/to/output.ppm}}
```

- Produce an image consisting of the input's odd-numbered rows:

```
pamdeinterlace {{[-takeo|-takeodd]}} {{path/to/image.ppm}} >
{{path/to/output.ppm}}
```

pamdepth

Reduce the depth (i.e. color resolution) in an image.

More information: https://netpbm.sourceforge.net/doc/pamdepth.html.

- Read a PBM image, set its maxval and save it to a file:

```
pamdepth {{maxval}} {{path/to/image.pbm}} > {{path/to/file.pbm}}
```

pamdice

Slice a Netpbm image vertically or horizontally.

See also: **pamundice**.

More information: https://netpbm.sourceforge.net/doc/pamdice.html.

- Slice a Netpbm image such that the resulting tiles have the specified height and width:

```
pamdice {{[-o|-outstem]}} {{path/to/filename_stem}} {{[-he|-height]}} {{value}} {{[-w|-width]}} {{value}} {{path/to/input.ppm}}
```

- Make the produced pieces overlap by the specified amount horizontally and vertically:

```
pamdice {{[-o|-outstem]}} {{path/to/filename_stem}} {{[-he|-height]}} {{value}} {{[-w|-width]}} {{value}} {{[-ho|-hoverlap]}} {{value}} {{[-vo|-voverlap]}} {{value}} {{path/to/input.ppm}}
```

pamditherbw

Apply dithering to a greyscale image, i.e. turn it into a pattern of black and white pixels that look the same as the original greyscale.

See also: **pbmreduce**.

More information: https://netpbm.sourceforge.net/doc/pamditherbw.html.

- Read a PGM image, apply dithering and save it to a file:

```
pamditherbw {{path/to/image.pgm}} > {{path/to/file.pgm}}
```

- Use the specified quantization method:

```
pamditherbw -{{floyd|fs|atkinson|threshold|hilbert|...}}
{{path/to/image.pgm}} > {{path/to/file.pgm}}
```

- Use the atkinson quantization method and the specified seed for a pseudo-random number generator:

```
pamditherbw {{[-a|-atkinson]}} {{[-r|-randomseed]}} {{1337}}
{{path/to/image.pgm}} > {{path/to/file.pgm}}
```

- Specify the thresholding value for quantization methods that perform some sort of thresholding:

```
pamditherbw -{{fs|atkinson|thresholding}} {{[-va|-value]}}
{{0.3}} {{path/to/image.pgm}} > {{path/to/file.pgm}}
```

pamedge

Perform edge-detection on a Netpbm image.

More information: https://netpbm.sourceforge.net/doc/pamedge.html.

- Perform edge-detection on a Netpbm image:

```
pamedge {{path/to/input.pam}} > {{path/to/output.pam}}
```

pamenlarge

Enlarge a PAM image by duplicating pixels.

See also: **pbmreduce**, **pamditherbw**, **pbmpscale**.

More information: https://netpbm.sourceforge.net/doc/pamenlarge.html.

- Enlarge the specified image by the specified factor:

```
pamenlarge {{[-s|-scale]}} {{n}} {{path/to/image.pam}} >
{{path/to/output.pam}}
```

- Enlarge the specified image by the specified factors horizontally and vertically:

```
pamenlarge {{[-x|-xscale]}} {{xn}} {{[-y|-yscale]}} {{yn}}
{{path/to/image.pam}} > {{path/to/output.pam}}
```

pamexec

Execute a shell command on each image in a Netpbm file.

See also: **`pamfile`**, **`pampick`**, **`pamsplit`**.

More information: https://netpbm.sourceforge.net/doc/pamexec.html.

- Execute a shell command on each image in a Netpbm file:

```
pamexec {{command}} {{path/to/image.pam}}
```

- Stop processing if a command terminates with a nonzero exit status:

```
pamexec {{command}} {{path/to/image.pam}} {{[-c|-check]}}
```

pamfile

Describe Netpbm (PAM or PNM) files.

More information: https://netpbm.sourceforge.net/doc/pamfile.html.

- Describe the specified Netpbm files:

```
pamfile {{path/to/file1 path/to/file2 ...}}
```

- Describe every image in each input file (as opposed to only the first image in each file) in a machine-readable format:

```
pamfile {{[-a|-allimages]}} -machine {{path/to/file}}
```

- Display a count on how many images the input files contain:

```
pamfile {{[-cou|-count]}} {{path/to/file}}
```

pamfix

Fix errors in PAM, PBM, PGM and PPM files.

See also: **`pamfile`**, **`pamvalidate`**.

More information: https://netpbm.sourceforge.net/doc/pamfix.html.

- Fix a Netpbm file that is missing its last part:

```
pamfix {{[-t|-truncate]}} {{path/to/corrupted.ext}} > {{path/to/output.ext}}
```

- Fix a Netpbm file where pixel values exceed the image's `maxval` by lowering the offending pixels' values:

```
pamfix {{[-cl|-clip]}} {{path/to/corrupted.ext}} > {{path/to/output.ext}}
```

- Fix a Netpbm file where pixel values exceed the image's `maxval` by increasing it:

```
pamfix {{[-ch|-changemaxval]}} {{path/to/corrupted.pam|pbm|pgm|ppm}} > {{path/to/output.pam|pbm|pgm|ppm}}
```

pamfixtrunc

This command has been superseded by **`pamfix -truncate`**.

More information: https://netpbm.sourceforge.net/doc/pamfixtrunc.html.

- View documentation for the current command:

```
tldr pamfix
```

pamflip

Flip or rotate a PAM or PNM image.

More information: https://netpbm.sourceforge.net/doc/pamflip.html.

- Rotate the input image counter-clockwise for a specific degree:

```
pamflip {{[-r|-rotate]}}{{90|180|270}} {{path/to/input.pam}} > {{path/to/output.pam}}
```

- Flip left for right:

```
pamflip {{[-lr|-leftright]}} {{path/to/input.pam}} > {{path/to/output.pam}}
```

- Flip top for bottom:

```
pamflip {{[-tb|-topbottom]}} {{path/to/input.pam}} > {{path/to/output.pam}}
```

- Flip the input image on the main diagonal:

```
pamflip {{[-xy|-transpose]}} {{path/to/input.pam}} > {{path/to/output.pam}}
```

pamfunc

Apply a simple arithmetic function to a Netpbm image.

More information: https://netpbm.sourceforge.net/doc/pamfunc.html.

- Apply the specified arithmetic function with n as the second argument to each sample in the specified PAM image:

```
pamfunc -{{multiplier|divisor|adder|subtractor|min|max}}
{{n}} {{path/to/input.pam}} > {{path/to/output.pam}}
```

- Apply the specified bit string function with n as the second argument to each sample in the specified PAM image:

```
pamfunc -{{andmask|ormask|xormask|shiftleft|shiftright}}
{{n}} {{path/to/input.pam}} > {{path/to/output.pam}}
```

pamixer

A simple command-line mixer for PulseAudio.

More information: https://github.com/cdemoulins/pamixer#installation.

- List all sinks and sources with their corresponding IDs:

```
pamixer --list-sinks --list-sources
```

- Set the volume to 75% on the default sink:

```
pamixer --set-volume {{75}}
```

- Toggle mute on a sink other than the default:

```
pamixer --toggle-mute --sink {{ID}}
```

- Increase the volume on default sink by 5%:

```
pamixer {{[-i|--increase]}} {{5}}
```

- Decrease the volume on a source by 5%:

```
pamixer {{[-d|--decrease]}} {{5}} --source {{ID}}
```

- Use the allow boost option to increase, decrease, or set the volume above 100%:

```
pamixer --set-volume {{105}} --allow-boost
```

- Mute the default sink (use `--unmute` instead to unmute):

```
pamixer {{[-m|--mute]}}
```

pammixinterlace

Merge each row in an image with its two neighbours.

See also: **pamdeinterlace**.

More information: https://netpbm.sourceforge.net/doc/pammixinterlace.html.

- Merge each row in an image with its two neighbours:

```
pammixinterlace {{path/to/image.ppm}} > {{path/to/
output.ppm}}
```

- Use the specified filtering mechanism:

```
pammixinterlace {{[-f|-filter]}} {{linear|fir|ffmpeg}}
{{path/to/image.ppm}} > {{path/to/output.ppm}}
```

- Turn on adaptive filtering mode, i.e., only modify pixels that are obviously part of a comb pattern:

```
pammixinterlace {{[-a|-adaptive]}} {{path/to/image.ppm}} >
{{path/to/output.ppm}}
```

pamnoraw

This command is an alias of **`pamtopnm -plain`**.

- View documentation for the original command:

  ```
  tldr pamtopnm
  ```

pamoil

Turn a PAM image into an oil painting.

More information: https://netpbm.sourceforge.net/doc/pamoil.html.

- Turn a PAM image into an oil painting:

```
pamoil {{path/to/input_file.pam}} > {{path/to/
output_file.pam}}
```

- Consider a neighborhood of n pixels for the "smearing" effect:

```
pamoil -n {{n}} {{path/to/input_file.pam}} > {{path/to/
output_file.pam}}
```

pampick

Pick images out of a multi-image Netpbm stream.

See also: **pamfile**, **pamsplit**.

More information: https://netpbm.sourceforge.net/doc/pampick.html.

- Execute a shell command on each image in a Netpbm file:

```
pampick < {{path/to/image.pam}} {{image_number1 image_number2 ...}} > {{path/to/output.pam}}
```

pampop9

Simulate a multi-lens camera such as the Pop9.

More information: https://netpbm.sourceforge.net/doc/pampop9.html.

- Tile the input image xtiles by ytiles times, increasing the offset each time as determined by xdelta and ydelta:

```
pampop9 {{path/to/input.pam}} {{xtiles}} {{ytiles}} {{xdelta}} {{ydelta}} > {{path/to/output.pam}}
```

pamrgbatopng

This command has been superseded by **pamtopng**.

More information: https://netpbm.sourceforge.net/doc/pamrgbatopng.html.

- View documentation for the current command:

```
tldr pamtopng
```

pamscale

Scale a Netpbm image.

More information: https://netpbm.sourceforge.net/doc/pamscale.html.

- Scale an image such that the result has the specified dimensions:

```
pamscale {{[-wid|-width]}} {{width}} {{[-h|-height]}} {{height}} {{path/to/input.pam}} > {{path/to/output.pam}}
```

- Scale an image such that the result has the specified width, keeping the aspect ratio:

```
pamscale {{[-wid|-width]}} {{width}} {{path/to/input.pam}} > {{path/to/output.pam}}
```

- Scale an image such that its width and height is changed by the specified factors:

```
pamscale {{[-xsc|-xscale]}} {{x_factor}} {{[-ysc|-yscale]}} {{y_factor}} {{path/to/input.pam}} > {{path/to/output.pam}}
```

- Scale an image such that it fits into the specified bounding box while preserving its aspect ratio:

```
pamscale -xyfit {{bbox_width}} {{bbox_height}} {{path/to/input.pam}} > {{path/to/output.pam}}
```

- Scale an image such that it completely fills the specified box while preserving its aspect ratio:

```
pamscale -xyfill {{box_width}} {{box_height}} {{path/to/input.pam}} > {{path/to/output.pam}}
```

pamshadedrelief

Generate a shaded relief from an elevation map.

See also: **pamcrater**, **ppmrelief**.

More information: https://netpbm.sourceforge.net/doc/pamshadedrelief.html.

- Generate a shaded relief image with the input image interpreted as an elevation map:

```
pamshadedrelief < {{path/to/input.pam}} > {{path/to/output.pam}}
```

- Gamma adjust the image by the specified factor:

```
pamshadedrelief < {{path/to/input.pam}} {{[-g|-gamma]}} {{factor}} > {{path/to/output.pam}}
```

pamslice

Extract one line of values out of a PAM image.

More information: https://netpbm.sourceforge.net/doc/pamslice.html.

- Print the values of the pixels in the n'th row in a table:

```
pamslice {{[-r|-row]}} {{n}} {{path/to/image.pam}}
```

- Print the values of the pixels in the n'th column in a table:

```
pamslice {{[-c|-column]}} {{n}} {{path/to/image.pam}}
```

- Consider the m'th plane of the input image only:

```
pamslice {{[-r|-row]}} {{n}} -plane {{m}} {{path/to/
image.pam}}
```

- Produce output in a format suitable for input to an `xmgr` for visualisation:

```
pamslice {{[-r|-row]}} {{n}} {{[-x|-xmgr]}} {{path/to/
image.pam}}
```

pamsplit

Split a multi-image Netpbm file into multiple single-image Netpbm files.

See also: **`pamfile`**, **`pampick`**, **`pamexec`**.

More information: https://netpbm.sourceforge.net/doc/pamsplit.html.

- Split a multi-image Netpbm file into multiple single-image Netpbm files:

```
pamsplit {{path/to/image.pam}}
```

- Specify a pattern for naming output files:

```
pamsplit {{path/to/image.pam}} {{file_%d.pam}}
```

pamstack

Stack the planes of multiple PAM images into one PAM image.

More information: https://netpbm.sourceforge.net/doc/pamstack.html.

- Stack the planes of the specified PAM images in the specified order:

```
pamstack {{path/to/image1.pam path/to/image2.pam ...}} >
{{path/to/output.pam}}
```

- Specify the tuple type name of the output PAM file (maximum of 255 characters):

```
pamstack {{[-t|-tupletype]}} {{tuple_type}} {{path/to/
image1.pam path/to/image2.pam ...}} > {{path/to/output.pam}}
```

pamstretch-gen

Scale up a PAM image by decimal values.

See also: **pamstretch**, **pamenlarge**, **pamscale**.

More information: https://netpbm.sourceforge.net/doc/pamstretch-gen.html.

- Scale up a PAM image by the specified decimal factor:

```
pamstretch-gen {{n}} {{path/to/image.pam}} > {{path/to/
output.pam}}
```

pamstretch

Scale up a PAM image by interpolating between pixels.

See also: **pamstretch-gen**, **pamenlarge**, **pamscale**.

More information: https://netpbm.sourceforge.net/doc/pamstretch.html.

- Scale up a PAM image by an integer factor:

```
pamstretch {{n}} {{path/to/image.pam}} > {{path/to/
output.pam}}
```

- Scale up a PAM image by the specified factors in the horizontal and vertical directions:

```
pamstretch {{[-x|-xscale]}} {{xn}} {{[-y|-yscale]}} {{yn}}
{{path/to/image.pam}} > {{path/to/output.pam}}
```

pamtofits

Convert a Netpbm image to the Flexible Image Transport System (FITS) format.

See also: **`fitstopnm`**.

More information: https://netpbm.sourceforge.net/doc/pamtofits.html.

- Convert a Netpbm image to the FITS format:

```
pamtofits {{path/to/image.pam}} > {{path/to/output.fits}}
```

pamtogif

Convert a Netpbm image into an unanimated GIF image.

See also: **`giftopnm`**, **`gifsicle`**.

More information: https://netpbm.sourceforge.net/doc/pamtogif.html.

- Convert a Netpbm image into an unanimated GIF image:

```
pamtogif {{path/to/image.pam}} > {{path/to/output.gif}}
```

- Mark the specified color as transparent in the output GIF file:

```
pamtogif {{[-t|-transparent]}} {{color}} {{path/to/
image.pam}} > {{path/to/output.gif}}
```

- Include the specified text as a comment in the output GIF file:

```
pamtogif {{[-c|-comment]}} "{{Hello World!}}" {{path/to/
image.pam}} > {{path/to/output.gif}}
```

pamtopam

Copy a PAM image.

More information: https://netpbm.sourceforge.net/doc/pamtopam.html.

- Copy a PAM image (i.e. a PBM, PGM, PPM or PAM image) from `stdin` to `stdout`:

```
pamtopam < {{path/to/image.pam}} > {{path/to/output.pam}}
```

- Display version:

```
pamtopam {{[-v|-version]}}
```

pamtopng

Convert a PAM image to PNG.

See also: **pnmtopng**, **pngtopam**.

More information: https://netpbm.sourceforge.net/doc/pamtopng.html.

- Convert the specified PAM image to PNG:

```
pamtopng {{path/to/image.pam}} > {{path/to/output.png}}
```

- Mark the specified color as transparent in the output image:

```
pamtopng {{[-t|-transparent]}} {{color}} {{path/to/
image.pam}} > {{path/to/output.png}}
```

- Include the text in the specified file as tEXt chunks in the output:

```
pamtopng {{[-te|-text]}} {{path/to/file.txt}} {{path/to/
image.pam}} > {{path/to/output.png}}
```

- Cause the output file to be interlaced in Adam7 format:

```
pamtopng {{[-in|-interlace]}} {{path/to/image.pam}} > {{path/
to/output.png}}
```

pamtopnm

Convert a PAM image to an equivalent PNM image.

More information: https://netpbm.sourceforge.net/doc/pamtopnm.html.

- Convert a PAM image to an equivalent PNM image, i.e. a PBM, PGM or PPM image:

```
pamtopnm {{path/to/image.pam}} > {{path/to/output.pbm|pgm|ppm}}
```

- Display version:

```
pamtopnm {{[-v|-version]}}
```

pamtoqoi

Convert a Netpbm image to a QOI image (Quite OK Image format).

More information: https://netpbm.sourceforge.net/doc/pamtoqoi.html.

- Convert a Netpbm image to the QOI format:

```
pamtoqoi {{path/to/image.pnm}} > {{path/to/output.qoi}}
```

pamtosrf

Convert Netpbm images to SRF format.

See also: **srftopam**.

More information: https://netpbm.sourceforge.net/doc/pamtosrf.html.

- Convert a Netpbm image or a stream of Netpbm images to SRF format:

```
pamtosrf {{path/to/file.pam}} > {{path/to/output.srf}}
```

pamtotga

Convert a Netpbm image to a TrueVision Targa file.

More information: https://netpbm.sourceforge.net/doc/pamtotga.html.

- Convert a Netpbm image to a TrueVision Targa file:

```
pamtotga {{path/to/file.pam}} > {{path/to/output.tga}}
```

- Specify the color map of the output image:

```
pamtotga -{{cmap|cmap16|mono|rgb}} {{path/to/file.pam}} >
{{path/to/output.tga}}
```

- Display version:

```
pamtotga {{[-v|-version]}}
```

pamtotiff

Convert a PAM image to a TIFF file.

More information: https://netpbm.sourceforge.net/doc/pamtotiff.html.

- Convert a PAM image to a TIFF image:

```
pamtotiff {{path/to/input_file.pam}} > {{path/to/
output_file.tiff}}
```

- Explicitly specify a compression method for the output file:

```
pamtotiff -{{none|packbits|lzw|g3|g4|flate|adobeflate}}
{{path/to/input_file.pam}} > {{path/to/output_file.tiff}}
```

- Always produce a color TIFF image, even if the input image is greyscale:

```
pamtotiff {{[-c|-color]}} {{path/to/input_file.pam}} >
{{path/to/output_file.tiff}}
```

pamtouil

Convert a PNM or PAM file into a Motif UIL icon file.

More information: https://netpbm.sourceforge.net/doc/pamtouil.html.

- Convert a PNM or PAM file into a Motif UIL icon file:

```
pamtouil {{path/to/input.pnm|pam}} > {{path/to/output.uil}}
```

- Specify a prefix string to be printed in the output UIL file:

```
pamtouil {{[-n|-name]}} {{uilname}} {{path/to/input.pnm|pam}}
> {{path/to/output.uil}}
```

pamtowinicon

Convert a PAM image to a Windows ICO file.

More information: https://netpbm.sourceforge.net/doc/pamtowinicon.html.

- Convert a PAM image file to an ICO file:

```
pamtowinicon {{path/to/input_file.pam}} > {{path/to/
output.ico}}
```

- Encode images with resolutions smaller than t in the BMP format and all other images in the PNG format:

```
pamtowinicon {{[-pn|-pngthreshold]}} {{t}} {{path/to/
input_file.pam}} > {{path/to/output.ico}}
```

- Make all pixels outside the non-opaque area black:

```
pamtowinicon {{[-t|-truetransparent]}} {{path/to/
input_file.pam}} > {{path/to/output.ico}}
```

pamtoxvmini

Convert a Netpbm image to an XV thumbnail picture.

More information: https://netpbm.sourceforge.net/doc/pamtoxvmini.html.

- Convert a PAM image to an XV thumbnail picture:

```
pamtoxvmini {{path/to/input_file.pam}} > {{path/to/
output_file}}
```

pamundice

Combine a grid of Netpbm images into one.

See also: **pamdice**.

More information: https://netpbm.sourceforge.net/doc/pamundice.html.

- Combine the images whose names match the `printf`-style filename expression. Assume a grid with a specific size:

```
pamundice {{filename_%1d_%1a.ppm}} {{[-a|-across]}} {{grid_width}} {{[-d|-down]}} {{grid_height}} > {{path/to/output.ppm}}
```

- Assume that the tiles overlap horizontally and vertically by the specified amount:

```
pamundice {{filename_%1d_%1a.ppm}} {{[-a|-across]}} {{x_value}} {{[-d|-down]}} {{y_value}} {{[-ho|-hoverlap]}} {{value}} {{[-vo|-voverlap]}} {{value}} > {{path/to/output.ppm}}
```

- Specify the images to be combined through a text file containing one filename per line:

```
pamundice {{[-l|-listfile]}} {{path/to/file.txt}} {{[-a|-across]}} {{x_value}} {{[-d|-down]}} {{y_value}} > {{path/to/output.ppm}}
```

pamvalidate

Validate PAM, PGM, PBM and PPM files.

See also: **`pamfile`**, **`pamfix`**.

More information: https://netpbm.sourceforge.net/doc/pamvalidate.html.

- Copy a Netpbm file from `stdin` to `stdout` if and only if it valid; fail otherwise:

```
{{command}} | pamvalidate > {{path/to/output.ext}}
```

pandoc

Convert documents between various formats.

More information: https://pandoc.org/MANUAL.html.

- Convert a Markdown file to PDF using `pdflatex` (the formats are determined by file extensions):

```
pandoc {{path/to/input.md}} {{[-o|--output]}} {{path/to/output.pdf}}
```

- Convert the output from another command to PDF, using a specific PDF engine:

```
{{command}} | pandoc {{[-f|--from]}} {{input_format}} --pdf-engine {{tectonic|weasyprint|typst|...}} {{[-o|--output]}} {{path/to/output.pdf}}
```

- Convert to a standalone file with the appropriate headers/footers (for LaTeX, HTML, etc.):

```
pandoc {{path/to/input.md}} {{[-s|--standalone]}} {{[-o|--output]}} {{path/to/output.html}}
```

- Manually specify formats (overriding automatic format detection using the filename extension, or when there is no extension):

```
pandoc {{[-f|--from]}} {{docx|...}} {{path/to/input}} {{[-t|--to]}} {{pdf|...}} {{[-o|--output]}} {{path/to/output}}
```

- Transform a document using a Lua script (see https://pandoc.org/lua-filters.html for more information):

```
pandoc {{path/to/input}} {{[-L|--lua-filter]}} {{path/to/filter.lua}} {{[-o|--output]}} {{path/to/output}}
```

- Convert a remote HTML file to markdown and print the result to `stdout`:

```
pandoc {{[-f|--from]}} html {{[-t|--to]}} markdown {{https://example.com}}
```

- List all supported input formats:

```
pandoc --list-input-formats
```

- List all supported output formats:

```
pandoc --list-output-formats
```

pangolin

Implements the dynamic nomenclature of SARS-CoV-2 lineages (Pango nomenclature).

More information: https://cov-lineages.org/resources/pangolin/usage.html.

- Run `pangolin` on the specified FASTA file:

  ```
  pangolin {{path/to/file.fa}}
  ```

- Use the specified analysis engine:

  ```
  pangolin --analysis-mode {{accurate|fast|pangolearn|usher}}
  ```

pants

Fast, scalable, user-friendly, open-source build and developer workflow tool.

More information: https://www.pantsbuild.org/stable/docs/using-pants/command-line-help.

- List all targets:

```
pants list ::
```

- Run all tests:

```
pants test ::
```

- Fix, format, and lint only uncommitted files:

```
pants --changed-since=HEAD fix fmt lint
```

- Typecheck only uncommitted files and their dependents:

```
pants --changed-since=HEAD --changed-dependents=transitive check
```

- Create a distributable package for the specified target:

```
pants package {{path/to/directory:target-name}}
```

- Auto-generate BUILD file targets for new source files:

```
pants tailor ::
```

- Display help:

```
pants help
```

paperkey

An OpenPGP key archiver.

More information: https://www.jabberwocky.com/software/paperkey/.

- Take a specific secret key and generate a text file with the secret data:

```
paperkey --secret-key {{path/to/secret_key.gpg}} --output
{{path/to/secret_data.txt}}
```

- Take the secret key data in `secret_data.txt` and combine it with the public key to reconstruct the secret key:

```
paperkey --pubring {{path/to/public_key.gpg}} --secrets
{{path/to/secret_data.txt}} --output {{secret_key.gpg}}
```

- Export a specific secret key and generate a text file with the secret data:

```
gpg --export-secret-key {{key}} | paperkey --output {{path/
to/secret_data.txt}}
```

papermc

Manage a Minecraft server.

More information: https://github.com/Edenhofer/minecraft-server/blob/main/minecraftd.sh.in#L480.

- Start a server:

```
papermc start
```

- Stop a server:

```
papermc stop
```

- Restart a server:

```
papermc restart
```

- Backup a world:

```
papermc backup
```

- Restore world data from a backup:

```
papermc restore {{path/to/file}}
```

- Display help:

```
papermc
```

par2

File verification and repair using PAR 2.0 compatible parity archives (.par2 files).

More information: https://github.com/Parchive/par2cmdline/.

- Create a parity archive with a set percentage level of redundancy:

```
par2 create -r{{1..100}} -- {{path/to/file}}
```

- Create a parity archive with a chosen number of volume files (in addition to the index file):

```
par2 create -n{{1..32768}} -- {{path/to/file}}
```

- Verify a file with a parity archive:

```
par2 verify -- {{path/to/file.par2}}
```

- Repair a file with a parity archive:

```
par2 repair -- {{path/to/file.par2}}
```

parallel-lint

Check the syntax of PHP files in parallel.

More information: https://github.com/JakubOnderka/PHP-Parallel-Lint.

- Lint a specific directory:

```
parallel-lint {{path/to/directory}}
```

- Lint a directory using the specified number of parallel processes:

```
parallel-lint -j {{processes}} {{path/to/directory}}
```

- Lint a directory, excluding the specified directory:

```
parallel-lint --exclude {{path/to/excluded_directory}} {{path/to/directory}}
```

- Lint a directory of files using a comma-separated list of extension(s):

```
parallel-lint -e {{php,html,phpt}} {{path/to/directory}}
```

- Lint a directory and output the results as JSON:

```
parallel-lint --json {{path/to/directory}}
```

- Lint a directory and show Git Blame results for rows containing errors:

```
parallel-lint --blame {{path/to/directory}}
```

parallel

Run commands on multiple CPU cores.

More information: https://www.gnu.org/software/parallel/man.html.

- Gzip several files at once, using all cores:

```
parallel gzip ::: {{path/to/file1 path/to/file2 ...}}
```

- Read arguments from `stdin`, run 4 jobs at once:

```
ls *.txt | parallel {{[-j|--jobs]}} 4 gzip
```

- Convert JPEG images to PNG using replacement strings:

```
parallel convert {} {.}.png ::: *.jpg
```

- Parallel xargs, cram as many args as possible onto one command:

```
{{args}} | parallel -X {{command}}
```

- Break `stdin` into ~1M blocks, feed each block to `stdin` of new command:

```
cat {{big_file.txt}} | parallel --pipe --block 1M {{command}}
```

- Run on multiple machines via SSH:

```
parallel {{[-S|--sshlogin]}} {{machine1}},{{machine2}}
{{command}} ::: {{arg1}} {{arg2}}
```

- Download 4 files simultaneously from a text file containing links showing progress:

```
parallel {{[-j|--jobs]}} 4 --bar --eta wget {{[-q|--quote]}}
{} :::: {{path/to/links.txt}}
```

- Print the jobs which `parallel` is running in `stderr`:

```
parallel {{[-t|--verbose]}} {{command}} ::: {{args}}
```

parquet-tools

Show, inspect and manipulate Parquet file.

More information: https://github.com/apache/parquet-mr.

- Display the content of a Parquet file:

```
parquet-tools cat {{path/to/parquet}}
```

- Display the first few lines of a Parquet file:

```
parquet-tools head {{path/to/parquet}}
```

- Print the schema of a Parquet file:

```
parquet-tools schema {{path/to/parquet}}
```

- Print the metadata of a Parquet file:

```
parquet-tools meta {{path/to/parquet}}
```

- Print the content and metadata of a Parquet file:

```
parquet-tools dump {{path/to/parquet}}
```

- Concatenate several Parquet files into the target one:

```
parquet-tools merge {{path/to/parquet1}} {{path/to/parquet2}}
{{path/to/target_parquet}}
```

- Print the count of rows in a Parquet file:

```
parquet-tools rowcount {{path/to/parquet}}
```

- Print the column and offset indexes of a Parquet file:

```
parquet-tools column-index {{path/to/parquet}}
```

particle

Interact with Particle devices.

More information: https://docs.particle.io/tutorials/developer-tools/cli.

- Log in or create an account for the Particle CLI:

  ```
  particle setup
  ```

- Display a list of devices:

  ```
  particle list
  ```

- Create a new Particle project interactively:

  ```
  particle project create
  ```

- Compile a Particle project:

  ```
  particle compile {{device_type}} {{path/to/source_code.ino}}
  ```

- Update a device to use a specific app remotely:

  ```
  particle flash {{device_name}} {{path/to/program.bin}}
  ```

- Update a device to use the latest firmware via serial:

  ```
  particle flash --serial {{path/to/firmware.bin}}
  ```

- Execute a function on a device:

  ```
  particle call {{device_name}} {{function_name}}
  {{function_arguments}}
  ```

pass otp

A pass extension for managing one-time-password (OTP) tokens.

More information: https://manned.org/pass-otp.

- Prompt for an otpauth URI token and create a new pass file:

```
pass otp insert {{path/to/pass}}
```

- Prompt for an otpauth URI token and append to an existing pass file:

```
pass otp append {{path/to/pass}}
```

- Print a 2FA code using the OTP token in a pass file:

```
pass otp {{path/to/pass}}
```

- Copy and don't print a 2FA code using the OTP token in a pass file:

```
pass otp {{[-c|--clip]}} {{path/to/pass}}
```

- Display a QR code using the OTP token stored in a pass file:

```
pass otp uri {{[-q|--qrcode]}} {{path/to/pass}}
```

- Prompt for an OTP secret value specifying issuer and account (at least one must be specified) and append to existing pass file:

```
pass otp append {{[-s|--secret]}} {{[-i|--issuer]}} {{issuer_name}} {{[-a|--account]}} {{account_name}} {{path/to/pass}}
```

pass

Store and read passwords or other sensitive data.

All data is GPG-encrypted, and managed with a Git repository.

More information: https://www.passwordstore.org.

- Initialize (or re-encrypt) the storage using one or more GPG IDs:

```
pass init {{gpg_id_1}} {{gpg_id_2}}
```

- Save a new password and additional information (press `<Ctrl d>` on a new line to complete):

```
pass insert {{[-m|--multiline]}} {{path/to/data}}
```

- Edit an entry:

```
pass edit {{path/to/data}}
```

- Copy a password (first line of the data file) to the clipboard:

```
pass {{[-c|--clip]}} {{path/to/data}}
```

- List the whole store tree:

```
pass
```

- Generate a new random password with a given length, and copy it to the clipboard:

```
pass generate {{[-c|--clip]}} {{path/to/data}} {{num}}
```

- Initialize a new Git repository (any changes done by pass will be committed automatically):

```
pass git init
```

- Run a Git command on behalf of the password storage:

```
pass git {{command}}
```

passwd

Change a user's password.

More information: https://manned.org/passwd.

- Change the password of the current user interactively:

```
passwd
```

- Change the password of a specific user:

```
sudo passwd {{username}}
```

- Get the current status of the user:

```
passwd {{[-S|--status]}}
```

- Make the password of the account blank (it will set the named account passwordless):

```
passwd {{[-d|--delete]}}
```

- Set password programmatically (ideal for install scripts):

```
yes {{password}} | passwd
```

paste

Merge lines of files.

More information: https://www.gnu.org/software/coreutils/manual/html_node/paste-invocation.html.

- Join all the lines into a single line, using TAB as delimiter:

```
paste {{[-s|--serial]}} {{path/to/file}}
```

- Join all the lines into a single line, using the specified delimiter:

```
paste {{[-sd|--serial --delimiters]}} {{delimiter}} {{path/to/file}}
```

- Merge two files side by side, each in its column, using TAB as delimiter:

```
paste {{path/to/file1}} {{path/to/file2}}
```

- Merge two files side by side, each in its column, using the specified delimiter:

```
paste {{[-d|--delimiters]}} {{delimiter}} {{path/to/file1}} {{path/to/file2}}
```

- Merge two files, with lines added alternatively:

```
paste {{[-d|--delimiters]}} '\n' {{path/to/file1}} {{path/to/file2}}
```

pastel

Generate, analyze, convert and manipulate colors.

More information: https://github.com/sharkdp/pastel.

- Convert colors from one format to another. Here from RGB to HSL:

```
pastel format {{hsl}} {{ff8000}}
```

- Show and analyze colors on the terminal:

```
pastel color "{{rgb(255,50,127)}}"
```

- Pick a color from somewhere on the screen:

```
pastel pick
```

- Generate a set of n visually distinct colors:

```
pastel distinct {{n}}
```

- List all X11/CSS color names:

```
pastel list
```

patator

A multi-purpose brute-forcer, with a modular design and a flexible usage.

More information: https://github.com/lanjelot/patator/wiki/Usage.

- Brute force ssh login with rate limit and timeout options (successful login will show login banner or something similar):

```
patator ssh_login host={{ip_or_host}} user=FILE0
password=FILE1 0={{path/to/users.txt}} 1={{path/to/
passwords.txt}} --rate_limit={{seconds}} --
timeout={{seconds}} -x ignore:mesg='Authentication failed.'
```

- Brute force encrypted zip file:

```
patator unzip_pass zipfile={{path/to/file.zip}}
password=FILE0 0={{path/to/passwords.txt}} -x ignore:code!=0
```

- Brute force http basic auth (payload file `userpass.txt` should be in the format `username:password`):

```
patator http_fuzz url={{http://host:port}} auth_type=basic
user_pass=COMBO00:COMBO01 0={{path/to/userpass.txt}} -x
ignore:code=401
```

- Brute force FTP/FTPS login:

```
patator ftp_login host={{ip_or_host}} user=FILE0
password=FILE1 0={{path/to/users.txt}} 1={{path/to/
passwords.txt}} tls={{0|1}} -x ignore:mesg='Login incorrect.'
-x ignore,reset,retry:code=500
```

- List all available modules:

```
patator --help
```

- Display help for a particular module:

```
patator {{module_name}} --help
```

patch

Patch a file (or files) with a diff file.

Note that diff files should be generated by the **`diff`** command.

More information: https://manned.org/patch.

- Apply a patch using a diff file (filenames must be included in the diff file):

```
patch < {{patch.diff}}
```

- Apply a patch to a specific file:

```
patch < {{patch.diff}} {{path/to/file}}
```

- Patch a file writing the result to a different file:

```
patch < {{patch.diff}} {{path/to/input_file}} {{[-o|--output]}} {{path/to/output_file}}
```

- Apply a patch to the current directory:

```
patch < {{patch.diff}} {{[-p|--strip]}} 1
```

- Apply the reverse of a patch:

```
patch < {{patch.diff}} {{[-R|--reverse]}}
```

patchwork

Render an image of a **squareified treemap** network graph from a **graphviz** file.

Layouts: **dot**, **neato**, **twopi**, **circo**, **fdp**, **sfdp**, **osage** & **patchwork**.

More information: https://graphviz.org/doc/info/command.html.

- Render a PNG image with a filename based on the input filename and output format (uppercase -O):

```
patchwork -T {{png}} -O {{path/to/input.gv}}
```

- Render a SVG image with the specified output filename (lowercase -o):

```
patchwork -T {{svg}} -o {{path/to/image.svg}} {{path/to/input.gv}}
```

- Render the output in PS, PDF, SVG, Fig, PNG, GIF, JPEG, JSON, or DOT format:

```
patchwork -T {{format}} -O {{path/to/input.gv}}
```

- Render a `gif` image using `stdin` and `stdout`:

```
echo "{{digraph {this -> that} }}" | patchwork -T {{gif}} > {{path/to/image.gif}}
```

- Display help:

```
patchwork -?
```

pathchk

Check the validity and portability of pathnames.

More information: https://www.gnu.org/software/coreutils/manual/html_node/pathchk-invocation.html.

- Check pathnames for validity in the current system:

```
pathchk {{path1 path2 ...}}
```

- Check pathnames for validity on a wider range of POSIX compliant systems:

```
pathchk -p {{path1 path2 ...}}
```

- Check pathnames for validity on all POSIX compliant systems:

```
pathchk {{[-p -P|--portability]}} {{path1 path2 ...}}
```

- Only check for empty pathnames or leading dashes (-):

```
pathchk -P {{path1 path2 ...}}
```

pax

Archiving and copying utility.

More information: https://manned.org/pax.1p.

- List the contents of an archive:

```
pax -f {{archive.tar}}
```

- List the contents of a `gzip` archive:

```
pax -zf {{archive.tar.gz}}
```

- Create an archive from files:

```
pax -wf {{target.tar}} {{path/to/file1 path/to/file2 ...}}
```

- Create an archive from files, using output redirection:

```
pax -w {{path/to/file1 path/to/file2 ...}} > {{target.tar}}
```

- Extract an archive into the current directory:

```
pax -rf {{source.tar}}
```

- Copy to a directory, while keeping the original metadata; `target/` must exist:

```
pax -rw {{path/to/file1}} {{path/to/directory1 path/to/
directory2 ...}} {{target/}}
```

pbmclean

Clean up a PBM image by erasing isolated black and white pixels.

More information: https://netpbm.sourceforge.net/doc/pbmclean.html.

- Clean up a PBM image by erasing isolated black and white pixels:

```
pbmclean {{path/to/image.pbm}} > {{path/to/output.pbm}}
```

- Clean up only black/white pixels:

```
pbmclean -{{black|white}} {{path/to/image.pbm}} > {{path/to/output.pbm}}
```

- Specify the minimum number of neighbouring pixels of the same color in order for a pixel not to be considered isolated:

```
pbmclean {{[-m|-minneighbours]}} {{3}} {{path/to/image.pbm}} > {{path/to/output.pbm}}
```

pbmlife

Apply Conway's Rules of Life to a PBM image.

More information: https://netpbm.sourceforge.net/doc/pbmlife.html.

- Apply the Rules of Life to an input PBM image file for one generation and the output the result as a PBM image file:

```
pbmlife {{path/to/file.pbm}}
```

- Display version:

```
pbmlife {{[-v|-version]}}
```

pbmmake

Create a blank bitmap.

More information: https://netpbm.sourceforge.net/doc/pbmmake.html.

- Create a blank bitmap of the specified dimensions:

```
pbmmake {{width}} {{height}} > {{path/to/output_file.pbm}}
```

- Specify the color of the created bitmap:

```
pbmmake -{{white|black|grey}} {{width}} {{height}} > {{path/to/output_file.pbm}}
```

pbmmask

Create a mask bitmap from a regular bitmap.

See also: **pambackground**.

More information: https://netpbm.sourceforge.net/doc/pbmmask.html.

- Create a mask bitmap separating background from foreground:

```
pbmmask {{path/to/image.pbm}} > {{path/to/output.pbm}}
```

- Expand the generated mask by one pixel:

```
pbmmask {{[-r|-expand]}} {{path/to/image.pbm}} > {{path/to/output.pbm}}
```

pbmnoise

Generate white noise.

More information: https://netpbm.sourceforge.net/doc/pbmnoise.html.

- Generate a PGM image containing white noise:

```
pbmnoise {{width}} {{height}} > {{path/to/output.pbm}}
```

- Specify the seed for the pseudo-random number generator:

```
pbmnoise {{width}} {{height}} -randomseed {{value}} > {{path/to/output.pbm}}
```

- Specify the desired rate of white to black pixels:

```
pbmnoise {{width}} {{height}} -ratio {{1/3}} > {{path/to/output.pbm}}
```

pbmpage

Generate a test pattern for printing.

More information: https://netpbm.sourceforge.net/doc/pbmpage.html.

- Generate a test pattern for printing onto US standard paper:

```
pbmpage > {{path/to/file.pbm}}
```

- Generate a test pattern for printing onto A4 paper:

```
pbmpage -a4 > {{path/to/file.pbm}}
```

- Specify the pattern to use:

```
pbmpage {{1|2|3}} > {{path/to/file.pbm}}
```

pbmpscale

Enlarge a PBM image with edge smoothing.

See also: **`pamenlarge`**.

More information: https://netpbm.sourceforge.net/doc/pbmpscale.html.

- Enlarge a PBM image by the specified factor with edge smoothing:

```
pbmpscale {{n}} {{path/to/image.pbm}} > {{path/to/file.pbm}}
```

pbmreduce

Proportionally reduce a PBM image.

See also: **pamenlarge**, **pamditherbw**.

More information: https://netpbm.sourceforge.net/doc/pbmreduce.html.

- Reduce the specified image by the specified factor:

```
pbmreduce {{n}} {{path/to/image.pbm}} > {{path/to/
output.pbm}}
```

- Use simple thresholding when reducing:

```
pbmreduce {{[-t|-threshold]}} {{n}} {{path/to/image.pbm}} >
{{path/to/output.pbm}}
```

- Use the specified threshold for all quantizations:

```
pbmreduce {{[-va|-value]}} {{0.6}} {{n}} {{path/to/
image.pbm}} > {{path/to/output.pbm}}
```

pbmtext

Render text as a PBM image.

See also: **pbmtextps**.

More information: https://netpbm.sourceforge.net/doc/pbmtext.html.

- Render a single line of text as a PBM image:

```
pbmtext "{{Hello World!}}" > {{path/to/output.pbm}}
```

- Render multiple lines of text as a PBM image:

```
echo "{{Hello\nWorld!}}" | pbmtext > {{path/to/output.pbm}}
```

- Render text using a custom font supplied as a PBM file:

```
pbmtext {{[-f|-font]}} {{path/to/font.pbm}} "{{Hello
World!}}" > {{path/to/output.pbm}}
```

- Specify the number of pixels between characters and lines:

```
echo "{{Hello\nWorld!}}" | pbmtext {{[-s|-space]}} {{3}} {{[-
ls|-lspace]}} {{10}} > {{path/to/output.pbm}}
```

pbmtextps

Render text as a PBM image using PostScript.

See also: **pbmtext**.

More information: https://netpbm.sourceforge.net/doc/pbmtextps.html.

- Render a single line of text as a PBM image:

```
pbmtextps "{{Hello World!}}" > {{path/to/output.pbm}}
```

- Specify the font and font size:

```
pbmtextps -font {{Times-Roman}} -fontsize {{30}} "{{Hello World!}}" > {{path/to/output.pbm}}
```

- Specify the desired left and top margins:

```
pbmtextps {{[-l|-leftmargin]}} {{70}} {{[-t|-topmargin]}} {{162}} "{{Hello World!}}" > {{path/to/output.pbm}}
```

- Do not output the rendered text as a PBM image, but a PostScript program that would create this image:

```
pbmtextps {{[-du|-dump-ps]}} "{{Hello World!}}" > {{path/to/output.ps}}
```

pbmto4425

Display a PBM image on an AT&T 4425 terminal.

See also: **`ppmtoterm`**, **`pbmtoascii`**.

More information: https://netpbm.sourceforge.net/doc/pbmto4425.html.

- Display a PBM image on an AT&T 4425 terminal using the terminal's mosaic graphics character set:

```
pbmto4425 {{path/to/image.pbm}}
```

pbmtoascii

Convert a PBM image to ASCII graphics.

See also: **`ppmtoascii`**, **`asciitopgm`**, **`ppmtoterm`**.

More information: https://netpbm.sourceforge.net/doc/pbmtoascii.html.

- Read a PBM file as input and produce an ASCII output:

```
pbmtoascii {{path/to/input_file.pbm}}
```

- Read a PBM file as input and save an ASCII output into a file:

```
pbmtoascii {{path/to/input_file.pbm}} > {{path/to/output_file}}
```

- Read a PBM file as input while setting the pixel mapping (defaults to 1x2):

```
pbmtoascii -{{1x2|2x4}} {{path/to/input_file.pbm}}
```

- Display version:

```
pbmtoascii {{[-v|-version]}}
```

pbmtoatk

Convert a PBM image to an Andrew Toolkit raster object.

See also: **atktopbm**.

More information: https://netpbm.sourceforge.net/doc/pbmtoatk.html.

- Convert a PBM image to an Andrew Toolkit raster object:

```
pbmtoatk {{path/to/image.pbm}} > {{path/to/output.atk}}
```

pbmtobbnbg

Convert a PBM image to a BitGraph graphic.

More information: https://netpbm.sourceforge.net/doc/pbmtobbnbg.html.

- Convert a PBM image to a BitGraph terminal Display Pixel Data sequence:

```
pbmtobbnbg < {{path/to/image.pbm}} > {{path/to/output.dpd}}
```

- Specify the rasterop:

```
pbmtobbnbg < {{path/to/image.pbm}} {{3}} > {{path/to/
output.dpd}}
```

pbmtocmuwm

Convert a PBM image to a CMU window manager bitmap.

See also: **`cmuwmtopbm`**.

More information: https://netpbm.sourceforge.net/doc/pbmtocmuwm.html.

- Convert a PBM image to a CMU window manager bitmap:

```
pbmtocmuwm {{path/to/image.pbm}} > {{path/to/output.bmp}}
```

pbmtoepsi

Convert a PBM image to an encapsulated PostScript style preview bitmap.

More information: https://netpbm.sourceforge.net/doc/pbmtoepsi.html.

- Convert a PBM image to an encapsulated PostScript style preview bitmap:

```
pbmtoepsi {{path/to/image.pbm}} > {{path/to/output.bmp}}
```

- Produce a quadratic output image with the specified resolution:

```
pbmtoepsi {{[-d|-dpi]}} {{144}} {{path/to/image.pbm}} >
{{path/to/output.bmp}}
```

- Produce an output image with the specified horizontal and vertical resolution:

```
pbmtoepsi {{[-d|-dpi]}} {{72x144}} {{path/to/image.pbm}} >
{{path/to/output.bmp}}
```

- Only create a boundary box:

```
pbmtoepsi {{[-b|-bbonly]}} {{path/to/image.pbm}} > {{path/to/
output.bmp}}
```

pbmtoepson

Convert a PBM image to an Epson printer graphic.

See also: **pbmtoescp2**.

More information: https://netpbm.sourceforge.net/doc/pbmtoepson.html.

- Convert a PBM image to an Epson printer graphic:

```
pbmtoepson {{path/to/image.pbm}} > {{path/to/output.epson}}
```

- Specify the printer protocol of the output:

```
pbmtoepson {{[-pr|-protocol]}} {{escp9|escp}} {{path/to/
image.pbm}} > {{path/to/output.epson}}
```

- Specify the horizontal DPI of the output:

```
pbmtoepson {{[-d|-dpi]}} {{60|72|80|90|120|144|240}} {{path/
to/image.pbm}} > {{path/to/output.epson}}
```

pbmtoescp2

Convert a PBM image to a ESC/P2 printer file.

See also: **pbmtoepson**, **escp2topbm**.

More information: https://netpbm.sourceforge.net/doc/pbmtoescp2.html.

- Convert a PBM image to a ESC/P2 printer file:

```
pbmtoescp2 {{path/to/image.pbm}} > {{path/to/output.escp2}}
```

- Specify the compression of the output:

```
pbmtoescp2 {{[-c|-compression]}} {{0|1}} {{path/to/
image.pbm}} > {{path/to/output.escp2}}
```

- Specify the horizontal and vertical resolution of the output in dots per inch:

```
pbmtoescp2 {{[-re|-resolution]}} {{180|360|720}} {{path/to/
image.pbm}} > {{path/to/output.escp2}}
```

- Place a formfeed command at the end of the output:

```
pbmtoescp2 {{[-f|-formfeed]}} {{path/to/image.pbm}} > {{path/
to/output.escp2}}
```

pbmtogem

Read a PBM image as input and produce a compressed GEM .img file as output.

pbmtogem cannot compress repeated lines.

More information: https://netpbm.sourceforge.net/doc/pbmtogem.html.

- Convert a PBM image into a GEM .img file:

```
pbmtogem {{path/to/file.pbm}} > {{path/to/file.img}}
```

- Suppress all informational messages:

```
pbmtogem {{[-q|-quiet]}}
```

- Display version:

```
pbmtogem {{[-v|-version]}}
```

pbmtogo

Convert a PBM image to a compressed GraphOn graphic.

More information: https://netpbm.sourceforge.net/doc/pbmtogo.html.

- Convert a PBM image to a compressed GraphOn graphic:

```
pbmtogo {{path/to/image.pbm}} > {{path/to/output.go}}
```

pbmtoicon

This command has been superseded by **pbmtosunicon**.

More information: https://netpbm.sourceforge.net/doc/pbmtoicon.html.

- View documentation for the current command:

```
tldr pbmtosunicon
```

pbmtolj

Convert a PBM file to an HP LaserJet file.

More information: https://netpbm.sourceforge.net/doc/pbmtolj.html.

- Convert a PBM file to an HP LaserJet file:

```
pbmtolj {{path/to/input.pbm}} > {{path/to/output.lj}}
```

- Compress the output file using the specified method:

```
pbmtolj -{{packbits|delta|compress}} {{path/to/input.pbm}} >
{{path/to/output.lj}}
```

- Specify the required resolution:

```
pbmtolj {{[-r|-resolution]}} {{75|100|150|300|600}} {{path/
to/input.pbm}} > {{path/to/output.lj}}
```

pbmtomacp

Convert a PBM image to a MacPaint file.

See also: **macptopbm**.

More information: https://netpbm.sourceforge.net/doc/pbmtomacp.html.

- Convert a PBM image to a MACP file:

```
pbmtomacp {{path/to/image.pbm}} > {{path/to/output.macp}}
```

- Do not compress the output file:

```
pbmtomacp {{[-n|-norle]}} {{path/to/image.pbm}} > {{path/to/output.macp}}
```

pbmtomda

Convert a PBM image to a Microdesign MDA file.

See also: **`mdatopbm`**.

More information: https://netpbm.sourceforge.net/doc/pbmtomda.html.

- Convert a PBM image to a MDA file:

```
pbmtomda {{path/to/image.pbm}} > {{path/to/output.mda}}
```

- Invert the colors in the input image:

```
pbmtomda -i {{path/to/image.pbm}} > {{path/to/output.mda}}
```

- Halve the input image's height:

```
pbmtomda -d {{path/to/image.pbm}} > {{path/to/output.mda}}
```

pbmtomgr

Convert a PBM image into a MGR bitmap.

See also: **mgrtopbm**.

More information: https://netpbm.sourceforge.net/doc/pbmtomgr.html.

- Convert a PBM image into a MGR bitmap:

```
pbmtomgr {{path/to/image.pbm}} > {{path/to/output.mgr}}
```

pbmtonokia

Convert a PBM image to one of Nokia's Smart Messaging Formats .

More information: https://netpbm.sourceforge.net/doc/pbmtonokia.html.

- Convert a PBM image into a Nokia Operator Logo as hexcode:

```
pbmtonokia {{[-f|-fmt]}} NEX_NOL -net
{{network_operator_code}} {{path/to/image.pbm}} > {{path/to/
output.hex}}
```

- Convert a PBM image into a Nokia Group Graphic as hexcode:

```
pbmtonokia {{[-f|-fmt]}} NEX_NGG {{path/to/image.pbm}} >
{{path/to/output.hex}}
```

- Convert a PBM image into a Nokia Picture Message with the specified text as hexcode:

```
pbmtonokia {{[-f|-fmt]}} NEX_NPM -txt {{text_message}}
{{path/to/image.pbm}} > {{path/to/output.hex}}
```

- Convert a PBM image into a Nokia Operator Logo as a NOL file:

```
pbmtonokia {{[-f|-fmt]}} NOL {{path/to/image.pbm}} > {{path/
to/output.nol}}
```

- Convert a PBM image into a Nokia Group Graphic as an NGG file:

```
pbmtonokia {{[-f|-fmt]}} NGG {{path/to/image.pbm}} > {{path/
to/output.ngg}}
```

- Convert a PBM image into a Nokia Picture Message as an NPM file:

```
pbmtonokia {{[-f|-fmt]}} NPM {{path/to/image.pbm}} > {{path/
to/output.npm}}
```

pbmtopgm

Convert a PBM image to PGM by averaging areas surrounding individual pixels.

See also: **pnmconvol**, **pamditherbw**.

More information: https://netpbm.sourceforge.net/doc/pbmtopgm.html.

- Convert PBM image to PGM by averaging the wxh-sized area surrounding each pixel:

```
pbmtopgm {{w}} {{h}} {{path/to/image.pbm}} > {{path/to/output.pgm}}
```

pbmtopi3

Convert a PBM image to an Atari Degas PI3 image.

See also: **`pi3topbm`**.

More information: https://netpbm.sourceforge.net/doc/pbmtopi3.html.

- Convert a PBM image to an Atari Degas PI3 image:

```
pbmtopi3 {{path/to/image.pbm}} > {{path/to/atari_image.pi3}}
```

pbmtoplot

Convert a PBM image into a UNIX plot file.

More information: https://netpbm.sourceforge.net/doc/pbmtoplot.html.

- Convert a PBM image into a UNIX plot file:

```
pbmtoplot {{path/to/image.pbm}} > {{path/to/output.plot}}
```

pbmtoppa

Convert a PBM image to HP Printer Performance Architecture format.

More information: https://netpbm.sourceforge.net/doc/pbmtoppa.html.

- Convert a PBM image into a PPA file:

```
pbmtoppa {{path/to/image.pbm}} > {{path/to/output.ppa}}
```

- Specify the desired dots-per-inch and paper size:

```
pbmtoppa -d {{300}} -s {{a4}} {{path/to/image.pbm}} > {{path/
to/output.ppa}}
```

pbmtoptx

Convert a PBM image into Printronix printer graphics.

More information: https://netpbm.sourceforge.net/doc/pbmtoptx.html.

- Convert a PBM image into a PTX file:

```
pbmtoptx {{path/to/image.pbm}} > {{path/to/output.ptx}}
```

pbmtosunicon

Convert a PBM image into a Sun icon.

More information: https://netpbm.sourceforge.net/doc/pbmtosunicon.html.

- Convert a PBM image into a Sun icon:

```
pbmtosunicon {{path/to/input.pbm}} > {{path/to/output.ico}}
```

pbmtowbmp

Convert a PBM image to a wireless bitmap file.

More information: https://netpbm.sourceforge.net/doc/pbmtowbmp.html.

- Convert a PBM image to a WBMP file:

```
pbmtowbmp {{path/to/input_file.pbm}} > {{path/to/
output_file.wbmp}}
```

pbmtox10bm

This command has been superseded by **`pbmtoxbm -x10`**.

More information: https://netpbm.sourceforge.net/doc/pbmtox10bm.html.

- View documentation for the current command:

```
tldr pbmtoxbm
```

pbmtoxbm

Convert a PBM image to a X11 or X10 bitmap.

More information: https://netpbm.sourceforge.net/doc/pbmtoxbm.html.

- Convert a PBM image to a X11 XBM file:

```
pbmtoxbm {{path/to/input_file.pbm}} > {{path/to/
output_file.xbm}}
```

- Explicitly specify whether an X11 or X10 bitmap should be generated:

```
pbmtoxbm -{{x11|x10}} {{path/to/input_file.pbm}} > {{path/to/
output_file.xbm}}
```

pbmtoybm

Convert a PBM file to a Bennet Yee "face" file.

More information: https://netpbm.sourceforge.net/doc/pbmtoybm.html.

- Convert a PBM image file to YBM:

```
pbmtoybm {{path/to/input_file.pbm}} > {{path/to/
output_file.ybm}}
```

pbmtozinc

Convert a PBM image to a Zinc bitmap as used by the Zinc Interface Library Version 1.0.

More information: https://netpbm.sourceforge.net/doc/pbmtozinc.html.

- Convert a PBM image to a Zinc bitmap:

```
pbmtozinc {{path/to/image.pbm}} > {{path/to/output.zinc}}
```

pbmupc

Generate a PBM image of a Universal Product Code (UPC).

More information: https://netpbm.sourceforge.net/doc/pbmupc.html.

- Generate a UPC image for the specified product type, manufacturer code, and product code:

```
pbmupc {{product_type}} {{manufacturer_code}}
{{product_code}} > {{path/to/output.pbm}}
```

- Use an alternative style that does not display the checksum:

```
pbmupc -s2 {{product_type}} {{manufacturer_code}}
{{product_code}} > {{path/to/output.pbm}}
```

pbzip2

A parallel implementation of the **bzip2** file compressor.

See also: **bzip2**, **tar**.

More information: https://manned.org/pbzip2.

- Compress a file:

```
pbzip2 {{path/to/file}}
```

- Compress a file using the specified number of processors:

```
pbzip2 -p{{4}} {{path/to/file}}
```

- Compress in combination with tar (options can be passed to pbzip2):

```
tar -cf {{path/to/compressed_file}}.tar.bz2 {{[-I|--use-
compress-program]}} "pbzip2 {{-option1 -
option2 ...}}" {{path/to/file}}
```

- Decompress a file:

```
pbzip2 {{[-d|--decompress]}} {{path/to/compressed_file.bz2}}
```

- Display help:

```
pbzip2 {{[-h|--help]}}
```

pcapfix

Repair damaged or corrupted PCAP and PcapNG files.

More information: https://f00l.de/pcapfix/.

- Repair a PCAP/PCapNG file (Note: For PCAP files, only the first 262144 bytes of each packet are scanned):

```
pcapfix {{path/to/file.pcapng}}
```

- Repair an entire PCAP file:

```
pcapfix --deep-scan {{path/to/file.pcap}}
```

- Repair a PCAP/PcapNG file and write the repaired file to the specified location:

```
pcapfix --outfile {{path/to/repaired.pcap}} {{path/to/
file.pcap}}
```

- Treat the specified file as a PcapNG file, ignoring automatic recognition:

```
pcapfix --pcapng {{path/to/file.pcapng}}
```

- Repair a file and show the process in detail:

```
pcapfix --verbose {{path/to/file.pcap}}
```

pcdindex

This command has been renamed to **`pcdovtoppm`**.

More information: https://netpbm.sourceforge.net/doc/pcdindex.html.

- View documentation for the command under its current name:

```
tldr pcdovtoppm
```

pcdovtoppm

Create an index image for a photo CD based on its overview file.

More information: https://netpbm.sourceforge.net/doc/pcdovtoppm.html.

- Create a PPM index image from a PCD overview file:

```
pcdovtoppm {{path/to/file.pcd}} > {{path/to/output.ppm}}
```

- Specify the maximum width of the output image and the maximum size of each of the images contained in the output:

```
pcdovtoppm {{[-m|-maxwidth]}} {{width}} {{[-s|-size]}}
{{size}} {{path/to/file.pcd}} > {{path/to/output.ppm}}
```

- Specify the maximum number of images across and the maximum number of colours:

```
pcdovtoppm {{[-a|-across]}} {{n_images}} {{[-c|-colors]}}
{{n_colours}} {{path/to/file.pcd}} > {{path/to/output.ppm}}
```

- Use the specified font for annotations and paint the background white:

```
pcdovtoppm {{[-f|-font]}} {{font}} {{[-w|-white]}} {{path/to/
file.pcd}} > {{path/to/output.ppm}}
```

pcxtoppm

Convert a PCX file to a PPM image.

More information: https://netpbm.sourceforge.net/doc/pcxtoppm.html.

- Convert a PCX file to a PPM image:

```
pcxtoppm {{path/to/file.pcx}} > {{path/to/file.ppm}}
```

- Use a predefined standard palette even if the PCX file provides one:

```
pcxtoppm {{[-s|-stdpalette]}} {{path/to/file.pcx}} > {{path/to/file.ppm}}
```

- Print information on the PCX header to `stdout`:

```
pcxtoppm {{[-verb|-verbose]}} {{path/to/file.pcx}} > {{path/to/file.ppm}}
```

pdf-parser

Identify fundamental elements of a PDF file without rendering it.

More information: https://blog.didierstevens.com/programs/pdf-tools.

- Display statistics for a PDF file:

```
pdf-parser {{[-a|--stats]}} {{path/to/file.pdf}}
```

- Display objects of a specific type (`/Font`, `/URI`, ...) in a PDF file:

```
pdf-parser {{[-t|--type]}} {{/object_type}} {{path/to/
file.pdf}}
```

- Search for strings in indirect objects:

```
pdf-parser {{[-s|--search]}} {{search_string}} {{path/to/
file.pdf}}
```

pdffonts

Portable Document Format (PDF) file fonts information viewer.

More information: https://www.xpdfreader.com/pdffonts-man.html.

- Print PDF file fonts information:

```
pdffonts {{path/to/file.pdf}}
```

- Specify user password for PDF file to bypass security restrictions:

```
pdffonts -upw {{password}} {{path/to/file.pdf}}
```

- Specify owner password for PDF file to bypass security restrictions:

```
pdffonts -opw {{password}} {{path/to/file.pdf}}
```

- Print additional information on location of the font that will be used when the PDF file is rasterized:

```
pdffonts -loc {{path/to/file.pdf}}
```

- Print additional information on location of the font that will be used when the PDF file is converted to PostScript:

```
pdffonts -locPS {{path/to/file.pdf}}
```

pdfgrep

Search text in PDF files.

More information: https://pdfgrep.org/doc.html.

- Find lines that match pattern in a PDF:

```
pdfgrep {{pattern}} {{file.pdf}}
```

- Include file name and page number for each matched line:

```
pdfgrep {{[-H|--with-filename]}} {{[-n|--page-number]}}
{{pattern}} {{file.pdf}}
```

- Do a case-insensitive search for lines that begin with `file_name` and return the first 3 matches:

```
pdfgrep {{[-m|--max-count]}} {{3}} {{[-i|--ignore-case]}}
'{{^file_name}}' {{file.pdf}}
```

- Find pattern in files with a `.pdf` extension in the current directory recursively:

```
pdfgrep {{[-r|--recursive]}} {{pattern}}
```

- Find pattern on files that match a specific glob in the current directory recursively:

```
pdfgrep {{[-r|--recursive]}} --include
'{{*book.pdf}}' {{pattern}}
```

pdfimages

Utility for extracting images from PDFs.

More information: https://manned.org/pdfimages.

- Extract all images from a PDF file and save them as PNGs:

```
pdfimages -png {{path/to/file.pdf}} {{filename_prefix}}
```

- Extract images from pages 3 to 5:

```
pdfimages -f {{3}} -l {{5}} {{path/to/file.pdf}}
{{filename_prefix}}
```

- Extract images from a PDF file and include the page number in the output filenames:

```
pdfimages -p {{path/to/file.pdf}} {{filename_prefix}}
```

- List information about all the images in a PDF file:

```
pdfimages -list {{path/to/file.pdf}}
```

pdfinfo

Portable Document Format (PDF) file information viewer.

More information: https://www.xpdfreader.com/pdfinfo-man.html.

- Print PDF file information:

```
pdfinfo {{path/to/file.pdf}}
```

- Specify user password for PDF file to bypass security restrictions:

```
pdfinfo -upw {{password}} {{path/to/file.pdf}}
```

- Specify owner password for PDF file to bypass security restrictions:

```
pdfinfo -opw {{password}} {{path/to/file.pdf}}
```

pdfjam

Shell frontend for the LaTeX pdfpages package for mingling PDFs.

More information: https://github.com/pdfjam/pdfjam/blob/master/doc/pdfjam-help.txt.

- Merge two (or more) PDFs:

```
pdfjam {{path/to/file1.pdf path/to/file2.pdf ...}} {{[-o|--outfile]}} {{path/to/output_file.pdf}}
```

- Merge the first page of each file together:

```
pdfjam {{path/to/file1.pdf 1 path/to/file2.pdf 1 ...}} {{[-o|--outfile]}} {{path/to/output_file.pdf}}
```

- Merge subranges from two PDFs:

```
pdfjam {{path/to/file1.pdf 3-5,1}} {{path/to/file2.pdf 4-6}} {{[-o|--outfile]}} {{path/to/output_file.pdf}}
```

- Sign an A4 page (adjust delta to height for other formats) with a scanned signature by overlaying them:

```
pdfjam {{path/to/file.pdf}} {{path/to/signature}} --fitpaper true {{[-o|--outfile]}} {{path/to/signed.pdf}} --nup "{{1x2}}" --delta "{{0 -842pt}}"
```

- Arrange the pages from the input file into a fancy 2x2 grid:

```
pdfjam {{path/to/file.pdf}} --nup {{2x2}} --suffix {{4up}} --preamble '{{\usepackage{fancyhdr} \pagestyle{fancy}}}'
```

- Reverse the order of pages within each given file and concatenate them:

```
pdfjam {{path/to/file1.pdf last-1 path/to/file2.pdf last-1 ...}} --suffix {{reversed}}
```

pdfjoin

PDF merging utility based on pdfjam.

More information: https://github.com/rrthomas/pdfjam-extras.

- Merge two PDFs into one with the default suffix "joined":

```
pdfjoin {{path/to/file1.pdf}} {{path/to/file2.pdf}}
```

- Merge the first page of each given file together:

```
pdfjoin {{path/to/file1.pdf path/to/file2.pdf ...}} {{1}} --
outfile {{output_file}}
```

- Save pages 3 to 5 followed by page 1 to a new PDF with custom suffix:

```
pdfjoin {{path/to/file.pdf}} {{3-5,1}} --suffix
{{rearranged}}
```

- Merge page subranges from two PDFs:

```
pdfjoin /{{path/to/file1.pdf}} {{2-}} {{file2}} {{last-3}} --
outfile {{output_file}}
```

pdflatex

Compile a PDF document from LaTeX source files.

More information: https://manned.org/pdflatex.

- Compile a PDF document:

```
pdflatex {{source.tex}}
```

- Compile a PDF document specifying an output directory:

```
pdflatex -output-directory={{path/to/directory}}
{{source.tex}}
```

- Compile a PDF document, exiting on each error:

```
pdflatex -halt-on-error {{source.tex}}
```

pdfposter

Convert a large-sheeted PDF into multiple A4 pages for printing.

More information: https://pdfposter.readthedocs.io.

- Convert an A2 poster into 4 A4 pages:

```
pdfposter --poster-size a2 {{input_file.pdf}}
{{output_file.pdf}}
```

- Scale an A4 poster to A3 and then generate 2 A4 pages:

```
pdfposter --scale 2 {{input_file.pdf}} {{output_file.pdf}}
```

pdfseparate

Portable Document Format (PDF) file page extractor.

More information: https://manpages.debian.org/latest/poppler-utils/pdfseparate.1.en.html.

- Extract pages from PDF file and make a separate PDF file for each page:

```
pdfseparate {{path/to/source_filename.pdf}} {{path/to/destination_filename-%d.pdf}}
```

- Specify the first/start page for extraction:

```
pdfseparate -f {{3}} {{path/to/source_filename.pdf}} {{path/to/destination_filename-%d.pdf}}
```

- Specify the last page for extraction:

```
pdfseparate -l {{10}} {{path/to/source_filename.pdf}} {{path/to/destination_filename-%d.pdf}}
```

pdftex

Compile a PDF document from TeX source files.

More information: https://www.tug.org/applications/pdftex/.

- Compile a PDF document:

```
pdftex {{source.tex}}
```

- Compile a PDF document, specifying an output directory:

```
pdftex -output-directory={{path/to/directory}} {{source.tex}}
```

- Compile a PDF document, exiting on each error:

```
pdftex -halt-on-error {{source.tex}}
```

pdftk

PDF toolkit.

More information: https://www.pdflabs.com/docs/pdftk-man-page/.

- Extract pages 1-3, 5 and 6-10 from a PDF file and save them as another one:

```
pdftk {{input.pdf}} cat {{1-3 5 6-10}} output {{output.pdf}}
```

- Merge (concatenate) a list of PDF files and save the result as another one:

```
pdftk {{file1.pdf file2.pdf ...}} cat output {{output.pdf}}
```

- Split each page of a PDF file into a separate file, with a given filename output pattern:

```
pdftk {{input.pdf}} burst output {{out_%d.pdf}}
```

- Rotate all pages by 180 degrees clockwise:

```
pdftk {{input.pdf}} cat {{1-endsouth}} output {{output.pdf}}
```

- Rotate third page by 90 degrees clockwise and leave others unchanged:

```
pdftk {{input.pdf}} cat {{1-2 3east 4-end}} output
{{output.pdf}}
```

pdftocairo

Convert PDF files to PNG/JPEG/TIFF/PDF/PS/EPS/SVG using cairo.

More information: https://manned.org/pdftocairo.

- Convert a PDF file to JPEG:

```
pdftocairo {{path/to/file.pdf}} -jpeg
```

- Convert to PDF expanding the output to fill the paper:

```
pdftocairo {{path/to/file.pdf}} {{output.pdf}} -pdf -expand
```

- Convert to SVG specifying the first/last page to convert:

```
pdftocairo {{path/to/file.pdf}} {{output.svg}} -svg -f
{{first_page}} -l {{last_page}}
```

- Convert to PNG with 200ppi resolution:

```
pdftocairo {{path/to/file.pdf}} {{output.png}} -png -r 200
```

- Convert to grayscale TIFF setting paper size to A3:

```
pdftocairo {{path/to/file.pdf}} -tiff -gray -paper A3
```

- Convert to PNG cropping x and y pixels from the top-left corner:

```
pdftocairo {{path/to/file.pdf}} -png -x {{x_pixels}} -y
{{y_pixels}}
```

pdftotext

Convert PDF files to plain text format.

More information: https://www.xpdfreader.com/pdftotext-man.html.

- Convert `filename.pdf` to plain text and print it to `stdout`:

```
pdftotext {{filename.pdf}} -
```

- Convert `filename.pdf` to plain text and save it as `filename.txt`:

```
pdftotext {{filename.pdf}}
```

- Convert `filename.pdf` to plain text and preserve the layout:

```
pdftotext -layout {{filename.pdf}}
```

- Convert `input.pdf` to plain text and save it as `output.txt`:

```
pdftotext {{input.pdf}} {{output.txt}}
```

- Convert pages 2, 3 and 4 of `input.pdf` to plain text and save them as `output.txt`:

```
pdftotext -f {{2}} -l {{4}} {{input.pdf}} {{output.txt}}
```

pdfunite

PDF merging utility.

More information: https://github.com/mtgrosser/pdfunite.

- Merge 2 PDFs into a single PDF:

```
pdfunite {{path/to/fileA.pdf}} {{path/to/fileB.pdf}} {{path/to/merged_output.pdf}}
```

- Merge a directory of PDFs into a single PDF:

```
pdfunite {{path/to/directory/*.pdf}} {{path/to/merged_output.pdf}}
```

pecl

Manage PHP extensions from the PHP Extension Community Library.

More information: https://manned.org/pecl.

- Display a list of available commands:

```
pecl
```

- Install a package:

```
pecl install {{package}}
```

- List installed packages:

```
pecl list
```

- Upgrade all packages:

```
pecl upgrade
```

- Upgrade a specific package:

```
pecl upgrade {{package}}
```

- Uninstall a specific package:

```
pecl uninstall {{package}}
```

- Display information about a specific package:

```
pecl info {{package}}
```

- Update the available package sources:

```
pecl update-channels
```

peco

Interactive filtering tool.

More information: https://github.com/peco/peco.

- Start `peco` on all files in the specified directory:

```
find {{path/to/directory}} -type f | peco
```

- Start `peco` for running processes:

```
ps aux | peco
```

- Start `peco` with a specified query:

```
peco --query "{{query}}"
```

pee

Tee **`stdin`** to pipes.

See also: **`tee`**.

More information: https://manned.org/pee.

- Run each command, providing each one with a distinct copy of `stdin`:

```
pee {{command1 command2 ...}}
```

- Write a copy of `stdin` to `stdout` (like `tee`):

```
pee cat {{command1 command2 ...}}
```

- Immediately terminate upon SIGPIPEs and write errors:

```
pee --no-ignore-sigpipe --no-ignore-write-errors {{command1 command2 ...}}
```

peerflix

Stream video- or audio-based torrents to a media player.

More information: https://github.com/mafintosh/peerflix.

- Stream the largest media file in a torrent:

```
peerflix "{{torrent_url|magnet_link}}"
```

- List all streamable files contained in a torrent (given as a magnet link):

```
peerflix "{{magnet:?xt=urn:btih:
0123456789abcdef0123456789abcdef01234567}}" {{[-l|--list]}}
```

- Stream the largest file in a torrent, given as a torrent URL, to VLC:

```
peerflix "{{http://example.net/music.torrent}}" {{[-v|--
vlc]}}
```

- Stream the largest file in a torrent to MPlayer, with subtitles:

```
peerflix "{{torrent_url|magnet_link}}" {{[-m|--mplayer]}}
{{[-t|--subtitles]}} {{subtitle-file.srt}}
```

- Stream all files from a torrent to Airplay:

```
peerflix "{{torrent_url|magnet_link}}" {{[-a|--all]}} {{[-
s|--airplay]}}
```

peerindex

Inspect MRT TABLE_DUMPV2 Peer Index Table.

Can read files compressed with **gzip**, **bzip2** and **xz**.

More information: https://codeberg.org/1414codeforge/ubgpsuite.

- List all peers:

  ```
  peerindex {{master6.mrt}}
  ```

- Display all peers that have provided routing information:

  ```
  peerindex {{[-r|--only-refs]}} {{master6.mrt}}
  ```

peludna-prognoza

Fetch pollen measurement data for Croatian cities from your terminal using Pliva's allergies data API.

More information: https://github.com/vladimyr/peludna-prognoza.

- Start an interactive search for a city and fetch data for it:

  ```
  peludna-prognoza
  ```

- Fetch data for a city:

  ```
  peludna-prognoza "{{city}}"
  ```

- Display data in a machine-readable format:

  ```
  peludna-prognoza "{{city}}" --{{json|xml}}
  ```

- Display the pollen measurement page for a city at https://plivazdravlje.hr in the default web browser:

  ```
  peludna-prognoza "{{city}}" {{[-w|--web]}}
  ```

perl-rename

This command is an alias of **rename**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common rename
```

perl

The Perl 5 language interpreter.

See also: **`perldoc`**.

More information: https://perldoc.perl.org/perl.

- Print lines from `stdin` [m]atching `regex1` and case [i]nsensitive `regex2`:

```
perl -n -e 'print if m/regex1/ and m/regex2/i'
```

- Say ([-E]) first match group, using a `regex`, ignoring space ([x]) in `regex`:

```
perl -n -E 'say $1 if m/{{before}} ( {{group_regex}} ) {{after}}/x'
```

- [i]n-place, with backup, [s]ubstitute all occurrence ([g]) of `regex` with a replacement:

```
perl -i'.bak' -p -e 's/regex/{{replacement}}/g' {{path/to/files}}
```

perlbrew

Manage Perl installations in the home directory.

See also: **`asdf`**.

More information: https://github.com/gugod/App-perlbrew.

- Initialize a `perlbrew` environment:

```
perlbrew init
```

- List available Perl versions:

```
perlbrew available
```

- Install/uninstall a Perl version:

```
perlbrew {{install|uninstall}} {{version}}
```

- List perl installations:

```
perlbrew list
```

- Switch to an installation and set it as default:

```
perlbrew switch perl-{{version}}
```

- Use the system Perl again:

```
perlbrew off
```

- List installed CPAN modules for the installation in use:

```
perlbrew list-modules
```

- Clone CPAN modules from one installation to another:

```
perlbrew clone-modules {{source_installation}}
{{destination_installation}}
```

perldoc

Look up Perl documentation in `.pod` format.

More information: https://perldoc.perl.org/perldoc.

- View documentation for a builtin [f]unction, a [v]ariable or an [a]PI:

```
perldoc -{{f|v|a}} {{name}}
```

- Search in the [q]uestion headings of Perl FAQ:

```
perldoc -q {{regex}}
```

- Send output directly to `stdout` (by default, it is send to a pager):

```
perldoc -T {{page|module|program|URL}}
```

- Specify the language code of the desired translation:

```
perldoc -L {{language_code}} {{page|module|program|URL}}
```

pest

A PHP testing framework with a focus on simplicity.

More information: https://pestphp.com/docs/cli-api-reference.

- Initialize a standard Pest configuration in the current directory:

```
pest --init
```

- Run tests in the current directory:

```
pest
```

- Run tests annotated with the given group:

```
pest --group {{name}}
```

- Run tests and print the coverage report to `stdout`:

```
pest --coverage
```

- Run tests with coverage and fail if the coverage is less than the minimum percentage:

```
pest --coverage --min={{80}}
```

- Run tests in parallel:

```
pest --parallel
```

- Run tests with mutations:

```
pest --mutate
```

pfetch

Display system information.

More information: https://github.com/dylanaraps/pfetch.

- Display the ASCII art and default fields:

```
pfetch
```

- Display only the ASCII art and color palette fields:

```
PF_INFO="{{ascii palette}}" pfetch
```

- Display all possible fields:

```
PF_INFO="{{ascii title os host kernel uptime pkgs memory
shell editor wm de palette}}" pfetch
```

- Display a different username and hostname:

```
USER="{{user}}" HOSTNAME="{{hostname}}" pfetch
```

- Display without colors:

```
PF_COLOR={{0}} pfetch
```

pg_amcheck

Check for corruption in one or more PostgreSQL databases.

More information: https://www.postgresql.org/docs/current/app-pgamcheck.html.

- Check a specific database:

```
pg_amcheck {{dbname}}
```

- Check all databases:

```
pg_amcheck {{[-a|--all]}}
```

- Check databases matching the specified pattern:

```
pg_amcheck {{[-d|--database]}} {{pattern}}
```

- Check specific tables within a database:

```
pg_amcheck {{[-t|--table]}} {{pattern}} {{dbname}}
```

- Check specific schemas within a database:

```
pg_amcheck {{[-s|--schema]}} {{pattern}} {{dbname}}
```

- Display help:

```
pg_amcheck {{[-?|--help]}}
```

pg_archivecleanup

Remove old WAL archive files in PostgreSQL.

More information: https://www.postgresql.org/docs/current/pgarchivecleanup.html.

- Clean an archive directory up to a given WAL file:

```
pg_archivecleanup {{path/to/archive}} {{path/to/walfile}}
```

- Perform a dry run (list files that would be removed without actually doing it):

```
pg_archivecleanup {{[-n|--dry-run]}} {{path/to/archive}} {{path/to/walfile}}
```

- Strip a file extension before deciding deletion:

```
pg_archivecleanup {{[-x|--strip-extension]}} {{extension}} {{path/to/archive}} {{path/to/walfile}}
```

- Remove backup history files too:

```
pg_archivecleanup {{[-b|--clean-backup-history]}} {{path/to/archive}} {{path/to/walfile}}
```

- Enable debug logging output:

```
pg_archivecleanup {{[-d|--debug]}} {{path/to/archive}} {{path/to/walfile}}
```

pg_basebackup

Take a base backup of a running PostgreSQL cluster.

Used for full or incremental backups, point-in-time recovery, or setting up replication standbys.

More information: https://www.postgresql.org/docs/current/app-pgbasebackup.html.

- Take a base backup from a remote PostgreSQL server:

```
pg_basebackup {{[-h|--host]}} {{host}} {{[-D|--pgdata]}} {{path/to/backup_dir}}
```

- Take a backup with progress shown:

```
pg_basebackup {{[-h|--host]}} {{host}} {{[-D|--pgdata]}} {{path/to/backup_dir}} {{[-P|--progress]}}
```

- Create a compressed backup (`gzip`) in tar format:

```
pg_basebackup {{[-D|--pgdata]}} {{path/to/backup_dir}} {{[-F|--format]}} {{[t|tar]}} {{[-z|--gzip]}}
```

- Create an incremental backup using a previous manifest file:

```
pg_basebackup {{[-D|--pgdata]}} {{path/to/backup_dir}} {{[-i|--incremental]}} {{path/to/old_manifest}}
```

- Write a recovery configuration for setting up a standby:

```
pg_basebackup {{[-D|--pgdata]}} {{path/to/backup_dir}} {{[-R|--write-recovery-conf]}}
```

- Relocate a tablespace during backup:

```
pg_basebackup {{[-D|--pgdata]}} {{path/to/backup_dir}} {{[-T|--tablespace-mapping]}} {{path/to/old_tablespace}}={{path/to/new_tablespace}}
```

- Limit transfer rate to reduce server load:

```
pg_basebackup {{[-D|--pgdata]}} {{path/to/backup_dir}} {{[-r|--max-rate]}} {{100M}}
```

- Stream WAL logs while taking the backup:

```
pg_basebackup {{[-D|--pgdata]}} {{path/to/backup_dir}} {{[-
X|--wal-method]}} stream
```

pg_checksums

Enable, disable, or check data checksums in a PostgreSQL database cluster.

More information: https://www.postgresql.org/docs/current/app-pgchecksums.html.

- Check data checksums in a database cluster:

```
pg_checksums {{[-D|--pgdata]}} {{path/to/data}}
```

- Enable data checksums in a database cluster:

```
pg_checksums {{[-e|--enable]}} {{[-D|--pgdata]}} {{path/to/data}}
```

- Disable data checksums in a database cluster:

```
pg_checksums {{[-d|--disable]}} {{[-D|--pgdata]}} {{path/to/data}}
```

- Check data checksums with verbose output:

```
pg_checksums {{[-v|--verbose]}} {{[-D|--pgdata]}} {{path/to/data}}
```

- Check data checksums showing progress:

```
pg_checksums {{[-P|--progress]}} {{[-D|--pgdata]}} {{path/to/data}}
```

- Display help:

```
pg_checksums {{[-?|--help]}}
```

pg_combinebackup

Reconstruct a full (synthetic) PostgreSQL backup from an incremental backup chain.

When specifying multiple backups order them from oldest to newest.

More information: https://www.postgresql.org/docs/current/app-pgcombinebackup.html.

- Combine a full and incremental backup into one synthetic full backup:

```
pg_combinebackup {{path/to/full_backup}} {{path/to/incremental_backup}} {{[-o|--output]}} {{path/to/output_directory}}
```

- Perform a dry run to show what would be done, without creating files:

```
pg_combinebackup {{[-n|--dry-run]}} {{path/to/full_backup}} {{path/to/incremental_backup}} {{[-o|--output]}} {{path/to/output_directory}}
```

- Use hard links instead of copying files (faster, same filesystem required):

```
pg_combinebackup {{[-k|--link]}} {{path/to/full_backup}} {{path/to/incremental_backup}} {{[-o|--output]}} {{path/to/output_directory}}
```

- Use file cloning (reflinks) for efficient copy if supported:

```
pg_combinebackup --clone {{path/to/full_backup}} {{path/to/incremental_backup}} {{[-o|--output]}} {{path/to/output_directory}}
```

- Use the `copy_file_range` system call for efficient copying:

```
pg_combinebackup --copy-file-range {{path/to/full_backup}} {{path/to/incremental_backup}} {{[-o|--output]}} {{path/to/output_directory}}
```

- Relocate a tablespace during reconstruction:

```
pg_combinebackup {{path/to/backup1 path/to/backup2 ...}} {{[-T|--tablespace-mapping]}} /{{path/to/old_tablespace}}=/{{path/to/new_tablespace}} {{[-o|--output]}} {{path/to/output_directory}}
```

- Disable fsync for faster but unsafe writes (testing only):

```
pg_combinebackup {{[-N|--no-sync]}} {{path/to/backup1 path/
to/backup2 ...}} {{[-o|--output]}} {{path/to/
output_directory}}
```

- Show detailed debug output:

```
pg_combinebackup {{[-d|--debug]}} {{path/to/backup1 path/to/
backup2 ...}} {{[-o|--output]}} {{path/to/output_directory}}
```

pg_config

Display information about the installed PostgreSQL version.

More information: https://www.postgresql.org/docs/current/app-pgconfig.html.

- Show the directory where PostgreSQL binaries are installed:

```
pg_config --bindir
```

- Show the directory for library files:

```
pg_config --libdir
```

- Show the directory for include files (C headers):

```
pg_config --includedir
```

- Show the directory for shared data files:

```
pg_config --sharedir
```

- Show the directory for locale data:

```
pg_config --localedir
```

- Display help:

```
pg_config {{[-?|--help]}}
```

- Display version:

```
pg_config --version
```

pg_controldata

Display control information of a PostgreSQL database cluster.

More information: https://www.postgresql.org/docs/current/app-pgcontroldata.html.

- Display control information for a specific data directory:

```
pg_controldata {{[-D|--pgdata]}} {{path/to/data}}
```

- Display help:

```
pg_controldata {{[-?|--help]}}
```

- Display version:

```
pg_controldata {{[-V|--version]}}
```

pg_createsubscriber

Convert a physical replica into a new logical replica.

More information: https://www.postgresql.org/docs/current/app-pgcreatesubscriber.html.

- Convert a physical replica to a logical replica for a specific database:

```
pg_createsubscriber {{[-d|--database]}} {{dbname}} {{[-D|--pgdata]}} {{path/to/data}} {{[-P|--publisher-server]}} {{connstr}}
```

- Perform a dry run without modifying the target directory:

```
pg_createsubscriber {{[-n|--dry-run]}} {{[-d|--database]}} {{dbname}} {{[-D|--pgdata]}} {{path/to/data}} {{[-P|--publisher-server]}} {{connstr}}
```

- Enable two-phase commit for the subscription:

```
pg_createsubscriber {{[-T|--enable-two-phase]}} {{[-d|--database]}} {{dbname}} {{[-D|--pgdata]}} {{path/to/data}} {{[-P|--publisher-server]}} {{connstr}}
```

- Convert with verbose output:

```
pg_createsubscriber {{[-v|--verbose]}} {{[-d|--database]}} {{dbname}} {{[-D|--pgdata]}} {{path/to/data}} {{[-P|--publisher-server]}} {{connstr}}
```

- Display help:

```
pg_createsubscriber {{[-?|--help]}}
```

pg_ctl

Utility for controlling a PostgreSQL server and database cluster.

More information: https://www.postgresql.org/docs/current/app-pg-ctl.html.

- Initialize a new PostgreSQL database cluster:

```
pg_ctl {{[-D|--pgdata]}} {{data_directory}} init
```

- Start a PostgreSQL server:

```
pg_ctl {{[-D|--pgdata]}} {{data_directory}} start
```

- Stop a PostgreSQL server:

```
pg_ctl {{[-D|--pgdata]}} {{data_directory}} stop
```

- Restart a PostgreSQL server:

```
pg_ctl {{[-D|--pgdata]}} {{data_directory}} restart
```

- Reload the PostgreSQL server configuration:

```
pg_ctl {{[-D|--pgdata]}} {{data_directory}} reload
```

pg_dump

Extract a PostgreSQL database into a script file or other archive file.

More information: https://www.postgresql.org/docs/current/app-pgdump.html.

- Dump database into an SQL-script file:

```
pg_dump {{db_name}} > {{output_file.sql}}
```

- Same as above, customize username:

```
pg_dump {{[-U|--username]}} {{username}} {{db_name}} > {{output_file.sql}}
```

- Same as above, customize host and port:

```
pg_dump {{[-h|--host]}} {{host}} {{[-p|--port]}} {{port}} {{db_name}} > {{output_file.sql}}
```

- Dump a database into a custom-format archive file:

```
pg_dump {{[-F|--format]}} {{[c|custom]}} {{db_name}} > {{output_file.dump}}
```

- Dump only database data into an SQL-script file:

```
pg_dump {{[-a|--data-only]}} {{db_name}} > {{path/to/output_file.sql}}
```

- Dump only schema (data definitions) into an SQL-script file:

```
pg_dump {{[-s|--schema-only]}} {{db_name}} > {{path/to/output_file.sql}}
```

pg_dumpall

Extract a PostgreSQL database cluster into a script file or other archive file.

More information: https://www.postgresql.org/docs/current/app-pg-dumpall.html.

- Dump all databases:

```
pg_dumpall > {{path/to/file.sql}}
```

- Dump all databases using a specific username:

```
pg_dumpall {{[-U|--username]}} {{username}} > {{path/to/file.sql}}
```

- Same as above, customize host and port:

```
pg_dumpall {{[-h|--host]}} {{host}} {{[-p|--port]}} {{port}} > {{output_file.sql}}
```

- Dump only database data into an SQL-script file:

```
pg_dumpall {{[-a|--data-only]}} > {{path/to/file.sql}}
```

- Dump only schema (data definitions) into an SQL-script file:

```
pg_dumpall {{[-s|--schema-only]}} > {{output_file.sql}}
```

pg_isready

Check the connection status of a PostgreSQL server.

More information: https://www.postgresql.org/docs/current/app-pg-isready.html.

- Check connection:

```
pg_isready
```

- Check connection with a specific hostname and port:

```
pg_isready {{[-h|--host]}} {{hostname}} {{[-p|--port]}} {{port}}
```

- Check connection displaying a message only when the connection fails:

```
pg_isready {{[-q|--quiet]}}
```

pg_receivewal

Stream the write-ahead log from a running PostgreSQL cluster.

More information: https://www.postgresql.org/docs/current/app-pgreceivewal.html.

- Stream WAL to a local directory (minimum required):

```
pg_receivewal {{[-D|--directory]}} {{directory}}
```

- Same as above, specify host, port, username including verbose output:

```
pg_receivewal {{[-v|--verbose]}} {{[-h|--host]}} {{host}}
{{[-p|--port]}} {{port}} {{[-U|--username]}} {{username}}
{{[-D|--directory]}} {{directory}}
```

- Use replication slot (create-if-needed):

```
pg_receivewal {{[-S|--slot]}} {{slot_name}} --create-slot
{{[-h|--host]}} {{host}} {{[-p|--port]}} {{port}} {{[-U|--
username]}} {{username}} {{[-D|--directory]}} {{directory}}
```

- Stop at a given WAL position (LSN):

```
pg_receivewal {{[-E|--endpos]}} {{lsn}} {{[-D|--directory]}}
{{directory}} {{[-h|--host]}} {{host}} {{[-p|--port]}}
{{port}} {{[-U|--username]}} {{username}}
```

- Control looping on failure:

```
pg_receivewal {{[-n|--no-loop]}} {{[-D|--directory]}}
{{directory}} {{[-h|--host]}} {{host}} {{[-p|--port]}}
{{port}} {{[-U|--username]}} {{username}}
```

- Flush data synchronously (force WAL writes immediately):

```
pg_receivewal --synchronous {{[-D|--directory]}}
{{directory}} {{[-h|--host]}} {{host}} {{[-p|--port]}}
{{port}} {{[-U|--username]}} {{username}}
```

- Compress WAL output (gzip, level 0–9):

```
pg_receivewal {{[-Z|--compress]}} {{level|method}} {{[-D|--
directory]}} {{directory}} {{[-h|--host]}} {{host}} {{[-p|--
port]}} {{port}} {{[-U|--username]}} {{username}}
```

- Set status reporting interval:

```
pg_receivewal {{[-s|--status-interval]}} {{seconds}} {{[-D|--
directory]}} {{directory}} {{[-h|--host]}} {{host}} {{[-p|--
port]}} {{port}} {{[-U|--username]}} {{username}}
```

pg_recvlogical

Control PostgreSQL logical decoding streams.

More information: https://www.postgresql.org/docs/current/app-pgrecvlogical.html.

- Create a new logical replication slot:

```
pg_recvlogical {{[-d|--dbname]}} {{dbname}} {{[-S|--slot]}} {{slot_name}} --create-slot
```

- Start streaming changes from a logical replication slot to a file:

```
pg_recvlogical {{[-d|--dbname]}} {{dbname}} {{[-S|--slot]}} {{slot_name}} --start {{[-f|--file]}} {{filename}}
```

- Drop a logical replication slot:

```
pg_recvlogical {{[-d|--dbname]}} {{dbname}} {{[-S|--slot]}} {{slot_name}} --drop-slot
```

- Create a slot with two-phase commit enabled:

```
pg_recvlogical {{[-d|--dbname]}} {{dbname}} {{[-S|--slot]}} {{slot_name}} --create-slot {{[-t|--enable-two-phase]}}
```

- Display help:

```
pg_recvlogical {{[-?|--help]}}
```

pg_resetwal

Reset the write-ahead log and other control information of a PostgreSQL database cluster.

More information: https://www.postgresql.org/docs/current/app-pgresetwal.html.

- Reset the WAL and control information for a specific data directory:

```
pg_resetwal {{[-D|--pgdata]}} {{path/to/data}}
```

- Perform a dry run:

```
pg_resetwal {{[-D|--pgdata]}} {{path/to/data}} {{[-n|--dry-run]}}
```

- Force the WAL and control information reset:

```
pg_resetwal {{[-D|--pgdata]}} {{path/to/data}} {{[-f|--force]}}
```

- Display help:

```
pg_resetwal {{[-?|--help]}}
```

- Display version:

```
pg_resetwal {{[-V|--version]}}
```

pg_restore

Restore a PostgreSQL database from an archive file created by pg_dump.

More information: https://www.postgresql.org/docs/current/app-pgrestore.html.

- Restore an archive into an existing database:

```
pg_restore {{[-d|--dbname]}} {{db_name}}
{{archive_file.dump}}
```

- Same as above, customize username:

```
pg_restore {{[-U|--username]}} {{username}} {{[-d|--dbname]}}
{{db_name}} {{archive_file.dump}}
```

- Same as above, customize host and port:

```
pg_restore {{[-h|--host]}} {{host}} {{[-p|--port]}} {{port}}
{{[-d|--dbname]}} {{db_name}} {{archive_file.dump}}
```

- List database objects included in the archive:

```
pg_restore {{[-l|--list]}} {{archive_file.dump}}
```

- Clean database objects before creating them:

```
pg_restore {{[-c|--clean]}} {{[-d|--dbname]}} {{db_name}}
{{archive_file.dump}}
```

- Use multiple jobs to do the restoring:

```
pg_restore {{[-j|--jobs]}} {{2}} {{[-d|--dbname]}}
{{db_name}} {{archive_file.dump}}
```

pg_rewind

Synchronize a PostgreSQL data directory with another data directory that was forked from it.

More information: https://www.postgresql.org/docs/current/app-pgrewind.html.

- Synchronize target directory with source directory:

```
pg_rewind {{[-D|--target-pgdata]}} {{path/to/target_data}} --source-pgdata {{path/to/source_data}}
```

- Synchronize target with source server using connection string:

```
pg_rewind {{[-D|--target-pgdata]}} {{path/to/target_data}} --source-server {{connstr}}
```

- Perform a dry run:

```
pg_rewind {{[-D|--target-pgdata]}} {{path/to/target_data}} --source-pgdata {{path/to/source_data}} {{[-n|--dry-run]}}
```

- Show progress during synchronization:

```
pg_rewind {{[-D|--target-pgdata]}} {{path/to/target_data}} --source-pgdata {{path/to/source_data}} {{[-P|--progress]}}
```

- Display help:

```
pg_rewind {{[-?|--help]}}
```

pg_test_fsync

Determine the fastest wal_sync_method on your system.

More information: https://www.postgresql.org/docs/current/pgtestfsync.html.

- Run the default fsync benchmark (5 seconds):

```
pg_test_fsync
```

- Specify a custom test duration:

```
pg_test_fsync {{[-s|--secs-per-test]}} {{seconds}}
```

- Use a specific filename (it must be in same file system that the pg_wal directory is or will be placed in):

```
pg_test_fsync {{[-f|--filename]}} {{path/to/file}}
```

pg_test_timing

Measure timing overhead on your system (ensure clock monotonicity).

More information: https://www.postgresql.org/docs/current/pgtesttiming.html.

- Run the default timing test (3 seconds):

```
pg_test_timing
```

- Run for a custom duration:

```
pg_test_timing {{[-d|--duration]}} {{seconds}}
```

pg_upgrade

Upgrade a PostgreSQL database cluster to a new major version.

More information: https://www.postgresql.org/docs/current/pgupgrade.html.

- Check clusters before upgrading:

```
pg_upgrade {{[-c|--check]}} {{[-b|--old-bindir]}} {{path/to/old_bin}} {{[-B|--new-bindir]}} {{path/to/new_bin}} {{[-d|--old-datadir]}} {{path/to/old_data}} {{[-D|--new-datadir]}} {{path/to/new_data}}
```

- Perform the actual upgrade:

```
pg_upgrade {{[-b|--old-bindir]}} {{path/to/old_bin}} {{[-B|--new-bindir]}} {{path/to/new_bin}} {{[-d|--old-datadir]}} {{path/to/old_data}} {{[-D|--new-datadir]}} {{path/to/new_data}}
```

- Use multiple parallel jobs during the upgrade:

```
pg_upgrade {{[-j|--jobs]}} {{njobs}} {{[-b|--old-bindir]}} {{path/to/old_bin}} {{[-B|--new-bindir]}} {{path/to/new_bin}} {{[-d|--old-datadir]}} {{path/to/old_data}} {{[-D|--new-datadir]}} {{path/to/new_data}}
```

- Specify the old and new PostgreSQL ports:

```
pg_upgrade {{[-p|--old-port]}} {{port}} {{[-P|--new-port]}} {{port}} {{[-b|--old-bindir]}} {{path/to/old_bin}} {{[-B|--new-bindir]}} {{path/to/new_bin}} {{[-d|--old-datadir]}} {{path/to/old_data}} {{[-D|--new-datadir]}} {{path/to/new_data}}
```

- Use hard links instead of copying files to the new cluster:

```
pg_upgrade {{[-k|--link]}} {{[-b|--old-bindir]}} {{path/to/old_bin}} {{[-B|--new-bindir]}} {{path/to/new_bin}} {{[-d|--old-datadir]}} {{path/to/old_data}} {{[-D|--new-datadir]}} {{path/to/new_data}}
```

- Display help:

```
pg_upgrade {{[-?|--help]}}
```

pg_verifybackup

Verify the integrity of a base backup of a PostgreSQL cluster.

More information: https://www.postgresql.org/docs/current/app-pgverifybackup.html.

- Verify a backup stored in a specific directory:

```
pg_verifybackup {{path/to/backup}}
```

- Verify a backup showing progress information:

```
pg_verifybackup {{[-P|--progress]}} {{path/to/backup}}
```

- Verify a backup and exit immediately on first error:

```
pg_verifybackup {{[-e|--exit-on-error]}} {{path/to/backup}}
```

- Verify a backup ignoring specific files or directories:

```
pg_verifybackup {{[-i|--ignore]}} {{path/to/
file_or_directory}} {{path/to/backup}}
```

- Verify a backup with a manifest file in a different location:

```
pg_verifybackup {{[-m|--manifest-path]}} {{path/to/
backup_manifest}} {{path/to/backup}}
```

- Display help:

```
pg_verifybackup {{[-?|--help]}}
```

pg_waldump

Display a human-readable rendering of the write-ahead log (WAL) of a PostgreSQL database cluster.

More information: https://www.postgresql.org/docs/current/pgwaldump.html.

- Display WAL records from a specific segment:

```
pg_waldump {{start_segment}}
```

- Display WAL records between two segments:

```
pg_waldump {{start_segment}} {{end_segment}}
```

- Specify the WAL file directory:

```
pg_waldump {{start_segment}} {{end_segment}} {{[-p|--path]}} {{path}}
```

- Follow new WAL entries as they arrive:

```
pg_waldump {{start_segment}} {{end_segment}} {{[-f|--follow]}}
```

- Limit number of records shown:

```
pg_waldump {{start_segment}} {{end_segment}} {{[-n|--limit]}} {{count}}
```

- Display summary statistics instead of individual records:

```
pg_waldump {{start_segment}} {{end_segment}} {{[-z|--stats]}}
```

- Filter by resource manager:

```
pg_waldump {{start_segment}} {{end_segment}} {{[-r|--rmgr]}} {{rmgr_name}}
```

- Display help:

```
pg_waldump {{[-?|--help]}}
```

pg_walsummary

Print contents of WAL summary files.

More information: https://www.postgresql.org/docs/current/app-pgwalsummary.html.

- Convert a WAL summary file to text:

```
pg_walsummary {{path/to/file}}
```

- Print one line per individual modified block (rather than ranges):

```
pg_walsummary {{[-i|--individual]}} {{path/to/file}}
```

- Suppress normal output (only errors):

```
pg_walsummary {{[-q|--quiet]}} {{path/to/file}}
```

pgbench

Run a benchmark test on PostgreSQL.

More information: https://www.postgresql.org/docs/current/pgbench.html.

- Initialize a database with a scale factor of 50 times the default size:

```
pgbench --initialize --scale={{50}} {{database_name}}
```

- Benchmark a database with 10 clients, 2 worker threads, and 10,000 transactions per client:

```
pgbench --client={{10}} --jobs={{2}} --transactions={{10000}}
{{database_name}}
```

pgcli

A modern PostgreSQL CLI with auto-completion and syntax highlighting.

More information: https://www.pgcli.com/#usage.

- Connect to a PostgreSQL database using a connection string:

```
pgcli postgresql://{{user}}@{{host}}/{{database}}
```

- Connect to a database using flags:

```
pgcli {{[-h|--host]}} {{host}} {{[-U|--username]}}
{{username}} {{[-d|--dbname]}} {{database}}
```

- Display help:

```
pgcli --help
```

pgmbentley

Bentleyize a PGM image.

More information: https://netpbm.sourceforge.net/doc/pgmbentley.html.

- Apply the Bentley Effect on a PGM image:

```
pgmbentley {{path/to/input_file.pgm}} > {{path/to/
output_file.pgm}}
```

pgmcrater

This command has been superseded by **`pamcrater`**, **`pamshadedrelief`**, and **`pamtopnm`**.

More information: https://netpbm.sourceforge.net/doc/pgmcrater.html.

- View documentation for `pamcrater`:

```
tldr pamcrater
```

- View documentation for `pamshadedrelief`:

```
tldr pamshadedrelief
```

- View documentation for `pamtopnm`:

```
tldr pamtopnm
```

pgmdeshadow

Deshadow a PGM image.

More information: https://netpbm.sourceforge.net/doc/pgmdeshadow.html.

- Remove grey shadows from a PGM image:

```
pgmdeshadow {{path/to/input_file.pgm}} > {{path/to/
output_file.pgm}}
```

pgmedge

This command has been superseded by **pamedge**.

More information: https://netpbm.sourceforge.net/doc/pgmedge.html.

- View documentation for the current command:

```
tldr pamedge
```

pgmenhance

Edge-enhance a PGM image.

See also: **pamedge**.

More information: https://netpbm.sourceforge.net/doc/pgmenhance.html.

- Edge-enhance a PGM image:

```
pgmenhance {{path/to/image.pgm}} > {{path/to/output.pgm}}
```

- Specify the level of enhancement:

```
pgmenhance -{{1..9}} {{path/to/image.pgm}} > {{path/to/
output.pgm}}
```

pgmhist

Print a histogram of the values present in a PGM image.

See also: **`ppmhist`**.

More information: https://netpbm.sourceforge.net/doc/pgmhist.html.

- Display the histogram for human reading:

```
pgmhist {{path/to/image.pgm}}
```

- Display the median grey value:

```
pgmhist {{[-me|-median]}} {{path/to/image.pgm}}
```

- Display four quartile grey value:

```
pgmhist {{[-qua|-quartile]}} {{path/to/image.pgm}}
```

- Report the existence of invalid grey values:

```
pgmhist {{[-f|-forensic]}} {{path/to/image.pgm}}
```

- Display machine-readable output:

```
pgmhist {{[-ma|-machine]}} {{path/to/image.pgm}}
```

pgmkernel

Generate a convolution kernel to be used with **pnmconvol**.

See also: **pnmconvol**.

More information: https://netpbm.sourceforge.net/doc/pgmkernel.html.

- Generate a convolution kernel:

```
pgmkernel {{width}} {{height}} > {{path/to/output.pgm}}
```

- Generate a quadratic convolution kernel:

```
pgmkernel {{size}} > {{path/to/output.pgm}}
```

- Specify the weight of the center in the generated kernel:

```
pgmkernel {{[-w|-weight]}} {{value}} {{width}} {{height}} >
{{path/to/output.pgm}}
```

pgmmake

Create PGM image with a uniform gray level.

More information: https://netpbm.sourceforge.net/doc/pgmmake.html.

- Create PGM image with a uniform gray level (specified as a number between 0 and 1) and the specified dimensions:

```
pgmmake {{graylevel}} {{width}} {{height}} > {{path/to/output_file.pgm}}
```

pgmnoise

Generate white noise.

More information: https://netpbm.sourceforge.net/doc/pgmnoise.html.

- Generate a PGM image containing white noise:

```
pgmnoise {{width}} {{height}} > {{path/to/output.pgm}}
```

- Specify the seed for the pseudo-random number generator:

```
pgmnoise {{width}} {{height}} -randomseed {{value}} > {{path/to/output.pgm}}
```

pgmnorm

This command has been superseded by **`pnmnorm`**.

More information: https://netpbm.sourceforge.net/doc/pgmnorm.html.

- View documentation for the current command:

```
tldr pnmnorm
```

pgmoil

This command has been superseded by **`pamoil`**.

More information: https://netpbm.sourceforge.net/doc/pgmoil.html.

- View documentation for the current command:

```
tldr pamoil
```

pgmramp

Generate a greyscale map.

More information: https://netpbm.sourceforge.net/doc/pgmramp.html.

- Generate a left-to-right greyscale map:

```
pgmramp -lr > {{path/to/output.pgm}}
```

- Generate a top-to-bottom greyscale map:

```
pgmramp -tb > {{path/to/output.pgm}}
```

- Generate a rectangular greyscale map:

```
pgmramp -rectangle > {{path/to/output.pgm}}
```

- Generate a elliptical greyscale map:

```
pgmramp -ellipse {{path/to/image.pgm}} > {{path/to/
output.pgm}}
```

- Generate a greyscale map from the top-left corner to the bottom-right corner:

```
pgmramp -diagonal {{path/to/image.pgm}} > {{path/to/
output.pgm}}
```

pgmslice

This command has been superseded by **`pamslice`**.

More information: https://netpbm.sourceforge.net/doc/pgmslice.html.

- View documentation for the current command:

```
tldr pamslice
```

pgmtexture

Extract textural features from a PGM image.

More information: https://netpbm.sourceforge.net/doc/pgmtexture.html.

- Extract textural features from a PGM image:

```
pgmtexture {{path/to/image.pgm}} > {{path/to/output.pgm}}
```

- Specify the distance parameter for the feature extraction algorithm:

```
pgmtexture -d {{distance}} {{path/to/image.pgm}} > {{path/to/
output.pgm}}
```

pgmtofs

Convert a PGM image to Usenix FaceSaver format.

See also: **`fstopgm`**.

More information: https://netpbm.sourceforge.net/doc/pgmtofs.html.

- Convert the specified PGM image to Usenix FaceSave format:

```
pgmtofs {{path/to/input.pgm}} > {{path/to/output.fs}}
```

pgmtolispm

Convert a PGM image to Lisp Machine format.

See also: **`lispmtopgm`**.

More information: https://netpbm.sourceforge.net/doc/pgmtolispm.html.

- Convert the specified PGM image to Lisp Machine format:

```
pgmtolispm {{path/to/input.pgm}} > {{path/to/output.lispm}}
```

pgmtopbm

This command has been superseded by **pamditherbw**.

More information: https://netpbm.sourceforge.net/doc/pgmtopbm.html.

- View documentation for the current command:

```
tldr pamditherbw
```

pgmtopgm

Copy a PGM image file.

More information: https://netpbm.sourceforge.net/doc/pgmtopgm.html.

- Copy PGM file from `stdin` to `stderr`:

```
pgmtopgm
```

- Display version:

```
pgmtopgm -version
```

pgmtoppm

Colorize a PGM image.

More information: https://netpbm.sourceforge.net/doc/pgmtoppm.html.

- Map all greyscale values of the input image to all colors between the two specified colors:

```
pgmtoppm {{[-b|-black]}} {{red}} {{[-w|-white]}} {{blue}} {{path/to/input.pgm}} > {{path/to/output.ppm}}
```

- Map all greyscale values of the input image to colors according to the specified colormap:

```
pgmtoppm {{[-m|-map]}} {{path/to/colormap.ppm}} {{path/to/input.pgm}} > {{path/to/output.ppm}}
```

pgmtosbig

Convert a PGM image to the SBIG CCDOPS format.

More information: https://netpbm.sourceforge.net/doc/pgmtosbig.html.

- Convert a PGM image file to the SBIG CCDOPS format:

```
pgmtosbig {{path/to/input_file.pgm}} > {{path/to/
output.sbig}}
```

pgmtost4

Convert a PGM image to the SBIG ST-4 format.

More information: https://netpbm.sourceforge.net/doc/pgmtost4.html.

- Convert a PGM image file to the SBIG ST-4 format:

```
pgmtost4 {{path/to/input_file.pgm}} > {{path/to/output.st4}}
```

pgrep

Find or signal processes by name.

More information: https://manned.org/pgrep.

- Return PIDs of any running processes with a matching command string:

```
pgrep {{process_name}}
```

- Search for processes including their command-line options:

```
pgrep {{[-f|--full]}} "{{process_name}} {{parameter}}"
```

- Search for processes run by a specific user:

```
pgrep {{[-u|--euid]}} root {{process_name}}
```

phan

A static analysis tool for PHP.

More information: https://github.com/phan/phan.

- Generate a `.phan/config.php` in the current directory:

```
phan --init
```

- Generate a Phan configuration file using a specific level (1 being strictest to 5 being the least strict):

```
phan --init --init-level {{level}}
```

- Analyze the current directory:

```
phan
```

- Analyze one or more directories:

```
phan {{--directory path/to/directory1 --directory path/to/directory2 ...}}
```

- Specify a configuration file (defaults to `.phan/config.php`):

```
phan --config-file {{path/to/config.php}}
```

- Specify the output mode:

```
phan --output-mode {{text|verbose|json|csv|codeclimate|checkstyle|pylint|html}}
```

- Specify the number of parallel processes:

```
phan --processes {{number_of_processes}}
```

phing

A PHP build tool based on Apache Ant.

More information: https://www.phing.info/guide/chunkhtml/ch03s03.html.

- Perform the default task in the `build.xml` file:

```
phing
```

- Initialize a new build file:

```
phing {{[-i|--init]}} {{path/to/build.xml}}
```

- Perform a specific task:

```
phing {{task_name}}
```

- Use the given build file path:

```
phing {{[-f|-buildfile]}} {{path/to/build.xml}} {{task_name}}
```

- Log to the given file:

```
phing -logfile {{path/to/log_file}} {{task_name}}
```

- Use custom properties in the build:

```
phing -D{{property}}={{value}} {{task_name}}
```

- Specify a custom listener class:

```
phing -listener {{class_name}} {{task_name}}
```

- Build using verbose output:

```
phing -verbose {{task_name}}
```

phive

The Phar Installation and Verification Environment for secure PHP application deployment.

More information: https://phar.io/#Usage.

- Display a list of available aliased Phars:

```
phive list
```

- Install a specified Phar to the local directory:

```
phive install {{alias|url}}
```

- Install a specified Phar globally:

```
phive install {{alias|url}} {{[-g|--global]}}
```

- Install a specified Phar to a target directory:

```
phive install {{alias|url}} {{[-t|--target]}} {{path/to/directory}}
```

- Update all Phar files to the latest version:

```
phive update
```

- Remove a specified Phar file:

```
phive remove {{alias|url}}
```

- Remove unused Phar files:

```
phive purge
```

- List all available commands:

```
phive help
```

php artisan

Laravel's Artisan command-line interface.

More information: https://laravel.com/docs/artisan.

- Start PHP's built-in web server for the current Laravel application:

```
php artisan serve
```

- Start an interactive PHP command-line interface:

```
php artisan tinker
```

- Generate a new Eloquent model class with a migration, factory and resource controller:

```
php artisan make:model {{ModelName}} --all
```

- Display a list of all available commands:

```
php artisan help
```

php-coveralls

A PHP client for Coveralls.

More information: https://php-coveralls.github.io/php-coveralls/#cli-options.

- Send coverage information to Coveralls:

```
php-coveralls
```

- Send coverage information to Coveralls for a specific directory:

```
php-coveralls {{[-r|--root_dir]}} {{path/to/directory}}
```

- Send coverage information to Coveralls with a specific config:

```
php-coveralls {{[-c|--config]}} {{path/to/.coveralls.yml}}
```

- Send coverage information to Coveralls with verbose output:

```
php-coveralls {{[-v|--verbose]}}
```

- Send coverage information to Coveralls excluding source files with no executable statements:

```
php-coveralls --exclude-no-stmt
```

- Send coverage information to Coveralls with a specific environment name:

```
php-coveralls {{[-e|--env]}} {{test|dev|prod}}
```

- Specify multiple Coverage Clover XML files to upload:

```
php-coveralls {{[-x|--coverage_clover]}} {{path/to/
first_clover.xml}} --coverage_clover {{path/to/
second_clover.xml}}
```

- Output the JSON that will be sent to Coveralls to a specific file:

```
php-coveralls {{[-o|--json_path]}} {{path/to/coveralls-
upload.json}}
```

php-cs-fixer

Automatic coding style fixer for PHP.

More information: https://github.com/FriendsOfPHP/PHP-CS-Fixer.

- Execute code style fixing in the current directory:

```
php-cs-fixer fix
```

- Execute code style fixing for a specific directory:

```
php-cs-fixer fix {{path/to/directory}}
```

- Execute code style linting without applying changes:

```
php-cs-fixer fix --dry-run
```

- Execute code style fixes using specific rules:

```
php-cs-fixer fix --rules={{rules}}
```

- Display the rules that have been applied:

```
php-cs-fixer fix --verbose
```

- Output the results in a different format:

```
php-cs-fixer fix --format={{txt|json|xml|checkstyle|junit|gitlab}}
```

- Display files that require fixing:

```
php-cs-fixer list-files
```

- Describe a rule or ruleset:

```
php-cs-fixer describe {{rule}}
```

php yii

Yii Framework's command-line interface.

More information: https://yiiframework.com.

- Start PHP's built-in web server for the current Yii application:

```
php yii {{serve}}
```

- Generate a controller, views and related files for the CRUD actions on the specified model class:

```
php yii {{gii/crud}} --modelClass={{ModelName}} --
controllerClass={{ControllerName}}
```

- Display help:

```
php yii {{help}}
```

php

PHP command-line interface.

More information: https://www.php.net/manual/en/features.commandline.options.php.

- Parse and execute a PHP script:

```
php {{path/to/file}}
```

- Check syntax on (i.e. [l]int) a PHP script:

```
php {{[-l|--syntax-check]}} {{path/to/file}}
```

- Run PHP inter[a]ctively:

```
php {{[-a|--interactive]}}
```

- Run PHP code (Notes: Don't use `<? ?>` tags; escape double quotes with backslash):

```
php {{[-r|--run]}} "{{code}}"
```

- Start a PHP built-in web server in the current directory:

```
php {{[-S|--server]}} {{host}}:{{port}}
```

- List installed PHP extensions:

```
php {{[-m|--modules]}}
```

- Display information about the current PHP configuration:

```
php {{[-i|--info]}}
```

- Display information about a specific function:

```
php {{[--rf|--rfunction]}} {{function_name}}
```

phpbu

A backup utility framework for PHP.

More information: https://phpbu.de/manual/current/en/phpbu-manual.html#cli.

- Run backups using the default `phpbu.xml` configuration file:

  ```
  phpbu
  ```

- Run backups using a specific configuration file:

  ```
  phpbu --configuration={{path/to/configuration_file.xml}}
  ```

- Only run the specified backups:

  ```
  phpbu --limit={{backup_task_name}}
  ```

- Simulate the actions that would have been performed:

  ```
  phpbu --simulate
  ```

phpcbf

Fix violations detected by phpcs.

More information: https://github.com/squizlabs/PHP_CodeSniffer.

- Fix issues in the specified directory (defaults to the PEAR standard):

```
phpcbf {{path/to/directory}}
```

- Display a list of installed coding standards:

```
phpcbf -i
```

- Specify a coding standard to validate against:

```
phpcbf {{path/to/directory}} --standard {{standard}}
```

- Specify comma-separated file extensions to include when sniffing:

```
phpcbf {{path/to/directory}} --extensions
{{file_extension1,file_extension2,...}}
```

- A comma-separated list of files to load before processing:

```
phpcbf {{path/to/directory}} --bootstrap {{path/to/
file1,path/to/file2,...}}
```

- Don't recurse into subdirectories:

```
phpcbf {{path/to/directory}} -l
```

phpcpd

A copy and paste detector for PHP code.

More information: https://github.com/sebastianbergmann/phpcpd.

- Analyze duplicated code for a specific file or directory:

```
phpcpd {{path/to/file_or_directory}}
```

- Analyze using fuzzy matching for variable names:

```
phpcpd --fuzzy {{path/to/file_or_directory}}
```

- Specify a minimum number of identical lines (defaults to 5):

```
phpcpd --min-lines {{number_of_lines}} {{path/to/
file_or_directory}}
```

- Specify a minimum number of identical tokens (defaults to 70):

```
phpcpd --min-tokens {{number_of_tokens}} {{path/to/
file_or_directory}}
```

- Exclude a directory from analysis (must be relative to the source):

```
phpcpd --exclude {{path/to/excluded_directory}} {{path/to/
file_or_directory}}
```

- Output the results to a PHP-CPD XML file:

```
phpcpd --log-pmd {{path/to/log_file}} {{path/to/
file_or_directory}}
```

phpcs

Tokenize PHP, JavaScript and CSS files to detect violations of a defined set of coding standards.

More information: https://github.com/squizlabs/PHP_CodeSniffer.

- Sniff the specified directory for issues (defaults to the PEAR standard):

```
phpcs {{path/to/directory}}
```

- Display a list of installed coding standards:

```
phpcs -i
```

- Specify a coding standard to validate against:

```
phpcs {{path/to/directory}} --standard {{standard}}
```

- Specify comma-separated file extensions to include when sniffing:

```
phpcs {{path/to/directory}} --extensions
{{file_extension1,file_extension2,...}}
```

- Specify the format of the output report (e.g. `full`, `xml`, `json`, `summary`):

```
phpcs {{path/to/directory}} --report {{format}}
```

- Set configuration variables to be used during the process:

```
phpcs {{path/to/directory}} --config-set {{key}} {{value}}
```

- A comma-separated list of files to load before processing:

```
phpcs {{path/to/directory}} --bootstrap {{path/to/file1,path/
to/file2,...}}
```

- Don't recurse into subdirectories:

```
phpcs {{path/to/directory}} -l
```

phpdox

A PHP documentation generator.

More information: https://phpdox.net.

- Display an annotated skeleton configuration XML file:

```
phpdox --skel
```

- Generate documentation for the current working directory:

```
phpdox
```

- Generate documentation using a specific configuration file:

```
phpdox {{[-f|--file]}} {{path/to/phpdox.xml}}
```

- Only run the metadata collection process:

```
phpdox {{[-c|--collector]}}
```

- Only run the documentation generator process:

```
phpdox {{[-g|--generator]}}
```

phpenv

A PHP version manager for development purposes.

More information: https://github.com/phpenv/phpenv.

- Install a PHP version globally:

```
phpenv install {{version}}
```

- Refresh shim files for all PHP binaries known to `phpenv`:

```
phpenv rehash
```

- List all installed PHP versions:

```
phpenv versions
```

- Display the currently active PHP version:

```
phpenv version
```

- Set the global PHP version:

```
phpenv global {{version}}
```

- Set the local PHP version, which overrides the global version:

```
phpenv local {{version}}
```

- Unset the local PHP version:

```
phpenv local --unset
```

phpize

Prepare a PHP extension for compiling.

More information: https://www.php.net/manual/install.pecl.phpize.

- Prepare the PHP extension in the current directory for compiling:

```
phpize
```

- Delete files previously created by phpize:

```
phpize --clean
```

phploc

Quickly measure the size and analyzing the structure of a PHP project.

More information: https://github.com/sebastianbergmann/phploc.

- Analyze a directory and print the result:

```
phploc {{path/to/directory}}
```

- Include only specific files from a comma-separated list (globs are allowed):

```
phploc {{path/to/directory}} --names '{{path/to/file1,path/to/file2,...}}'
```

- Exclude specific files from a comma-separated list (globs are allowed):

```
phploc {{path/to/directory}} --names-exclude '{{path/to/file1,path/to/file2,...}}'
```

- Exclude a specific directory from analysis:

```
phploc {{path/to/directory}} --exclude {{path/to/exclude_directory}}
```

- Log the results to a specific CSV file:

```
phploc {{path/to/directory}} --log-csv {{path/to/file}}
```

- Log the results to a specific XML file:

```
phploc {{path/to/directory}} --log-xml {{path/to/file}}
```

- Count PHPUnit test case classes and test methods:

```
phploc {{path/to/directory}} --count-tests
```

phpmd

PHP mess detector: check for common potential problems.

More information: https://github.com/phpmd/phpmd.

- Display a list of available rulesets and formats:

```
phpmd
```

- Scan a file or directory for problems using comma-separated rulesets:

```
phpmd {{path/to/file_or_directory}} {{xml|text|html}}
{{ruleset1,ruleset2,...}}
```

- Specify the minimum priority threshold for rules:

```
phpmd {{path/to/file_or_directory}} {{xml|text|html}}
{{ruleset1,ruleset2,...}} --minimumpriority {{priority}}
```

- Include only the specified extensions in analysis:

```
phpmd {{path/to/file_or_directory}} {{xml|text|html}}
{{ruleset1,ruleset2,...}} --suffixes {{extensions}}
```

- Exclude the specified comma-separated directories:

```
phpmd {{path/to/file_or_directory1,path/to/
file_or_directory2,...}} {{xml|text|html}}
{{ruleset1,ruleset2,...}} --exclude {{directory_patterns}}
```

- Output the results to a file instead of `stdout`:

```
phpmd {{path/to/file_or_directory}} {{xml|text|html}}
{{ruleset1,ruleset2,...}} --reportfile {{path/to/
report_file}}
```

- Ignore the use of warning-suppressive PHPDoc comments:

```
phpmd {{path/to/file_or_directory}} {{xml|text|html}}
{{ruleset1,ruleset2,...}} --strict
```

phpspec

A Behaviour Driven Development tool for PHP.

More information: https://phpspec.net.

- Create a specification for a class:

```
phpspec describe {{class_name}}
```

- Run all specifications in the "spec" directory:

```
phpspec run
```

- Run a single specification:

```
phpspec run {{path/to/class_specification_file}}
```

- Run specifications using a specific configuration file:

```
phpspec run {{[-c|--config]}} {{path/to/configuration_file}}
```

- Run specifications using a specific bootstrap file:

```
phpspec run {{[-b|--bootstrap]}} {{path/to/bootstrap_file}}
```

- Disable code generation prompts:

```
phpspec run --no-code-generation
```

- Enable fake return values:

```
phpspec run --fake
```

phpstan

A PHP static analysis tool to discover bugs in code.

More information: https://phpstan.org/user-guide/command-line-usage.

- Analyze one or more directories:

```
phpstan analyse {{path/to/directory1 path/to/directory2 ...}}
```

- Analyze a directory using a configuration file:

```
phpstan analyse {{path/to/directory}} {{[-c|--
configuration]}} {{path/to/config}}
```

- Analyze using a specific rule level (0-10, higher is stricter):

```
phpstan analyse {{path/to/directory}} {{[-l|--level]}}
{{level}}
```

- Specify an autoload file to load before analyzing:

```
phpstan analyse {{path/to/directory}} {{[-a|--autoload-
file]}} {{path/to/autoload_file}}
```

- Specify a memory limit during analysis:

```
phpstan analyse {{path/to/directory}} --memory-limit
{{memory_limit}}
```

- Display available options for analysis:

```
phpstan analyse --help
```

phpstorm

A cross-platform IDE for PHP based on the JetBrains IntelliJ platform.

More information: https://www.jetbrains.com/help/phpstorm/working-with-the-ide-features-from-command-line.html#arguments.

- Open a specific directory:

```
phpstorm {{path/to/directory}}
```

- Open a file:

```
phpstorm {{path/to/file}}
```

- Open a file at a specific line:

```
phpstorm --line {{line_number}} {{path/to/file}}
```

- View the differences between two files:

```
phpstorm diff {{path/to/left_file}} {{path/to/right_file}}
```

phpunit

PHPUnit test runner.

More information: https://phpunit.de.

- Run tests in the current directory. Note: Expects you to have a 'phpunit.xml':

```
phpunit
```

- Run tests in a specific file:

```
phpunit {{path/to/TestFile.php}}
```

- Run tests annotated with the given group:

```
phpunit --group {{name}}
```

- Run tests and generate a coverage report in HTML:

```
phpunit --coverage-html {{path/to/directory}}
```

pi1toppm

Convert an Atari Degas PI1 image to a PPM image.

See also: **ppmtopi1**.

More information: https://netpbm.sourceforge.net/doc/pi1toppm.html.

- Convert an Atari Degas PI1 image into PPM image:

```
pi1toppm {{path/to/atari_image.pi1}} > {{path/to/image.ppm}}
```

pi3topbm

Convert an Atari Degas PI3 image to PBM image.

See also: **pbmtopi3**.

More information: https://netpbm.sourceforge.net/doc/pi3topbm.html.

- Convert an Atari Degas PI3 image to PBM image:

```
pi3topbm {{path/to/atari_image.pi3}} > {{path/to/
output_image.pbm}}
```

piactl

The tool for Private Internet Access, a commercial VPN provider.

More information: https://helpdesk.privateinternetaccess.com/kb/articles/pia-desktop-command-line-interface-part-1.

- Log in to Private Internet Access:

```
piactl login {{path/to/login_file}}
```

- Connect to Private Internet Access:

```
piactl connect
```

- Disconnect from Private Internet Access:

```
piactl disconnect
```

- Enable or disable the Private Internet Access daemon in the background:

```
piactl background {{enable|disable}}
```

- List all available VPN regions:

```
piactl get regions
```

- Display the current VPN region:

```
piactl get region
```

- Set your VPN region:

```
piactl set region {{region}}
```

- Log out of Private Internet Access:

```
piactl logout
```

pic

Picture preprocessor for the groff (GNU Troff) document formatting system.

See also: **`groff`**, **`troff`**.

More information: https://manned.org/pic.

- Process input with pictures, saving the output for future typesetting with groff to PostScript:

```
pic {{path/to/input.pic}} > {{path/to/output.roff}}
```

- Typeset input with pictures to PDF using the [me] macro package:

```
pic -T {{pdf}} {{path/to/input.pic}} | groff -{{me}} -T
{{pdf}} > {{path/to/output.pdf}}
```

picard

Next generation MusicBrainz tagging application.

More information: https://picard-docs.musicbrainz.org/en/getting_started/starting.html.

- Start Picard:

```
picard
```

- Open a set of files:

```
picard {{path/to/file1.mp3 path/to/file2.mp3 ...}}
```

- Display the version of Picard installed:

```
picard {{[-V|--long-version]}}
```

picgo

Upload an image to an image hosting service (default is SM.MS).

More information: https://github.com/PicGo/PicGo-Core.

- Interactively select a hosting service:

  ```
  picgo set uploader
  ```

- Upload the image in current clipboard:

  ```
  picgo upload
  ```

- Upload an image from a specific path:

  ```
  picgo upload {{path/to/image}}
  ```

pickle

A PHP extension installer based on Composer.

More information: https://github.com/FriendsOfPHP/pickle#usage.

- Install a specific PHP extension:

```
pickle install {{extension_name}}
```

- Convert an existing PECL extension configuration to a Pickle configuration file:

```
pickle convert {{path/to/directory}}
```

- Validate a PECL extension:

```
pickle validate {{path/to/directory}}
```

- Package a PECL extension for release:

```
pickle release {{path/to/directory}}
```

pico

Text editor styled after the Alpine Composer.

More information: https://manned.org/pico.

- Start the editor:

```
pico {{path/to/file}}
```

- Start the editor with the cursor located n lines into the file:

```
pico +{{n}} {{path/to/file}}
```

- Start the editor with the cursor shown before the current selection:

```
pico -g {{path/to/file}}
```

- Define the quote string for files such as email:

```
pico -Q "{{quotestring}}" {{path/to/file}}
```

- Enable mouse functionality when run within an `xterm` window:

```
pico -m {{path/to/file}}
```

- Set the operating directory for `pico`:

```
pico -o {{path/to/directory}}
```

- Enable "view only" mode, which disallows any edits:

```
pico -v {{path/to/file}}
```

- Display all files including those beginning with a period:

```
pico -a
```

picocom

Minimal program to emulate serial consoles.

More information: https://manned.org/picocom.

- Connect to a serial console with the default baud rate of 9600:

```
sudo picocom {{/dev/ttyXYZ}}
```

- Connect to a serial console with a specified baud rate:

```
sudo picocom {{/dev/ttyXYZ}} {{[-b|--baud]}} {{baud_rate}}
```

- Map special characters (e.g. LF to CRLF):

```
sudo picocom {{/dev/ttyXYZ}} --imap {{lfcrlf}}
```

- Exit picocom:

```
<Ctrl a><Ctrl x>
```

- Display help:

```
picocom {{[-h|--help]}}
```

picom-trans

Set the window opacity for the **picom** window compositor.

More information: https://github.com/yshui/picom.

- Set the currently focused window opacity to a specific percentage:

```
picom-trans --current --opacity {{90}}
```

- Set the opacity of a window with a specific name:

```
picom-trans --name {{Firefox}} --opacity {{90}}
```

- Set the opacity of a specific window selected via mouse cursor:

```
picom-trans --select --opacity {{90}}
```

- Toggle the opacity of a specific window:

```
picom-trans --name {{Firefox}} --toggle
```

picotool

Manage Raspberry Pi Pico boards.

More information: https://github.com/raspberrypi/picotool.

- Display information about the currently loaded program on a Pico:

```
picotool info
```

- Load a binary onto a Pico:

```
picotool load {{path/to/binary}}
```

- Convert an ELF or BIN file to UF2:

```
picotool uf2 convert {{path/to/elf_or_bin}} {{path/to/output}}
```

- Reboot a Pico:

```
picotool reboot
```

- List all known registers:

```
picotool otp list
```

- Display version:

```
picotool version
```

- Display help:

```
picotool help
```

picttoppm

Convert a Macintosh PICT file to a PPM image.

More information: https://netpbm.sourceforge.net/doc/picttoppm.html.

- Convert a PICT file to a PPM image:

```
picttoppm {{path/to/file.pict}} > {{path/to/file.ppm}}
```

- Force any images in the PICT file to be output at full resolution:

```
picttoppm {{[-fu|-fullres]}} {{path/to/file.pict}} > {{path/to/file.ppm}}
```

- Do not assume that the input file contains a PICT header and execute quickdraw operations only:

```
picttoppm {{[-n|-noheader]}} {{[-quic|-quickdraw]}} {{path/to/file.pict}} > {{path/to/file.ppm}}
```

pie

The PHP Installer for Extensions.

More information: https://github.com/php/pie/blob/main/docs/usage.md.

- Install or update an extension:

```
pie install {{vendor}}/{{extension}}
```

- List installed extensions and their versions:

```
pie show
```

- Display information about a specific package:

```
pie info {{vendor}}/{{extension}}
```

- List the configured repositories:

```
pie repository:list
```

- Add a repository:

```
pie repository:add {{type}} {{url}}
```

- Remove a repository:

```
pie repository:remove {{url}}
```

pigz

Multithreaded zlib compression utility.

More information: https://github.com/madler/pigz.

- Compress a file with default options:

```
pigz {{path/to/file}}
```

- Compress a file using the best compression method:

```
pigz {{[-9|--best]}} {{path/to/file}}
```

- Compress a file using no compression and 4 processors:

```
pigz -0 {{[-p|--processes]}} {{4}} {{path/to/file}}
```

- Compress a directory using tar:

```
tar cf - {{path/to/directory}} | pigz > {{path/to/
file.tar.gz}}
```

- Decompress a file:

```
pigz {{[-d|--decompress]}} {{archive.gz}}
```

- List the contents of an archive:

```
pigz {{[-l|--list]}} {{archive.tar.gz}}
```

ping

Send ICMP ECHO_REQUEST packets to network hosts.

More information: https://manned.org/ping.

- Ping host:

```
ping {{host}}
```

- Ping a host only a specific number of times:

```
ping -c {{count}} {{host}}
```

- Ping host, specifying the interval in seconds between requests (default is 1 second):

```
ping -i {{seconds}} {{host}}
```

- Ping host without trying to lookup symbolic names for addresses:

```
ping -n {{host}}
```

- Ping host and ring the bell when a packet is received (if your terminal supports it):

```
ping -a {{host}}
```

- Also display a message if no response was received:

```
ping -O {{host}}
```

- Ping a host with specific number of pings, per-packet response timeout (`-W`), and total time limit (`-w`) of the entire ping run:

```
ping -c {{count}} -W {{seconds}} -w {{seconds}} {{host}}
```

ping.py

Simple ICMP ping using Impacket to check if an IPv4 host is reachable.

Sends ICMP echo requests and listens for echo replies. Requires root privileges for raw socket access (e.g., run with **sudo**).

More information: https://github.com/fortra/impacket.

- Ping a host from a specified source IPv4 address:

```
ping.py {{source_ipv4}} {{destination_ipv4}}
```

- Ping 192.168.1.100 from 192.168.1.10:

```
ping.py 192.168.1.10 192.168.1.100
```

ping6

Send ICMP ECHO_REQUEST packets to network hosts via IPv6 address.

Note: **`ping6`** has been merged to **`ping`** in modern systems.

More information: https://manned.org/ping6.

- Ping a host:

```
ping6 {{host}}
```

- Ping a host only a specific number of times:

```
ping6 -c {{count}} {{host}}
```

- Ping a host, specifying the interval in seconds between requests (default is 1 second):

```
ping6 -i {{seconds}} {{host}}
```

- Ping a host without trying to lookup symbolic names for addresses:

```
ping6 -n {{host}}
```

- Ping a host and ring the bell when a packet is received (if your terminal supports it):

```
ping6 -a {{host}}
```

ping6.py

Simple ICMPv6 ping using Impacket to check if an IPv6 host is reachable.

Sends ICMPv6 echo requests and listens for echo replies. Requires root privileges for raw socket access (e.g., run with **sudo**).

More information: https://github.com/fortra/impacket.

- Ping an IPv6 host from a specified source IPv6 address:

```
ping6.py {{source_ipv6}} {{destination_ipv6}}
```

- Ping 2001:db8::2 from 2001:db8::1:

```
ping6.py 2001:db8::1 2001:db8::2
```

pinky

Print user information using the **`finger`** protocol.

More information: https://manned.org/pinky.

- Display details about the current user:

```
pinky
```

- Display details for a specific user:

```
pinky {{user}}
```

- Display details in the long format:

```
pinky {{user}} -l
```

- Omit the user's home directory and shell in long format:

```
pinky {{user}} -lb
```

- Omit the user's project file in long format:

```
pinky {{user}} -lh
```

- Omit the column headings in short format:

```
pinky {{user}} -f
```

pint

An opinionated PHP code style fixer based on PHP-CS-Fixer.

More information: https://laravel.com/docs/pint.

- Execute code style fixing:

  ```
  pint
  ```

- Display all files that are changed:

  ```
  pint -v
  ```

- Execute code style linting without applying changes:

  ```
  pint --test
  ```

- Execute code style fixes using a specific configuration file:

  ```
  pint --config {{path/to/pint.json}}
  ```

- Execute code style fixes using a specific preset:

  ```
  pint --preset {{psr12}}
  ```

pinta

A free, open source program for drawing and image editing.

More information: https://www.pinta-project.com/user-guide/overview/.

- Start Pinta:

```
pinta
```

- Open specific files:

```
pinta {{path/to/image1 path/to/image2 ...}}
```

pio access

Set the access level on published resources (packages) in the registry.

More information: https://docs.platformio.org/en/latest/core/userguide/access/.

- Grant a user access to a resource:

```
pio access grant {{guest|maintainer|admin}} {{username}}
{{resource_urn}}
```

- Remove a user's access to a resource:

```
pio access revoke {{username}} {{resource_urn}}
```

- Show all resources that a user or team has access to and the access level:

```
pio access list {{username}}
```

- Restrict access to a resource to specific users or team members:

```
pio access private {{resource_urn}}
```

- Allow all users access to a resource:

```
pio access public {{resource_urn}}
```

pio account

Manage your PlatformIO account.

More information: https://docs.platformio.org/en/latest/core/userguide/account/.

- Register a new PlatformIO account:

```
pio account register {{[-u|--username]}} {{username}} {{[-e|--email]}} {{email}} {{[-p|--password]}} {{password}} --firstname {{firstname}} --lastname {{lastname}}
```

- Permanently delete your PlatformIO account and related data:

```
pio account destroy
```

- Log in to your PlatformIO account:

```
pio account login {{[-u|--username]}} {{username}} {{[-p|--password]}} {{password}}
```

- Log out of your PlatformIO account:

```
pio account logout
```

- Update your PlatformIO profile:

```
pio account update {{[-u|--username]}} {{username}} {{[-e|--email]}} {{email}} --firstname {{firstname}} --lastname {{lastname}} --current-password {{password}}
```

- Show detailed information about your PlatformIO account:

```
pio account show
```

- Reset your password using your username or email:

```
pio account forgot {{[-u|--username]}} {{username_or_email}}
```

pio boards

List pre-configured embedded boards available in PlatformIO.

More information: https://docs.platformio.org/en/latest/core/userguide/cmd_boards.html.

- List all available boards:

```
pio boards
```

- List only boards from installed platforms:

```
pio boards --installed
```

pio check

Perform a static analysis check on a PlatformIO project.

More information: https://docs.platformio.org/en/latest/core/userguide/cmd_check.html.

- Perform a basic analysis check on the current project:

```
pio check
```

- Perform a basic analysis check on a specific project:

```
pio check {{[-d|--project-dir]}} {{project_directory}}
```

- Perform an analysis check for a specific environment:

```
pio check {{[-e|--environment]}} {{environment}}
```

- Perform an analysis check and only report a specified defect severity type:

```
pio check --severity {{low|medium|high}}
```

- Perform an analysis check and show detailed information when processing environments:

```
pio check {{[-v|--verbose]}}
```

pio ci

Build PlatformIO projects with an arbitrary source code structure.

This will create a new temporary project which the source code will be copied into.

More information: https://docs.platformio.org/en/latest/core/userguide/cmd_ci.html.

- Build a PlatformIO project in the default system temporary directory and delete it afterwards:

  ```
  pio ci {{path/to/project}}
  ```

- Build a PlatformIO project and specify specific libraries:

  ```
  pio ci {{[-l|--lib]}} {{path/to/library_directory}} {{path/to/project}}
  ```

- Build a PlatformIO project and specify a specific board (`pio boards` lists all of them):

  ```
  pio ci {{[-b|--board]}} {{board}} {{path/to/project}}
  ```

- Build a PlatformIO project in a specific directory:

  ```
  pio ci --build-dir {{path/to/build_directory}} {{path/to/project}}
  ```

- Build a PlatformIO project and don't delete the build directory:

  ```
  pio ci --keep-build-dir {{path/to/project}}
  ```

- Build a PlatformIO project using a specific configuration file:

  ```
  pio ci {{[-c|--project-conf]}} {{path/to/platformio.ini}}
  ```

pio debug

Debug PlatformIO projects.

More information: https://docs.platformio.org/en/latest/core/userguide/cmd_debug.html.

- Debug the PlatformIO project in the current directory:

```
pio debug
```

- Debug a specific PlatformIO project:

```
pio debug {{[-d|--project-dir]}} {{path/to/
platformio_project}}
```

- Debug a specific environment:

```
pio debug {{[-e|--environment]}} {{environment}}
```

- Debug a PlatformIO project using a specific configuration file:

```
pio debug {{[-c|--project-conf]}} {{path/to/platformio.ini}}
```

- Debug a PlatformIO project using the `gdb` debugger:

```
pio debug --interface {{gdb}} {{gdb_options}}
```

pio device

Manage and monitor PlatformIO devices.

More information: https://docs.platformio.org/en/latest/core/userguide/device/.

- List all available serial ports:

```
pio device list
```

- List all available logical devices:

```
pio device list --logical
```

- Start an interactive device monitor:

```
pio device monitor
```

- Start an interactive device monitor and listen to a specific port:

```
pio device monitor {{[-p|--port]}} {{/dev/ttyUSBX}}
```

- Start an interactive device monitor and set a specific baud rate (defaults to 9600):

```
pio device monitor {{[-b|--baud]}} {{57600}}
```

- Start an interactive device monitor and set a specific EOL character (defaults to CRLF):

```
pio device monitor --eol {{CRLF|CR|LF}}
```

- Go to the menu of the interactive device monitor:

```
<Ctrl t>
```

pio home

Launch the PlatformIO Home web server.

More information: https://docs.platformio.org/en/latest/core/userguide/cmd_home.html.

- Open PlatformIO Home in the default web browser:

```
pio home
```

- Use a specific HTTP port (defaults to 8008):

```
pio home --port {{port}}
```

- Bind to a specific IP address (defaults to 127.0.0.1):

```
pio home --host {{ip_address}}
```

- Do not automatically open PlatformIO Home in the default web browser:

```
pio home --no-open
```

- Automatically shutdown the server on timeout (in seconds) when no clients are connected:

```
pio home --shutdown-timeout {{time}}
```

- Specify a unique session identifier to keep PlatformIO Home isolated from other instances and protected from 3rd party access:

```
pio home --session-id {{id}}
```

pio init

This command is an alias of **`pio project init`**.

- View documentation for the original command:

  ```
  tldr pio project
  ```

pio lib

Manage PlatformIO libraries.

More information: https://docs.platformio.org/en/latest/core/userguide/lib/.

- List installed libraries:

```
pio lib list
```

- List built-in libraries based on installed development platforms and their frameworks:

```
pio lib builtin
```

- Search for existing libraries:

```
pio lib search {{keyword}}
```

- Show details about a library:

```
pio lib show {{library}}
```

- Install a library:

```
pio lib install {{library}}
```

- Update installed libraries:

```
pio lib update
```

- Uninstall a library:

```
pio lib uninstall {{library}}
```

- Show PlatformIO library registry statistics:

```
pio lib stats
```

pio org

Manage PlatformIO organizations and their owners.

More information: https://docs.platformio.org/en/latest/core/userguide/org/.

- Create a new organization:

```
pio org create {{organization_name}}
```

- Delete an organization:

```
pio org destroy {{organization_name}}
```

- Add a user to an organization:

```
pio org add {{organization_name}} {{username}}
```

- Remove a user from an organization:

```
pio org remove {{organization_name}} {{username}}
```

- List all organizations the current user is a member of and their owners:

```
pio org list
```

- Update the name, email or display name of an organization:

```
pio org update --orgname {{new_organization_name}} --email
{{new_email}} --displayname {{new_display_name}}
{{organization_name}}
```

pio pkg

Manage packages in the registry.

Packages can only be removed within 72 hours (3 days) from the date that they are published.

More information: https://docs.platformio.org/en/latest/core/userguide/package/.

- Create a package tarball from the current directory:

```
pio pkg pack {{[-o|--output]}} {{path/to/package.tar.gz}}
```

- Create and publish a package tarball from the current directory:

```
pio pkg publish
```

- Publish the current directory and restrict public access to it:

```
pio pkg publish --private
```

- Publish a package:

```
pio pkg publish {{path/to/package.tar.gz}}
```

- Publish a package with a custom release date (UTC):

```
pio pkg publish {{path/to/package.tar.gz}} --released-at
"{{2021-04-08 21:15:38}}"
```

- Remove all versions of a published package from the registry:

```
pio pkg unpublish {{package}}
```

- Remove a specific version of a published package from the registry:

```
pio pkg unpublish {{package}}@{{version}}
```

- Undo the removal, putting all versions or a specific version of the package back into the registry:

```
pio pkg unpublish --undo {{package}}@{{version}}
```

pio platform

Manage PlatformIO development platforms.

More information: https://docs.platformio.org/en/latest/core/userguide/platforms/.

- List all installed development platforms:

```
pio platform list
```

- Search for existing development platforms:

```
pio platform search {{platform}}
```

- Show details about a development platform:

```
pio platform show {{platform}}
```

- Install a development platform:

```
pio platform install {{platform}}
```

- Update installed development platforms:

```
pio platform update
```

- Uninstall a development platform:

```
pio platform uninstall {{platform}}
```

- List all supported frameworks:

```
pio platform frameworks
```

pio project

Manage PlatformIO projects.

More information: https://docs.platformio.org/en/latest/core/userguide/project/.

- Initialize a new PlatformIO project:

```
pio project init
```

- Initialize a new PlatformIO project in a specific directory:

```
pio project init {{[-d|--project-dir]}} {{path/to/
project_directory}}
```

- Initialize a new PlatformIO project, specifying a board ID:

```
pio project init {{[-b|--board]}} {{ATmega328P|uno|...}}
```

- Initialize a new PlatformIO based project, specifying one or more project options:

```
pio project init {{[-O|--project-option]}} "{{option}}
={{value}}" {{[-O|--project-option]}} "{{option}}={{value}}"
```

- Print the configuration of a project:

```
pio project config
```

pio remote

Helper command for PlatformIO Remote Development.

`pio remote [command]` takes the same arguments as its locally executing counterpart **`pio [command]`**.

More information: https://docs.platformio.org/en/latest/core/userguide/remote/index.html.

- List all active Remote Agents:

```
pio remote agent list
```

- Start a new Remote Agent with a specific name and share it with friends:

```
pio remote agent start {{[-n|--name]}} {{agent_name}} {{[-s|--share]}} {{example1@example.com}} {{[-s|--share]}} {{example2@example.com}}
```

- List devices from specified Agents (omit `--agent` to specify all Agents):

```
pio remote --agent {{agent_name1}} --agent {{agent_name2}} device list
```

- Connect to the serial port of a remote device:

```
pio remote --agent {{agent_name}} device monitor
```

- Run all targets on a specified Agent:

```
pio remote --agent {{agent_name}} run
```

- Update installed core packages, development platforms and global libraries on a specific Agent:

```
pio remote --agent {{agent_name}} update
```

- Run all tests in all environments on a specific Agent:

```
pio remote --agent {{agent_name}} test
```

pio run

Run PlatformIO project targets.

More information: https://docs.platformio.org/en/latest/core/userguide/cmd_run.html.

- List all available project targets:

```
pio run --list-targets
```

- List all available project targets of a specific environment:

```
pio run --list-targets {{[-e|--environment]}} {{environment}}
```

- Run all targets:

```
pio run
```

- Run all targets of specified environments:

```
pio run {{[-e|--environment]}} {{environment1}} {{[-e|--environment]}} {{environment2}}
```

- Run specified targets:

```
pio run {{[-t|--target]}} {{target1}} {{[-t|--target]}} {{target2}}
```

- Run the targets of a specified configuration file:

```
pio run {{[-c|--project-conf]}} {{path/to/platformio.ini}}
```

pio settings

View and modify PlatformIO settings.

More information: https://docs.platformio.org/en/latest/core/userguide/cmd_settings.html.

- Display the names, values and descriptions of all PlatformIO settings:

```
pio settings get
```

- Display the name, value and description of a specific PlatformIO setting:

```
pio settings get {{setting}}
```

- Set a specific setting value:

```
pio settings set {{setting}} {{value}}
```

- Reset the values of all modified settings to their factory defaults:

```
pio settings reset
```

pio system

Miscellaneous system commands for PlatformIO.

More information: https://docs.platformio.org/en/latest/core/userguide/system/.

- Install shell completion for the current shell (supports Bash, fish, Zsh and PowerShell):

```
pio system completion install
```

- Uninstall shell completion for the current shell:

```
pio system completion uninstall
```

- Display system-wide PlatformIO information:

```
pio system info
```

- Remove unused PlatformIO data:

```
pio system prune
```

- Remove only cached data:

```
pio system prune --cache
```

- List unused PlatformIO data that would be removed but do not actually remove it:

```
pio system prune --dry-run
```

pio team

Manage PlatformIO teams.

More information: https://docs.platformio.org/en/latest/core/userguide/team/.

- Create a new team with the specified description:

```
pio team create --description {{description}}
{{organization_name}}:{{team_name}}
```

- Delete a team:

```
pio team destroy {{organization_name}}:{{team_name}}
```

- Add a new user to a team:

```
pio team add {{organization_name}}:{{team_name}} {{username}}
```

- Remove a user from a team:

```
pio team remove {{organization_name}}:{{team_name}}
{{username}}
```

- List all teams that the user is part of and their members:

```
pio team list
```

- List all teams in an organization:

```
pio team list {{organization_name}}
```

- Rename a team:

```
pio team update --name {{new_team_name}}
{{organization_name}}:{{team_name}}
```

- Change the description of a team:

```
pio team update --description {{new_description}}
{{organization_name}}:{{team_name}}
```

pio test

Run local tests on a PlatformIO project.

More information: https://docs.platformio.org/en/latest/core/userguide/cmd_test.html.

- Run all tests in all environments of the current PlatformIO project:

```
pio test
```

- Test only specific environments:

```
pio test {{[-e|--environment]}} {{environment1}} {{[-e|--environment]}} {{environment2}}
```

- Run only tests whose name matches a specific glob pattern:

```
pio test {{[-f|--filter]}} "{{pattern}}"
```

- Ignore tests whose name matches a specific glob pattern:

```
pio test {{[-i|--ignore]}} "{{pattern}}"
```

- Specify a port for firmware uploading:

```
pio test --upload-port {{upload_port}}
```

- Specify a custom configuration file for running the tests:

```
pio test {{[-c|--project-conf]}} {{path/to/platformio.ini}}
```

pio update

Update installed PlatformIO Core packages, development platforms and global libraries.

See also: **`pio platform update`**, **`pio lib update`**.

More information: https://docs.platformio.org/en/latest/core/userguide/cmd_update.html.

- Perform a full update of all packages, development platforms and global libraries:

  ```
  pio update
  ```

- Update core packages only (skips platforms and libraries):

  ```
  pio update --core-packages
  ```

- Check for new versions of packages, platforms and libraries but do not actually update them:

  ```
  pio update --dry-run
  ```

pio upgrade

Update PlatformIO to the latest version.

More information: https://docs.platformio.org/en/latest/core/userguide/cmd_upgrade.html.

- Update PlatformIO to the latest version:

```
pio upgrade
```

- Update PlatformIO to the latest development (unstable) version:

```
pio upgrade --dev
```

pio

Development environment for embedded boards.

Some subcommands such as **`run`** have their own usage documentation.

More information: https://docs.platformio.org/en/latest/core/userguide/.

- Display help and list subcommands:

```
pio {{[-h|--help]}}
```

- Display help for a specific subcommand:

```
pio {{subcommand}} {{[-h|--help]}}
```

- Display version:

```
pio --version
```

piodebuggdb

This command is an alias of **`pio debug --interface=gdb`**.

- View documentation for the original command:

  ```
  tldr pio debug
  ```

pip cache

Inspect and manage pip's wheel cache.

More information: https://pip.pypa.io/en/stable/cli/pip_cache/.

- Show the location of the pip cache directory:

```
pip cache dir
```

- List filenames of all packages currently stored in the cache:

```
pip cache list
```

- Remove all files from the pip cache:

```
pip cache purge
```

- Remove cached files matching a specific package name:

```
pip cache remove {{package_name}}
```

pip check

Check installed packages for broken or incompatible dependencies.

More information: https://pip.pypa.io/en/stable/cli/pip_check/.

- Check for broken or missing requirements:

```
pip check
```

- Check and log output to a file:

```
pip check --log {{path/to/file}}
```

pip config

Manage local and global configuration for pip.

More information: https://pip.pypa.io/en/stable/cli/pip_config/.

- List all configuration values:

```
pip config list
```

- Show configuration files and their values:

```
pip config debug
```

- Set the value for a command option:

```
pip config set {{command.option}} {{value}} {{--global|--user|--site}}
```

- Get the value for a command option:

```
pip config get {{command.option}} {{--global|--user|--site}}
```

- Unset the value for a command option:

```
pip config unset {{command.option}} {{--global|--user|--site}}
```

- Edit the configuration file with the default editor:

```
pip config edit {{--global|--user|--site}}
```

- Edit the configuration file with a specific editor:

```
pip config edit {{--global|--user|--site}} --editor {{path/to/editor_binary}}
```

pip debug

Display debug information about the current pip environment and configuration.

Intended for debugging; options and output may change without notice.

More information: https://pip.pypa.io/en/stable/cli/pip_debug/.

- Show general debug information:

```
pip debug
```

- Show debug information for a specific platform:

```
pip debug --platform {{platform}}
```

- Show debug information for a specific Python version:

```
pip debug --python-version {{version}}
```

- Show debug information for a specific Python implementation:

```
pip debug --implementation {{implementation}}
```

- Show debug information for a specific ABI:

```
pip debug --abi {{abi}}
```

pip download

Download Python packages without installing them.

More information: https://pip.pypa.io/en/stable/cli/pip_download/.

- Download a package wheel or source archive to the current directory:

```
pip download {{package}}
```

- Download a specific version of a package:

```
pip download {{package}}=={{version}}
```

- Download a package and its dependencies to a specific directory:

```
pip download {{package}} {{[-d|--dest]}} {{path/to/
directory}}
```

- Download a package for a specific platform and Python version:

```
pip download {{package}} --only-binary :all: --platform
{{platform}} --python-version {{version}}
```

- Download a package from a specific index URL:

```
pip download {{package}} {{[-i|--index-url]}} {{url}}
```

pip freeze

List installed packages in requirements format.

More information: https://pip.pypa.io/en/stable/cli/pip_freeze.

- List installed packages:

```
pip freeze
```

- Write installed packages to the `requirements.txt` file:

```
pip freeze > requirements.txt
```

- List installed packages in a virtual environment, excluding globally installed packages:

```
pip freeze {{[-l|--local]}}
```

- List installed packages in the user-site:

```
pip freeze --user
```

- List all packages, including `pip`, `distribute`, `setuptools`, and `wheel` (they are skipped by default):

```
pip freeze --all
```

pip hash

Compute hashes of package archives for verification.

More information: https://pip.pypa.io/en/stable/cli/pip_hash/.

- Generate hash for a package file:

```
pip hash {{path/to/package.whl}}
```

- Generate hash using a specific algorithm:

```
pip hash {{[-a|--algorithm]}} {{sha256|sha384|sha512|...}} {{path/to/package.whl}}
```

- Generate hashes for multiple files:

```
pip hash {{path/to/package1.whl path/to/package2.whl ...}}
```

- Generate hash for downloaded archive:

```
pip hash {{path/to/package.tar.gz}}
```

pip index

Inspect information available from package indexes.

More information: https://pip.pypa.io/en/stable/cli/pip_index/.

- List all available versions of a package:

```
pip index versions {{package}}
```

- List versions from a specific index:

```
pip index versions {{package}} --index-url {{https://test.pypi.org/simple/}}
```

- Include pre-release versions:

```
pip index versions {{package}} --pre
```

- Include an additional index:

```
pip index versions {{package}} --extra-index-url {{https://example.com/simple/}}
```

- List versions for a specific platform:

```
pip index versions {{package}} --platform {{linux_x86_64}}
```

pip inspect

Inspect the Python environment and produce a report in JSON format.

More information: https://pip.pypa.io/en/stable/cli/pip_inspect/.

- Inspect the current environment:

```
pip inspect
```

- Inspect and save output to a file:

```
pip inspect > {{environment_report.json}}
```

- Inspect only locally installed packages (not global):

```
pip inspect --local
```

- Inspect only user-installed packages:

```
pip inspect --user
```

- Inspect packages in a specific path:

```
pip inspect --path {{path/to/environment}}
```

- Inspect with verbose output (Note: the `-v` flag can be repeated to increase verbosity):

```
pip inspect {{[-v|--verbose]}}
```

pip install

Install Python packages.

More information: https://pip.pypa.io/en/stable/cli/pip_install/.

- Install a package:

```
pip install {{package}}
```

- Install a specific version of a package:

```
pip install {{package}}=={{version}}
```

- Install packages listed in a file:

```
pip install {{[-r|--requirement]}} {{path/to/
requirements.txt}}
```

- Install packages from an URL or local file archive (e.g. `.tar.gz`, `.whl`):

```
pip install {{[-f|--find-links]}} {{url|path/to/file}}
```

- Install the local package in the current directory in develop (editable) mode:

```
pip install {{[-e|--editable]}} .
```

pip list

List installed Python packages.

More information: https://pip.pypa.io/en/stable/cli/pip_list/.

- List installed packages:

```
pip list
```

- List outdated packages that can be upgraded:

```
pip list {{[-o|--outdated]}}
```

- List up-to-date packages:

```
pip list {{[-u|--uptodate]}}
```

- List packages with JSON formatting:

```
pip list --format json
```

- List packages that are not required by other packages:

```
pip list --not-required
```

- List packages installed in user-site only:

```
pip list --user
```

- List packages and exclude editable packages from output:

```
pip list --exclude-editable
```

- List packages in freeze format (unlike `pip freeze`, does not show editable install information):

```
pip list --format freeze
```

pip lock

Lock Python packages and their dependencies into a reproducible file.

Experimental feature of **`pip`**.

More information: https://pip.pypa.io/en/stable/cli/pip_lock/.

- Generate a `pylock.toml` for the current project:

  ```
  pip lock {{[-e|--editable]}} .
  ```

- Lock dependencies from a requirements file:

  ```
  pip lock {{[-r|--requirement]}} {{path/to/requirements.txt}}
  ```

- Specify a custom output file for the lock:

  ```
  pip lock {{[-o|--output]}} {{path/to/lockfile.toml}}
  ```

- Lock a specific package and its dependencies:

  ```
  pip lock {{package}}
  ```

pip search

Search for Python packages by name or summary.

Doesn't work with PyPI; may work with other package indexes.

More information: https://pip.pypa.io/en/stable/cli/pip_search/.

- Search for packages using a custom package index:

```
pip search {{[-i|--index]}} {{example.com}} {{query}}
```

pip show

Show information about installed packages.

More information: https://pip.pypa.io/en/stable/cli/pip_show/.

- Show information about a package:

```
pip show {{package}}
```

- Show all information about a package:

```
pip show {{[-v|--verbose]}} {{package}}
```

- Show all installed files for a package:

```
pip show {{[-f|--files]}} {{package}}
```

pip uninstall

Uninstall Python packages.

More information: https://pip.pypa.io/en/stable/cli/pip_uninstall/.

- Uninstall a package:

```
pip uninstall {{package}}
```

- Uninstall packages listed in a specific file:

```
pip uninstall {{[-r|--requirement]}} {{path/to/
requirements.txt}}
```

- Uninstall package without asking for confirmation:

```
pip uninstall {{[-y|--yes]}} {{package}}
```

pip wheel

Build wheel archives for packages and dependencies.

More information: https://pip.pypa.io/en/stable/cli/pip_wheel/.

- Build a wheel for a package:

```
pip wheel {{package}}
```

- Build wheels for packages in requirements file:

```
pip wheel {{[-r|--requirement]}} {{path/to/requirements.txt}}
```

- Build wheel to a specific directory:

```
pip wheel {{package}} {{[-w|--wheel-dir]}} {{path/to/
directory}}
```

- Build wheel without dependencies:

```
pip wheel {{package}} --no-deps
```

- Build wheel from local project:

```
pip wheel {{path/to/project}}
```

- Build wheel from Git repository:

```
pip wheel git+{{https://github.com/user/repo.git}}
```

pip

Python package manager.

Some subcommands such as **`install`** have their own usage documentation.

More information: https://pip.pypa.io/en/stable/cli/pip/.

- Install a package (see `pip install` for more install examples):

```
pip install {{package}}
```

- Install a package to the user's directory instead of the system-wide default location:

```
pip install --user {{package}}
```

- Upgrade a package:

```
pip install {{[-U|--upgrade]}} {{package}}
```

- Uninstall a package:

```
pip uninstall {{package}}
```

- Save installed packages to file:

```
pip freeze > {{requirements.txt}}
```

- List installed packages:

```
pip list
```

- Show installed package info:

```
pip show {{package}}
```

- Install packages from a file:

```
pip install {{[-r|--requirement]}} {{requirements.txt}}
```

pip3

Python package manager.

More information: https://pip.pypa.io.

- Install a package:

```
pip3 install {{package}}
```

- Install a specific version of a package:

```
pip3 install {{package}}=={{version}}
```

- Upgrade a package:

```
pip3 install {{[-U|--upgrade]}} {{package}}
```

- Uninstall a package:

```
pip3 uninstall {{package}}
```

- Save the list of installed packages to a file:

```
pip3 freeze > {{requirements.txt}}
```

- Install packages from a file:

```
pip3 install {{[-r|--requirement]}} {{requirements.txt}}
```

- Show installed package info:

```
pip3 show {{package}}
```

pipenv

Simple and unified Python development workflow.

Manage packages and the virtual environment for a project.

More information: https://pypi.org/project/pipenv.

- Create a new project:

  `pipenv`

- Create a new project using Python 3:

  `pipenv --three`

- Install a package:

  `pipenv install {{package}}`

- Install all the dependencies for a project:

  `pipenv install`

- Install all the dependencies for a project (including dev packages):

  `pipenv install --dev`

- Uninstall a package:

  `pipenv uninstall {{package}}`

- Start a shell within the created virtual environment:

  `pipenv shell`

- Generate a `requirements.txt` (list of dependencies) for a project:

  `pipenv lock --requirements`

piper

A fast, local neural text to speech system.

Try out and download speech models from https://rhasspy.github.io/piper-samples.

More information: https://github.com/OHF-Voice/piper1-gpl.

- Output a WAV [f]ile using a text-to-speech [m]odel (assuming a configuration file at model_path + .json):

```
echo {{Thing to say}} | piper -m {{path/to/model.onnx}} -f {{outputfile.wav}}
```

- Output a WAV [f]ile using a [m]odel and specifying its JSON [c]onfig file:

```
echo {{Thing to say}} | piper -m {{path/to/model.onnx}} -c {{path/to/model.onnx.json}} -f {{outputfile.wav}}
```

- Select a particular speaker in a voice with multiple speakers by specifying the speaker's ID number:

```
echo {{Warum?}} | piper -m {{de_DE-thorsten_emotional-medium.onnx}} --speaker {{1}} -f {{angry.wav}}
```

- Stream the output to the mpv media player:

```
echo {{Hello world}} | piper -m {{en_GB-northern_english_male-medium.onnx}} --output-raw -f - | mpv -
```

- Speak twice as fast, with huge gaps between sentences:

```
echo {{Speaking twice the speed. With added drama!}} | piper -m {{file.onnx}} --length_scale {{0.5}} --sentence_silence {{2}} -f {{drama.wav}}
```

pipes.sh

A Bash script which draws randomly pathed pipes over the terminal.

More information: https://github.com/pipeseroni/pipes.sh.

- Change the pa[t]tern of the pipes:

```
pipes.sh -t {{0..9}}
```

- Change the [c]olor of the pipes:

```
pipes.sh -c {{0..7}}
```

- Change the [f]ramerate of the pipes:

```
pipes.sh -f {{20..100}}
```

- Disable [C]olors:

```
pipes.sh -C
```

- Display [v]ersion:

```
pipes.sh -v
```

pipx

Install and run Python applications in isolated environments.

More information: https://github.com/pypa/pipx.

- Run an app in a temporary virtual environment:

```
pipx run {{pycowsay}} {{moo}}
```

- Install a package in a virtual environment and add entry points to path:

```
pipx install {{package}}
```

- List installed packages:

```
pipx list
```

- Run an app in a temporary virtual environment with a package name different from the executable:

```
pipx run --spec {{httpx-cli}} {{httpx}} {{http://
www.github.com}}
```

- Inject dependencies into an existing virtual environment:

```
pipx inject {{package}} {{dependency1 dependency2 ...}}
```

- Install a package in a virtual environment with pip arguments:

```
pipx install --pip-args='{{pip-args}}' {{package}}
```

- Upgrade/reinstall/uninstall all installed packages:

```
pipx {{upgrade-all|uninstall-all|reinstall-all}}
```

pixi config

Manage the configuration file.

More information: https://pixi.sh/latest/reference/cli/#config.

- Edit the configuration file:

```
pixi config edit
```

- List all configurations:

```
pixi config list
```

- Prepend a value to a list configuration key:

```
pixi config prepend {{key}} {{value}}
```

- Append a value to a list configuration key:

```
pixi config append {{key}} {{value}}
```

- Set a configuration key to a value:

```
pixi config set {{key}} {{value}}
```

- Unset a configuration key:

```
pixi config unset {{key}}
```

pixi global

Manage global packages.

More information: https://pixi.sh/latest/reference/cli/#global.

- Install a package globally and add to path:

```
pixi global install {{package1 package2 ...}}
```

- Uninstall a package globally:

```
pixi global remove {{package1 package2 ...}}
```

- List all globally installed packages:

```
pixi global list
```

- Update a globally installed package:

```
pixi global upgrade {{package}}
```

- Update all globally installed packages:

```
pixi global upgrade-all
```

pixi project

Modify the project configuration file.

More information: https://pixi.sh/latest/reference/cli/#project.

- Manage project channels:

```
pixi project channel {{command}}
```

- Manage project description:

```
pixi project description {{command}}
```

- Manage project platform:

```
pixi project platform {{command}}
```

- Manage project version:

```
pixi project version {{command}}
```

- Manage project environment:

```
pixi project environment {{command}}
```

pixi task

Manage tasks in the project environment.

More information: https://pixi.sh/latest/reference/cli/#task.

- Create a new task:

```
pixi task add {{task_name}} {{task_command}}
```

- List all tasks in the project:

```
pixi task list
```

- Remove a task:

```
pixi task remove {{task_name}}
```

- Create an alias for a task:

```
pixi task alias {{alias_name}} {{task1 task2 ...}}
```

pixi

Developer Workflow and Environment Management for projects.

More information: https://pixi.sh.

- Initialize a new project:

```
pixi init {{path/to/project}}
```

- Add project dependencies:

```
pixi add {{dependency1 dependency2 ...}}
```

- Start a pixi shell in the project environment:

```
pixi shell
```

- Run a task in the project environment:

```
pixi run {{task}}
```

- Manage tasks in the project environment:

```
pixi task {{command}}
```

- Display help:

```
pixi {{command}} --help
```

- Clean environment and task cache:

```
pixi clean
```

pixiecore

Manage the network booting of machines.

More information: https://github.com/danderson/netboot/tree/master/pixiecore.

- Start a PXE boot server which provides a `netboot.xyz` boot image:

```
pixiecore {{quick}} xyz --dhcp-no-bind
```

- Start a new PXE boot server which provides an Ubuntu boot image:

```
pixiecore {{quick}} ubuntu --dhcp-no-bind
```

- List all available boot images for quick mode:

```
pixiecore quick --help
```

pixterm

Image printing in the terminal.

See also: **`chafa`**, **`catimg`**.

More information: https://github.com/eliukblau/pixterm.

- Render a static image directly in the terminal:

```
pixterm {{path/to/file}}
```

- Use the image's original aspect ratio:

```
pixterm -s 2 {{path/to/file}}
```

- Specify a custom aspect ratio using a specific number of [t]erminal [r]ows and [c]olumns:

```
pixterm -tr {{24}} -tc {{80}} {{path/to/file}}
```

- Filter the output with a [m]atte background color and character [d]ithering:

```
pixterm -m {{000000}} -d 2 {{path/to/file}}
```

pjtoppm

Convert a HP PaintJet file to PPM.

More information: https://netpbm.sourceforge.net/doc/pjtoppm.html.

- Convert a HP PaintJet file to PPM:

```
pjtoppm {{path/to/input.pj}} > {{path/to/output.ppm}}
```

pkcs11-tool

Utility for managing and using PKCS #11 security tokens.

More information: https://github.com/OpenSC/OpenSC/wiki/Using-pkcs11-tool-and-OpenSSL.

- List slots and their potential token, using a specific module (e.g. `/usr/lib/softhsm/libsofthsm2.so`):

```
pkcs11-tool --module {{path/to/module.so}} {{[-L|--list-slots]}} {{[-T|--list-token-slots]}}
```

- List objects in a specific slot. (Note: `slot_id` is not the slot index shown as "Slot X"):

```
pkcs11-tool {{[-O|--list-objects]}} {{[-p|--pin]}} {{auth_pin}} --slot {{slot_id}}
```

- Create a new object with a specific label and type:

```
pkcs11-tool --slot {{slot_id}} {{[-p|--pin]}} {{auth_pin}} {{[-y|--type]}} {{cert|privkey|pubkey|secrkey|data|...}} {{[-a|--label]}} "{{label}}" {{[-d|--id]}} {{01}} {{[-w|--write-object]}} {{path/to/cert.crt}}
```

- Delete an object by its label and type:

```
pkcs11-tool --slot {{slot_id}} {{[-p|--pin]}} {{auth_pin}} {{[-y|--type]}} {{cert|privkey|pubkey|secrkey|data|...}} {{[-a|--label]}} "{{label}}" {{[-b|--delete-object]}}
```

pkg-config

Provide the details of installed libraries for compiling applications.

More information: https://manned.org/pkg-config.

- Get the list of libraries and their dependencies:

```
pkg-config --libs {{library1 library2 ...}}
```

- Get the list of libraries, their dependencies, and proper cflags for gcc:

```
pkg-config --cflags --libs {{library1 library2 ...}}
```

- Compile your code with libgtk-3, libwebkit2gtk-4.0 and all their dependencies:

```
c++ example.cpp $(pkg-config --cflags --libs gtk+-3.0
webkit2gtk-4.0) -o example
```

pkgz

Unified package manager for installing, removing, updating, and searching packages from multiple sources.

More information: https://github.com/roguehashrate/pkgz#-usage.

- Install a package:

```
pkgz install {{package_name}}
```

- Remove a package:

```
pkgz remove {{package_name}}
```

- Search for a package:

```
pkgz search {{package_name}}
```

- Update all packages in all sources:

```
pkgz update
```

- Display version:

```
pkgz --version
```

pkill

Signal process by name.

Mostly used for stopping processes.

More information: https://manned.org/pkill.

- Kill all processes which match:

```
pkill "{{process_name}}"
```

- Kill all processes which match their full command instead of just the process name:

```
pkill {{[-f|--full]}} "{{command_name}}"
```

- Force kill matching processes (can't be blocked):

```
pkill -9 "{{process_name}}"
```

- Send SIGUSR1 signal to processes which match:

```
pkill -USR1 "{{process_name}}"
```

- Kill the main `firefox` process to close the browser:

```
pkill {{[-o|--oldest]}} "{{firefox}}"
```

pkl

Manages, evaluates, and tests Pkl configuration modules.

More information: https://pkl-lang.org/main/current/pkl-cli.

- Evaluate the given Pkl modules and produce their rendering results:

```
pkl eval {{module.pkl}}
```

- Run as a server that communicates over `stdin` and `stdout`:

```
pkl server
```

- Evaluate Pkl modules as tests and produces a report:

```
pkl test {{module.pkl}}
```

- Start a REPL session:

```
pkl repl
```

- Prepare a Pkl project for publishing as a package:

```
pkl project package {{path/to/project_directory}}
```

- Resolve project dependencies and writes the resolved versions to a file at path `PklProject.deps.json`:

```
pkl project resolve {{path/to/project_directory}}
```

plantuml

Create UML diagrams from a plain text language and render them in different formats.

More information: https://plantuml.com/en/command-line.

- Render diagrams to default format (PNG):

```
plantuml {{diagram1.puml}} {{diagram2.puml}}
```

- Render a diagram in given format (e.g. `png`, `pdf`, `svg`, `txt`):

```
plantuml -t {{format}} {{diagram.puml}}
```

- Render all diagrams of a directory:

```
plantuml {{path/to/diagrams}}
```

- Render a diagram to the output directory:

```
plantuml -o {{path/to/output}} {{diagram.puml}}
```

- Render a diagram without storing the diagram's source code (Note: It's stored by default when the `-nometadata` option isn't specified):

```
plantuml -nometadata {{diagram.png}} > {{diagram.puml}}
```

- Retrieve source from a `plantuml` diagram's metadata:

```
plantuml -metadata {{diagram.png}} > {{diagram.puml}}
```

- Render a diagram with the configuration file:

```
plantuml -config {{config.cfg}} {{diagram.puml}}
```

- Display help:

```
plantuml -help
```

platformio

This command is an alias of **pio**.

- View documentation for the original command:

```
tldr pio
```

play

Audio player of SoX - Sound eXchange.

Plays any audio, with audio formats identified by the extension.

More information: https://sox.sourceforge.net.

- Play the given audio file:

```
play {{path/to/audio_file}}
```

- Play the given audio files:

```
play {{path/to/audio_file1 path/to/audio_file2 ...}}
```

- Play the given audio at twice the speed:

```
play {{path/to/audio_file}} speed 2.0
```

- Play the given audio in reverse:

```
play {{path/to/audio_file}} reverse
```

plenv

Switch between multiple versions of Perl.

More information: https://github.com/tokuhirom/plenv.

- Show the currently selected Perl version and how it was selected:

```
plenv version
```

- List all available installed Perl versions:

```
plenv versions
```

- Set the global Perl version (used unless a local or shell version takes priority):

```
plenv global {{version}}
```

- Set the local application-specific Perl version (used in the current directory and all directories below it):

```
plenv local {{version}}
```

- Set the shell-specific Perl version (used for the current session only):

```
plenv shell {{version}}
```

- Display help:

```
plenv
```

- Display help for a command:

```
plenv help {{command}}
```

plesk

Plesk hosting control panel.

More information: https://docs.plesk.com/en-US/obsidian/cli-linux/plesk-utility.75661/.

- Generate an auto login link for the admin user and print it:

```
plesk login
```

- Show product version information:

```
plesk version
```

- List all hosted domains:

```
plesk bin domain --list
```

- Start watching for changes in the `panel.log` file:

```
plesk log {{panel.log}}
```

- Start the interactive MySQL console:

```
plesk db
```

- Open the Plesk main configuration file in the default editor:

```
plesk conf {{panel.ini}}
```

plink

PuTTy's command line utility.

See also: **ssh**.

More information: https://manned.org/plink.

- Connect to an address:

```
plink {{IP_address}}
```

plocate

Find filenames quickly.

Make sure to run **`sudo updatedb`** to include new files.

More information: https://plocate.sesse.net.

- Look for patterns in the database (recomputed periodically):

  ```
  plocate {{pattern}}
  ```

- Look for a file by its exact filename (a pattern containing no globbing characters is interpreted as `*pattern*`):

  ```
  plocate */{{filename}}
  ```

pm2

Process manager for Node.js.

Used for log management, monitoring and configuring processes.

More information: https://pm2.keymetrics.io.

- Start a process with a name that can be used for later operations:

```
pm2 start {{app.js}} --name {{application_name}}
```

- List processes:

```
pm2 list
```

- Monitor all processes:

```
pm2 monit
```

- Stop a process:

```
pm2 stop {{application_name}}
```

- Restart a process:

```
pm2 restart {{application_name}}
```

- Dump all processes for resurrecting them later:

```
pm2 save
```

- Resurrect previously dumped processes:

```
pm2 resurrect
```

pngcheck

Forensics tool for validating the integrity of PNG based (PNG, JNG, MNG) image files.

Can also extract embedded images and text from a file.

More information: https://manned.org/pngcheck.

- Verify the integrity of an image file (width, height, and color depth):

```
pngcheck {{path/to/image.png}}
```

- Print information for an image with [c]olorized output:

```
pngcheck -c {{path/to/image.png}}
```

- Print [v]erbose information for an image:

```
pngcheck -cvt {{path/to/image.png}}
```

- Receive an image from `stdin` and display detailed information:

```
cat {{path/to/image.png}} | pngcheck -cvt
```

- [s]earch for PNGs within a specific file and display information about them:

```
pngcheck -s {{path/to/image.png}}
```

- Search for PNGs within another file and e[x]tract them:

```
pngcheck -x {{path/to/image.png}}
```

pngcrush

PNG compression utility.

More information: https://pmt.sourceforge.io/pngcrush.

- Compress a PNG file:

```
pngcrush {{in.png}} {{out.png}}
```

- Compress all PNGs and output them to the specified directory:

```
pngcrush -d {{path/to/output}} *.png
```

- Compress PNG file with all 114 available algorithms and pick the best result:

```
pngcrush -rem allb -brute -reduce {{in.png}} {{out.png}}
```

pngquant

PNG converter and lossy image compressor.

More information: https://manned.org/pngquant.

- Compress a specific PNG as much as possible and write result to a new file:

```
pngquant {{path/to/file.png}}
```

- Compress a specific PNG and override original:

```
pngquant --ext .png --force {{path/to/file.png}}
```

- Try to compress a specific PNG with custom quality (skip if below the min value):

```
pngquant --quality {{0-100}} {{path/to/file.png}}
```

- Compress a specific PNG with the number of colors reduced to 64:

```
pngquant {{64}} {{path/to/file.png}}
```

- Compress a specific PNG and skip if the file is larger than the original:

```
pngquant --skip-if-larger {{path/to/file.png}}
```

- Compress a specific PNG and remove metadata:

```
pngquant --strip {{path/to/file.png}}
```

- Compress a specific PNG and save it to the given path:

```
pngquant {{path/to/file.png}} --output {{path/to/file.png}}
```

- Compress a specific PNG and show progress:

```
pngquant --verbose {{path/to/file.png}}
```

pngtopam

Convert a PNG image to a Netpbm image.

See also: **pamtopng**.

More information: https://netpbm.sourceforge.net/doc/pngtopam.html.

- Convert the specified PNG image to a Netpbm image:

```
pngtopam {{path/to/image.png}} > {{path/to/output.pam}}
```

- Create an output image that includes both the main image and transparency mask of the input image:

```
pngtopam -alphapam {{path/to/image.png}} > {{path/to/
output.pam}}
```

- Replace transparent pixels by the specified color:

```
pngtopam {{[-m|-mix]}} {{[-ba|-background]}} {{color}}
{{path/to/image.png}} > {{path/to/output.pam}}
```

- Write tEXt chunks found in the input image to the specified text file:

```
pngtopam {{[-te|-text]}} {{path/to/file.txt}} {{path/to/
image.png}} > {{path/to/output.pam}}
```

pngtopnm

This command has been superseded by **`pngtopam`**.

More information: https://netpbm.sourceforge.net/doc/pngtopnm.html.

- View documentation for the current command:

```
tldr pngtopam
```

pnmalias

Apply antialiasing onto a PNM image.

More information: https://netpbm.sourceforge.net/doc/pnmalias.html.

- Perform antialiasing on a PNM image, taking black pixels as background and white pixels as foreground:

```
pnmalias {{path/to/input.pnm}} > {{path/to/output.ppm}}
```

- Explicitly specify the background and foreground color:

```
pnmalias -bcolor {{background_color}} -fcolor
{{foreground_color}} {{path/to/input.pnm}} > {{path/to/
output.ppm}}
```

- Apply altialiasing to foreground pixels only:

```
pnmalias {{[-fo|-fonly]}} {{path/to/input.pnm}} > {{path/to/
output.ppm}}
```

- Apply antialiasing to all surrounding pixels of background pixels:

```
pnmalias {{[-ba|-balias]}} {{path/to/input.pnm}} > {{path/to/
output.ppm}}
```

pnmarith

This command has been superseded by **pamarith**.

More information: https://netpbm.sourceforge.net/doc/pnmarith.html.

- View documentation for the current command:

```
tldr pamarith
```

pnmcolormap

Create quantization color map for a PNM image.

More information: https://netpbm.sourceforge.net/doc/pnmcolormap.html.

- Generate an image using only `n_colors` or less colors as close as possible to the input image:

```
pnmcolormap {{n_colors}} {{path/to/input.pnm}} > {{path/to/output.ppm}}
```

- Use the splitspread strategy for determining the output colors, possibly producing a better result for images with small details:

```
pnmcolormap {{[-splits|-splitspread]}} {{n_colors}} {{path/to/input.pnm}} > {{path/to/output.ppm}}
```

- Sort the resulting colormap, which is useful for comparing colormaps:

```
pnmcolormap {{[-so|-sort]}} {{path/to/input.pnm}} > {{path/to/output.ppm}}
```

pnmcomp

This command has been superseded by **pamcomp**.

More information: https://netpbm.sourceforge.net/doc/pnmcomp.html.

- View documentation for the current command:

```
tldr pamcomp
```

pnmconvol

Convolute a PNM image.

More information: https://netpbm.sourceforge.net/doc/pnmconvol.html.

- Convolve a PNM image with the specified convolution matrix:

```
pnmconvol -matrix=-1,3,-1 {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Convolve a PNM image with the convolution matrix in the specified files, one for each layer in the input image:

```
pnmconvol -matrixfile {{path/to/matrix1,path/to/matrix2,...}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Convolve a PNM image with the convolution matrix in the specified PNM file:

```
pnmconvol {{path/to/matrix.pnm}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Normalize the weights in the convolution matrix such that they add up to one:

```
pnmconvol -matrix=-1,3,-1 -normalize {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

pnmcrop

Crop PNM images.

More information: https://netpbm.sourceforge.net/doc/pnmcrop.html.

- Remove white borders on a PNM image:

```
pnmcrop {{[-w|-white]}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Remove borders of the specified color that are on the top and left side of the image:

```
pnmcrop -bg-color {{color}} {{[-t|-top]}} {{[-l|-left]}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Determine the color of the borders to be removed by the color of the pixel in the specified corner:

```
pnmcrop -bg-corner {{topleft|topright|bottomleft|bottomright}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Leave a border with a width of n pixels. Additionally, specify the behaviour if the image is entirely made out of background:

```
pnmcrop {{[-m|-margin]}} {{n}} {{[-blan|-blank-image]}} {{pass|minimize|maxcrop}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

pnmcut

This command has been superseded by **`pamcut`**.

More information: https://netpbm.sourceforge.net/doc/pnmcut.html.

- View documentation for the current command:

```
tldr pamcut
```

pnmdepth

This command is an alias of **pamdepth**.

- View documentation for the original command:

  `tldr pamdepth`

pnmenlarge

This command has been superseded by **`pamenlarge`**.

More information: https://netpbm.sourceforge.net/doc/pnmenlarge.html.

- View documentation for the current command:

```
tldr pamenlarge
```

pnmfile

This command has been superseded by **`pamfile`**.

More information: https://netpbm.sourceforge.net/doc/pnmfile.html.

- View documentation for the current command:

```
tldr pamfile
```

pnmflip

This command has been superseded by **`pamflip`**.

More information: https://netpbm.sourceforge.net/doc/pnmflip.html.

- View documentation for the current command:

```
tldr pamflip
```

pnmgamma

Perform gamma correction on PNM images.

More information: https://netpbm.sourceforge.net/doc/pnmgamma.html.

- Convert the image from BT.709 luminance to radiance or sRGB luminance:

```
pnmgamma -{{bt709tolinear|bt709tosrgb}} {{path/to/image.pnm}}
> {{path/to/output.pnm}}
```

- Convert the image from radiance or sRGB luminance to BT.709 luminance:

```
pnmgamma -{{lineartobt709|srgbtobt709}} {{path/to/image.pnm}}
> {{path/to/output.pnm}}
```

- Specify the gamma value used for the gamma transfer function:

```
pnmgamma {{[-ga|-gamma]}} {{value}} {{path/to/image.pnm}} >
{{path/to/output.pnm}}
```

- Specify the gamma value used for the gamma transfer function per color component:

```
pnmgamma {{[-rg|-rgamma]}} {{value}} {{[-gg|-ggamma]}}
{{value}} {{[-bg|-bgamma]}} {{value}} {{path/to/image.pnm}} >
{{path/to/output.pnm}}
```

pnmhisteq

Histogram-equalize a PNM image.

More information: https://netpbm.sourceforge.net/doc/pnmhisteq.html.

- Increase the contrast of a PNM image using histogram equalization:

```
pnmhisteq {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

- Only modify grey pixels:

```
pnmhisteq {{[-g|-grey]}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

- Do not include black or white pixels in the histogram equalization:

```
pnmhisteq -no{{black|white}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

pnmhistmap

Draw a histogram of a PNM image.

More information: https://netpbm.sourceforge.net/doc/pnmhistmap.html.

- Draw a histogram of a PNM image:

```
pnmhistmap {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

- Draw the histogram as dots instead of bars:

```
pnmhistmap {{[-d|-dots]}} {{path/to/input.pnm}} > {{path/to/
output.pnm}}
```

- Specify the range of intensity values to include:

```
pnmhistmap {{[-l|-lval]}} {{minval}} {{[-rv|-rval]}}
{{maxval}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

pnmindex

Build a visual index of multiple PNM images.

See also: **pamundice**.

More information: https://netpbm.sourceforge.net/doc/pnmindex.html.

- Produce an image containing thumbnails of the specified images in a grid:

```
pnmindex {{path/to/input1.pnm path/to/input2.pnm ...}} > {{path/to/output.pnm}}
```

- Specify the size of the (quadratic) thumbnails:

```
pnmindex {{[-s|-size]}} {{50}} {{path/to/input1.pnm path/to/input2.pnm ...}} > {{path/to/output.pnm}}
```

- Specify the number of thumbnails per row:

```
pnmindex {{[-a|-across]}} {{10}} {{path/to/input1.pnm path/to/input2.pnm ...}} > {{path/to/output.pnm}}
```

- Specify the maximum number of colors in the output:

```
pnmindex {{[-c|-colors]}} {{512}} {{path/to/input1.pnm path/to/input2.pnm ...}} > {{path/to/output.pnm}}
```

pnminterp

This command has been superseded by **pamstretch**.

More information: https://netpbm.sourceforge.net/doc/pnminterp.html.

- View documentation for the current command:

```
tldr pamstretch
```

pnminvert

Invert a PNM image.

More information: https://netpbm.sourceforge.net/doc/pnminvert.html.

- Invert the colors or greyscale values in a PNM image:

```
pnminvert {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

pnmmargin

Add a border to a PNM image.

More information: https://netpbm.sourceforge.net/doc/pnmmargin.html.

- Add a border of the specified size to a PNM image:

```
pnmmargin {{size}} {{path/to/image.pnm}} > {{path/to/
output.pnm}}
```

- Specify the color of the border:

```
pnmmargin {{[-c|-color]}} {{color}} {{size}} {{path/to/
image.pnm}} > {{path/to/output.pnm}}
```

pnmmercator

Perform Mercator transformations on Netpbm images.

See also: **pnmglobe**.

More information: https://netpbm.sourceforge.net/doc/pnmmercator.html.

- Convert a rectangular projection worldmap to Mercator projection:

```
pnmmercator {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Convert a Mercator projection worldmap to rectangular projection:

```
pnmmercator {{[-i|-inverse]}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

pnmmontage

Create a montage from multiple PNM images.

More information: https://netpbm.sourceforge.net/doc/pnmmontage.html.

- Produce a packing of the specified images:

```
pnmmontage {{path/to/image1.pnm path/to/image2.pnm ...}} > {{path/to/output.pnm}}
```

- Specify the quality of the packing (Note: Larger values produce smaller packings but take longer to compute.):

```
pnmmontage -{{0..9}} {{path/to/image1.pnm path/to/image2.pnm ...}} > {{path/to/output.pnm}}
```

- Produce a packing that is not larger than p percent of the optimal packing:

```
pnmmontage {{[-qua|-quality]}} {{p}} {{path/to/image1.pnm path/to/image2.pnm ...}} > {{path/to/output.pnm}}
```

- Write the positions of the input files within the packed image to a machine-readable file:

```
pnmmontage {{[-d|-data]}} {{path/to/datafile}} {{path/to/image1.pnm path/to/image2.pnm ...}} > {{path/to/output.pnm}}
```

pnmnlfilt

Apply a non-linear filter onto a PNM image.

More information: https://netpbm.sourceforge.net/doc/pnmnlfilt.html.

- Apply the "alpha trimmed mean" filter with the specified alpha and radius values onto the PNM image:

```
pnmnlfilt {{0.0..0.5}} {{radius}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Apply the "optimal estimation smoothing" filter with the specified noise threshold and radius onto the PNM image:

```
pnmnlfilt {{1.0..2.0}} {{radius}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Apply the "edge enhancement" filter with the specified alpha and radius onto the PNM image:

```
pnmnlfilt {{-0.9..(-0.1)}} {{radius}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

pnmnorm

Normalize the contrast in a PNM image.

See also: **pnmhisteq**.

More information: https://netpbm.sourceforge.net/doc/pnmnorm.html.

- Force the brightest pixels to be white, the darkest pixels to be black and spread out the ones in between linearly:

```
pnmnorm {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Force the brightest pixels to be white, the darkest pixels to be black and spread out the ones in between quadratically such that pixels with a brightness of n become 50 % bright:

```
pnmnorm {{[-midv|-midvalue]}} {{n}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Keep the pixels' hue, only modify the brightness:

```
pnmnorm {{[-k|-keephues]}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Specify a method to calculate a pixel's brightness:

```
pnmnorm -{{luminosity|colorvalue|saturation}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

pnmpad

Add borders to a PNM image.

See also: **pnmmargin**, **pamcut**, **pamcomp**.

More information: https://netpbm.sourceforge.net/doc/pnmpad.html.

- Add borders of the specified sizes to the image:

```
pnmpad {{[-l|-left]}} {{100}} {{[-ri|-right]}} {{150}} {{[-t|-top]}} {{123}} {{[-bo|-bottom]}} {{456}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Pad the image to the specified size:

```
pnmpad {{[-wi|-width]}} {{1000}} {{[-he|-height]}} {{500}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Pad the width of the image to the specified size, controlling the ratio between right and left padding:

```
pnmpad {{[-wi|-width]}} {{1000}} {{[-ha|-halign]}} {{0.7}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Pad the width of the image using the specified color:

```
pnmpad {{[-wi|-width]}} {{1000}} {{[-c|-color]}} {{red}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

pnmpaste

Paste a PNM image into another PNM image.

More information: https://netpbm.sourceforge.net/doc/pnmpaste.html.

- Paste a PNM image into another PNM image at the specified coordinates:

```
pnmpaste {{x}} {{y}} {{path/to/image1.pnm}} {{path/to/image2.pnm}} > {{path/to/output.pnm}}
```

- Paste the image read from `stdin` into the specified image:

```
{{command}} | pnmpaste {{x}} {{y}} {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Combine the overlapping pixels by the specified boolean operation, where white pixels represent `true` while black pixels represent `false`:

```
pnmpaste -{{and|nand|or|nor|xor|xnor}} {{x}} {{y}} {{path/to/image1.pnm}} {{path/to/image2.pnm}} > {{path/to/output.pnm}}
```

pnmpsnr

Compute the difference between two images.

More information: https://netpbm.sourceforge.net/doc/pnmpsnr.html.

- Compute the difference, i.e. the peak signal-to-noise ratio (PSNR) between two images:

```
pnmpsnr {{path/to/file1.pnm}} {{path/to/file2.pnm}}
```

- Compare the color components rather than the luminance and chrominance components of the images:

```
pnmpsnr {{path/to/file1.pnm}} {{path/to/file2.pnm}} -rgb
```

- Run in comparison mode, i.e. only output `nomatch` or `match` depending on whether the computing PSNR exceeds `n` or not:

```
pnmpsnr {{path/to/file1.pnm}} {{path/to/file2.pnm}} -target {{n}}
```

- Run in comparison mode and compare the individual image components, i.e. Y, Cb, and Cr, to the corresponding thresholds:

```
pnmpsnr {{path/to/file1.pnm}} {{path/to/file2.pnm}} -target1 {{threshold_Y}} -target2 {{threshold_Cb}} -target3 {{threshold_Cr}}
```

- Run in comparison mode and compare the individual image components, i.e. red, green, and blue to the corresponding thresholds:

```
pnmpsnr {{path/to/file1.pnm}} {{path/to/file2.pnm}} -rgb -target1 {{threshold_red}} -target2 {{threshold_green}} -target3 {{threshold_blue}}
```

- Produce machine-readable output:

```
pnmpsnr {{path/to/file1.pnm}} {{path/to/file2.pnm}} -machine
```

pnmquant

Quantize the colors in a PNM image into a smaller set.

This command is a combination of **`pnmcolormap`** and **`pnmremap`** and accepts the union of their options, except **`-mapfile`**.

See also: **`pnmquantall`**.

More information: https://netpbm.sourceforge.net/doc/pnmquant.html.

- Generate an image using only `n_colors` or less colors as close as possible to the input image:

```
pnmquant {{n_colors}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

pnmquantall

Run **pnmquant** on multiple files at once such that they share a common colormap.

See also: **pnmquant**.

More information: https://netpbm.sourceforge.net/doc/pnmquantall.html.

- Run pnmquant on multiple files with the specified parameters, overwriting the original files:

```
pnmquantall {{n_colors}} {{path/to/input1.pnm path/to/input2.pnm ...}}
```

- Save the quantised images to files named the same as the input files, but with the specified extension appended:

```
pnmquantall {{[-e|-ext]}} {{extension}} {{n_colors}} {{path/to/input1.pnm path/to/input2.pnm ...}}
```

pnmremap

Replace the colors in a PNM image.

More information: https://netpbm.sourceforge.net/doc/pnmremap.html.

- Replace the colors in an image with those in the specified color palette:

```
pnmremap {{[-ma|-mapfile]}} {{path/to/palette_file.ppm}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

- Use Floyd-Steinberg dithering for representing colors missing in the color palette:

```
pnmremap {{[-ma|-mapfile]}} {{path/to/palette_file.ppm}} {{[-fs|-floyd]}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

- Use the first color in the palette for representing colors missing in the color palette:

```
pnmremap {{[-ma|-mapfile]}} {{path/to/palette_file.ppm}} {{[-fi|-firstisdefault]}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

- Use the specified color for representing colors missing in the color palette:

```
pnmremap {{[-ma|-mapfile]}} {{path/to/palette_file.ppm}} {{[-m|-missingcolor]}} {{color}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

pnmrotate

Rotate a PNM image.

More information: https://netpbm.sourceforge.net/doc/pnmrotate.html.

- Rotate a PNM image by some angle (measured in degrees, counter-clockwise):

```
pnmrotate {{angle}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

- Specify the background color exposed by rotating the input image:

```
pnmrotate {{[-b|-background]}} {{color}} {{angle}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

- Disable anti-aliasing, improving performance but decreasing quality:

```
pnmrotate {{[-n|-noantialias]}} {{angle}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

pnmscale

This command has been replaced by **`pamscale`**.

More information: https://netpbm.sourceforge.net/doc/pnmscale.html.

- View documentation for `pamscale`:

```
tldr pamscale
```

pnmscalefixed

Scale a PNM file quickly with possibly reduced quality.

See also: **pamscale**.

More information: https://netpbm.sourceforge.net/doc/pnmscalefixed.html.

- Scale an image such that the result has the specified dimensions:

```
pnmscalefixed {{[-w|-width]}} {{width}} {{[-h|-height]}}
{{height}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

- Scale an image such that the result has the specified width, keeping the aspect ratio:

```
pnmscalefixed {{[-w|-width]}} {{width}} {{path/to/input.pnm}}
> {{path/to/output.pnm}}
```

- Scale an image such that its width and height is changed by the specified factors:

```
pnmscalefixed {{[-xsc|-xscale]}} {{x_factor}} {{[-ysc|-
yscale]}} {{y_factor}} {{path/to/input.pnm}} > {{path/to/
output.pnm}}
```

pnmshear

Shear a PNM image.

More information: https://netpbm.sourceforge.net/doc/pnmshear.html.

- Shear a PNM image by the specified angle:

```
pnmshear {{angle}} {{path/to/input.pnm}} > {{path/to/
output.pnm}}
```

- Specify the color of the background in the sheared image:

```
pnmshear {{[-b|-background]}} {{blue}} {{angle}} {{path/to/
input.pnm}} > {{path/to/output.pnm}}
```

- Do not perform anti-aliasing:

```
pnmshear {{[-n|-noantialias]}} {{angle}} {{path/to/
input.pnm}} > {{path/to/output.pnm}}
```

pnmsmooth

Smooth out a PNM image.

More information: https://netpbm.sourceforge.net/doc/pnmsmooth.html.

- Smooth out a PNM image using a convolution matrix of size 3x3:

```
pnmsmooth {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

- Smooth out a PNM image using a convolution matrix of size width times height:

```
pnmsmooth {{[-w|-width]}} {{width}} {{[-h|-height]}} {{height}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

pnmsplit

This command has been superseded by **pamsplit**.

More information: https://netpbm.sourceforge.net/doc/pnmsplit.html.

- View documentation for the current command:

```
tldr pamsplit
```

pnmtile

Replicate an image to fill a specified area.

More information: https://netpbm.sourceforge.net/doc/pnmtile.html.

- Replicate an image to fill an area of the specified dimensions:

```
pnmtile {{width}} {{height}} {{path/to/input.pnm}} > {{path/to/output.pnm}}
```

pnmtoddif

Convert a PNM image to a DDIF image file.

More information: https://netpbm.sourceforge.net/doc/pnmtoddif.html.

- Convert a PNM image to a DDIF image file:

```
pnmtoddif {{path/to/image.pnm}} > {{path/to/image.ddif}}
```

- Explicitly specify the horizontal and vertical resolution of the output image:

```
pnmtoddif {{[-r|-resolution]}} {{horizontal_dpi}}
{{vertical_dpi}} {{path/to/image.pnm}} > {{path/to/
image.ddif}}
```

pnmtofiasco

Convert a PNM image to a compressed FIASCO file.

More information: https://netpbm.sourceforge.net/doc/pnmtofiasco.html.

- Convert a PNM image to a compressed FIASCO file:

```
pnmtofiasco {{path/to/file.pnm}} > {{path/to/file.fiasco}}
```

- Specify the input files through a pattern:

```
pnmtofiasco {{[-i|--image-name]}} "{{img[01-09+1].pnm}}" >
{{path/to/file.fiasco}}
```

- Specify the compression quality:

```
pnmtofiasco {{[-q|--quality]}} {{quality_level}} {{path/to/
file.pnm}} > {{path/to/file.fiasco}}
```

- Load the options to be used from the specified configuration file:

```
pnmtofiasco {{[-f|--config]}} {{path/to/fiascorc}} {{path/to/
file.pnm}} > {{path/to/file.fiasco}}
```

pnmtofits

This command has been superseded by **pamtofits**.

More information: https://netpbm.sourceforge.net/doc/pnmtofits.html.

- View documentation for the current command:

```
tldr pamtofits
```

pnmtojpeg

Convert a PNM image file to the JPEG/JFIF/EXIF image format.

More information: https://netpbm.sourceforge.net/doc/pnmtojpeg.html.

- Read a PNM image as input and produce a JPEG/JFIF/EXIF image as output:

```
pnmtojpeg {{path/to/file.pnm}} > {{path/to/file.jpg}}
```

- Display version:

```
pnmtojpeg -version
```

pnmtopalm

Convert a PNM image to a Palm bitmap.

More information: https://netpbm.sourceforge.net/doc/pnmtopalm.html.

- Convert a PNM image to a Palm bitmap:

```
pnmtopalm {{path/to/file.pnm}} > {{path/to/file.palm}}
```

- Specify the color depth of the resulting bitmap:

```
pnmtopalm {{[-dep|-depth]}} {{1|2|4|8|16}} {{path/to/
file.pnm}} > {{path/to/file.palm}}
```

- Choose a compression method for the resulting bitmap:

```
pnmtopalm -{{scanline_compression|rle_compression|
packbits_compression}} {{path/to/file.pnm}} > {{path/to/
file.palm}}
```

- Build a custom colormap and include it in the resulting bitmap:

```
pnmtopalm {{[-c|-colormap]}} {{path/to/file.pnm}} > {{path/
to/file.palm}}
```

- Specify the bitmap's density:

```
pnmtopalm {{[-den|-density]}} {{72|108|144|216|288}} {{path/
to/file.pnm}} > {{path/to/file.palm}}
```

pnmtopclxl

Convert a PNM file to an HP LaserJet PCL XL printer stream.

More information: https://netpbm.sourceforge.net/doc/pnmtopclxl.html.

- Convert PNM files to an HP LaserJet PCL XL printer stream:

```
pnmtopclxl {{path/to/input1.pnm path/to/input2.pnm ...}} > {{path/to/output.pclxl}}
```

- Specify the resolution of the image as well as the location of the page from the upper left corner of each image:

```
pnmtopclxl -dpi {{resolution}} {{[-x|-xoffs]}} {{x_offset}} {{[-y|-yoffs]}} {{y_offset}} {{path/to/input1.pnm path/to/input2.pnm ...}} > {{path/to/output.pclxl}}
```

- Generate a duplex printer stream for the specified paper format:

```
pnmtopclxl {{[-du|-duplex]}} {{vertical|horizontal}} {{[-fo|-format]}} {{letter|legal|a3|a4|a5|...}} {{path/to/input1.pnm path/to/input2.pnm ...}} > {{path/to/output.pclxl}}
```

pnmtoplainpnm

This command is an alias of **`pamtopnm -plain`**.

- View documentation for the original command:

  ```
  tldr pamtopnm
  ```

pnmtopng

Convert a PNM image file to PNG image format.

More information: https://netpbm.sourceforge.net/doc/pnmtopng.html.

- Read a PNM image as input and produce a PNG image as output:

```
pnmtopng {{path/to/file.pnm}} > {{path/to/file.png}}
```

- Display version:

```
pnmtopng {{[-vers|-version]}}
```

pnmtopnm

This command is an alias of **pamtopnm**.

- View documentation for the original command:

```
tldr pamtopnm
```

pnmtops

Convert a PNM image to a PostScript file.

More information: https://netpbm.sourceforge.net/doc/pnmtops.html.

- Convert a PNM image to a PS file:

```
pnmtops {{path/to/file.pnm}} > {{path/to/file.ps}}
```

- Specify the dimensions of the output image in inches:

```
pnmtops {{[-imagew|-imagewidth]}} {{imagewidth}} {{[-imageh|-imageheight]}} {{imageheight}} {{path/to/file.pnm}} > {{path/to/file.ps}}
```

- Specify the dimensions of the page the output image resides on in inches:

```
pnmtops {{[-w|-width]}} {{width}} {{[-h|-height]}} {{height}} {{path/to/file.pnm}} > {{path/to/file.ps}}
```

pnmtorast

Convert a PNM file to a Sun rasterfile.

More information: https://netpbm.sourceforge.net/doc/pnmtorast.html.

- Convert a PNM image to a RAST image:

```
pnmtorast {{path/to/input.pnm}} > {{path/to/output.rast}}
```

- Force either `RT_STANDARD` or `RT_BYTE_ENCODED` form for the output:

```
pnmtorast -{{standard|rle}} {{path/to/input.pnm}} > {{path/
to/output.rast}}
```

pnmtorle

Convert a PNM file to an Utah Raster Tools RLE image file.

More information: https://netpbm.sourceforge.net/doc/pnmtorle.html.

- Convert a PNM image to an RLE image:

```
pnmtorle {{path/to/input.pnm}} > {{path/to/output.rle}}
```

- Print PNM header information to `stdout`:

```
pnmtorle {{[-verb|-verbose]}} {{path/to/input.pnm}} > {{path/to/output.rle}}
```

- Include a transparency channel in the output image in which every black pixel is set to fully transparent and every other pixel is set to fully opaque:

```
pnmtorle {{[-a|-alpha]}} {{path/to/input.pnm}} > {{path/to/output.rle}}
```

pnmtosgi

Convert a PNM file to an SGI image file.

More information: https://netpbm.sourceforge.net/doc/pnmtosgi.html.

- Convert a PNM image to an SGI image:

```
pnmtosgi {{path/to/input.pnm}} > {{path/to/output.sgi}}
```

- Disable or enable compression:

```
pnmtosgi -{{verbatim|rle}} {{path/to/input.pnm}} > {{path/to/output.sgi}}
```

- Write the specified string into the SGI image header's `imagename` field:

```
pnmtosgi {{[-i|-imagename]}} {{string}} {{path/to/input.pnm}} > {{path/to/output.sgi}}
```

pnmtosir

Convert a PNM file to a Solitaire Image Recorder file.

More information: https://netpbm.sourceforge.net/doc/pnmtosir.html.

- Convert a PNM image to a SIR image:

```
pnmtosir {{path/to/input.pnm}} > {{path/to/output.sir}}
```

pnmtotiff

This command has been superseded by **pamtotiff**.

More information: https://netpbm.sourceforge.net/doc/pnmtotiff.html.

- View documentation for the current command:

```
tldr pamtotiff
```

pnmtotiffcmyk

Convert a PNM image to a CMYK encoded TIFF.

More information: https://netpbm.sourceforge.net/doc/pnmtotiffcmyk.html.

- Convert a PNM image to a CMYK encoded TIFF:

```
pnmtotiffcmyk {{path/to/input_file.pnm}} > {{path/to/
output_file.tiff}}
```

- Specify the TIFF compression method:

```
pnmtotiffcmyk -{{none|packbits|lzw}} {{path/to/
input_file.pnm}} > {{path/to/output_file.tiff}}
```

- Control the fill order:

```
pnmtotiffcmyk -{{msb2lsb|lsb2msb}} {{path/to/input_file.pnm}}
> {{path/to/output_file.tiff}}
```

pnmtoxwd

Convert a PNM file into an X11 window dump file.

More information: https://netpbm.sourceforge.net/doc/pnmtoxwd.html.

- Convert a PNM image file to XWD:

```
pnmtoxwd {{path/to/input_file.pnm}} > {{path/to/
output_file.xwd}}
```

- Produce the output in the DirectColor format:

```
pnmtoxwd {{[-d|-directcolor]}} {{path/to/input_file.pnm}} >
{{path/to/output_file.xwd}}
```

- Set the color depth of the output to b bits:

```
pnmtoxwd {{[-ps|-pseudodepth]}} {{b}} {{path/to/
input_file.pnm}} > {{path/to/output_file.xwd}}
```

pnpm audit

Scan project dependencies.

Check for known security issues with the installed packages.

More information: https://pnpm.io/cli/audit.

- Identify vulnerabilities in the project:

```
pnpm audit
```

- Automatically fix vulnerabilities:

```
pnpm audit fix
```

- Generate a security report in JSON format:

```
pnpm audit --json > {{path/to/audit-report.json}}
```

- Audit only dev dependencies:

```
pnpm audit {{[-D|--dev]}}
```

- Audit only production dependencies:

```
pnpm audit {{[-P|--prod]}}
```

- Exclude optional dependencies from the audit:

```
pnpm audit --no-optional
```

- Ignore registry errors during the audit process:

```
pnpm audit --ignore-registry-errors
```

- Filter advisories by severity (low, moderate, high, critical):

```
pnpm audit --audit-level {{severity}}
```

pnpm outdated

Check for outdated packages.

The check can be limited to a subset of the installed packages by providing arguments (patterns are supported).

More information: https://pnpm.io/cli/outdated.

- Check for outdated packages:

  ```
  pnpm outdated
  ```

- Check for outdated dependencies found in every workspace package:

  ```
  pnpm outdated {{[-r|--recursive]}}
  ```

- Filter outdated packages using a package selector:

  ```
  pnpm outdated --filter {{package_selector}}
  ```

- List outdated packages globally:

  ```
  pnpm outdated {{[-g|--global]}}
  ```

- Print details of outdated packages:

  ```
  pnpm outdated --long
  ```

- Print outdated dependencies in a specific format:

  ```
  pnpm outdated --format {{format}}
  ```

- Print only versions that satisfy specifications in `package.json`:

  ```
  pnpm outdated --compatible
  ```

- Check only outdated dev dependencies:

  ```
  pnpm outdated {{[-D|--dev]}}
  ```

pnpm

Fast, disk space efficient package manager for Node.js.

Manage Node.js projects and their module dependencies.

More information: https://pnpm.io/pnpm-cli.

- Create a `package.json` file:

```
pnpm init
```

- Download all the packages listed as dependencies in `package.json`:

```
pnpm {{[i|install]}}
```

- Download a specific version of a package and add it to the list of dependencies in `package.json`:

```
pnpm add {{module_name}}@{{version}}
```

- Download a package and add it to the list of dev dependencies in `package.json`:

```
pnpm add {{module_name}} {{[-D|--save-dev]}}
```

- Download a package and install it globally:

```
pnpm add {{module_name}} {{[-g|--global]}}
```

- Uninstall a package and remove it from the list of dependencies in `package.json`:

```
pnpm {{[rm|remove]}} {{module_name}}
```

- Print a tree of locally installed modules:

```
pnpm {{[ls|list]}}
```

- List top-level globally installed modules:

```
pnpm {{[ls|list]}} {{[-g|--global]}} --depth {{0}}
```

pnpx

Directly execute binaries from npm packages, using **`pnpm`** instead of **`npm`**.

Note: This command is deprecated! Use **`pnpm exec`** and **`pnpm dlx`** instead.

More information: https://cuyl.github.io/pnpm.github.io/pnpx-cli.

- Execute the binary from a given `npm` module:

  ```
  pnpx {{module_name}}
  ```

- Execute a specific binary from a given `npm` module, in case the module has multiple binaries:

  ```
  pnpx --package {{package_name}} {{module_name}}
  ```

- Display help:

  ```
  pnpx --help
  ```

pocount

Translate Toolkit utility to get translation progress from file, supporting several formats.

More information: https://docs.translatehouse.org/projects/translate-toolkit/en/latest/commands/pocount.html.

- Print a colorful table with the translation progress of a file:

```
pocount {{path/to/file.po}}
```

- Print translation progress of various files, one line per file:

```
pocount --short {{translation_*.ts}}
```

- Generate a CSV file with the translation progress of various files:

```
pocount --csv {{translation_*.ts}} > {{path/to/translation_progress.csv}}
```

podman build

Daemonless tool for building container images.

Podman provides a Docker-CLI comparable command-line. Simply put: **`alias docker=podman`**.

More information: https://docs.podman.io/en/latest/markdown/podman-build.1.html.

- Create an image using a `Dockerfile` or `Containerfile` in the specified directory:

  ```
  podman build {{path/to/directory}}
  ```

- Create an image with a specified tag:

  ```
  podman build --tag {{image_name:version}} {{path/to/directory}}
  ```

- Create an image from a non-standard file:

  ```
  podman build --file {{Containerfile.different}} .
  ```

- Create an image without using any previously cached images:

  ```
  podman build --no-cache {{path/to/directory}}
  ```

- Create an image suppressing all output:

  ```
  podman build --quiet {{path/to/directory}}
  ```

podman-compose

Run and manage Compose Specification container definition.

More information: https://github.com/containers/podman-compose.

- List all running containers:

  ```
  podman-compose ps
  ```

- Create and start all containers in the background using a local `docker-compose.yml`:

  ```
  podman-compose up {{[-d|--detach]}}
  ```

- Start all containers, building if needed:

  ```
  podman-compose up --build
  ```

- Start all containers using an alternate compose file:

  ```
  podman-compose {{[-f|--file]}} {{path/to/file.yaml}} up
  ```

- Stop all running containers:

  ```
  podman-compose stop
  ```

- Remove all containers, networks, and volumes:

  ```
  podman-compose down {{[-v|--volumes]}}
  ```

- Follow logs for a container (omit all container names):

  ```
  podman-compose logs {{[-f|--follow]}} {{container_name}}
  ```

- Run a one-time command in a service with no ports mapped:

  ```
  podman-compose run {{service_name}} {{command}}
  ```

podman image

Manage Docker images.

See also: **`podman build`**, **`podman import`**, **`podman pull`**.

More information: https://docs.podman.io/en/latest/markdown/podman-image.1.html.

- List local Docker images:

  ```
  podman image ls
  ```

- Delete unused local Docker images:

  ```
  podman image prune
  ```

- Delete all unused images (not just those without a tag):

  ```
  podman image prune --all
  ```

- Show the history of a local Docker image:

  ```
  podman image history {{image}}
  ```

podman images

Manage Podman images.

More information: https://docs.podman.io/en/latest/markdown/podman-images.1.html.

- List all Podman images:

```
podman images
```

- List all Podman images including intermediates:

```
podman images --all
```

- List the output in quiet mode (only numeric IDs):

```
podman images --quiet
```

- List all Podman images not used by any container:

```
podman images --filter dangling=true
```

- List images that contain a substring in their name:

```
podman images "{{*image|image*}}"
```

podman login

Log in to a container registry.

Note: The default authfile path on Linux is **`$XDG_RUNTIME_DIR/containers/auth.json`**, which is usually stored in a **`tmpfs`** (in RAM).

More information: https://docs.podman.io/en/latest/markdown/podman-login.1.html.

- Log in to a registry (non-persistent on Linux; persistent on Windows/macOS):

```
podman login {{registry.example.org}}
```

- Log in to a registry persistently on Linux:

```
podman login --authfile $HOME/.config/containers/auth.json {{registry.example.org}}
```

- Log in to an insecure (HTTP) registry:

```
podman login --tls-verify=false {{registry.example.org}}
```

podman machine

Create and manage virtual machines running Podman.

Included with Podman version 4 or greater.

More information: https://docs.podman.io/en/latest/markdown/podman-machine.1.html.

- List existing machines:

```
podman machine ls
```

- Create a new default machine:

```
podman machine init
```

- Create a new machine with a specific name:

```
podman machine init {{name}}
```

- Create a new machine with different resources:

```
podman machine init --cpus={{4}} --memory={{4096}} --disk-size={{50}}
```

- Start or stop a machine:

```
podman machine {{start|stop}} {{name}}
```

- Connect to a running machine via SSH:

```
podman machine ssh {{name}}
```

- Inspect information about a machine:

```
podman machine inspect {{name}}
```

podman ps

List Podman containers.

More information: https://docs.podman.io/en/latest/markdown/podman-ps.1.html.

- List currently running Podman containers:

```
podman ps
```

- List all Podman containers (running and stopped):

```
podman ps --all
```

- Show the latest created container (includes all states):

```
podman ps --latest
```

- Filter containers that contain a substring in their name:

```
podman ps --filter "name={{name}}"
```

- Filter containers that share a given image as an ancestor:

```
podman ps --filter "ancestor={{image}}:{{tag}}"
```

- Filter containers by exit status code:

```
podman ps --all --filter "exited={{code}}"
```

- Filter containers by status (created, running, removing, paused, exited and dead):

```
podman ps --filter "status={{status}}"
```

- Filter containers that mount a specific volume or have a volume mounted in a specific path:

```
podman ps --filter "volume={{path/to/directory}}" --format
"table {{.ID}}\t{{.Image}}\t{{.Names}}\t{{.Mounts}}"
```

podman rmi

Remove Podman images.

More information: https://docs.podman.io/en/latest/markdown/podman-rmi.1.html.

- Remove one or more images given their names:

```
podman rmi {{image:tag image2:tag ...}}
```

- Force remove an image:

```
podman rmi --force {{image}}
```

- Remove an image without deleting untagged parents:

```
podman rmi --no-prune {{image}}
```

- Display help:

```
podman rmi
```

podman run

Run a command in a new Podman container.

More information: https://docs.podman.io/en/latest/markdown/podman-run.1.html.

- Run command in a new container from a tagged image:

```
podman run {{image:tag}} {{command}}
```

- Run command in a new container in background and display its ID:

```
podman run --detach {{image:tag}} {{command}}
```

- Run command in a one-off container in interactive mode and pseudo-TTY:

```
podman run --rm --interactive --tty {{image:tag}} {{command}}
```

- Run command in a new container with passed environment variables:

```
podman run --env '{{variable}}={{value}}' --env {{variable}} {{image:tag}} {{command}}
```

- Run command in a new container with bind mounted volumes:

```
podman run --volume /{{path/to/host_path}}:/{{path/to/container_path}} {{image:tag}} {{command}}
```

- Run command in a new container with published ports:

```
podman run --publish {{host_port}}:{{container_port}} {{image:tag}} {{command}}
```

- Run command in a new container overwriting the entrypoint of the image:

```
podman run --entrypoint {{command}} {{image:tag}}
```

- Run command in a new container connecting it to a network:

```
podman run --network {{network}} {{image:tag}}
```

podman

Simple management tool for pods, containers and images.

Podman provides a Docker-CLI comparable command-line. Simply put: **`alias docker=podman`**.

More information: https://github.com/containers/podman/blob/main/commands-demo.md.

- List all containers (both running and stopped):

  ```
  podman ps --all
  ```

- Create a container from an image, with a custom name:

  ```
  podman run --name {{container_name}} {{image}}
  ```

- Start or stop an existing container:

  ```
  podman {{start|stop}} {{container_name}}
  ```

- Pull an image from a registry (defaults to Docker Hub):

  ```
  podman pull {{image}}
  ```

- Display the list of already downloaded images:

  ```
  podman images
  ```

- Open a shell inside an already running container:

  ```
  podman exec --interactive --tty {{container_name}} {{sh}}
  ```

- Remove a stopped container:

  ```
  podman rm {{container_name}}
  ```

- Display the logs of one or more containers and follow log output:

  ```
  podman logs --follow {{container_name}} {{container_id}}
  ```

poetry about

Get information about Poetry, an application to manage Python packages and dependencies.

See also: **`asdf`**.

More information: https://python-poetry.org/docs/cli/#about.

- Display global information about Poetry, including the current version:

```
poetry about
```

poetry add

Add required packages to the **`pyproject.toml`** file in Poetry.

See also: **`asdf`**.

More information: https://python-poetry.org/docs/cli/#add.

- Add required packages:

```
poetry add {{package_name}}
```

- Add required packages to a specific group of dependencies:

```
poetry add {{package_name}} --group {{group_name}}
```

- Add a specific version of a package:

```
poetry add {{package_name}}=={{version}}
```

- Add a specific version of a package equal to or earlier than a given version:

```
poetry add {{package_name}}<={{version}}
```

- Add a specific version of a package equal to or later than a given version:

```
poetry add {{package_name}}>={{version}}
```

poetry build

Build a Poetry package as a tarball and a wheel.

More information: https://python-poetry.org/docs/cli/#build.

- Build a tarball and a wheel for the project:

```
poetry build
```

- Build a wheel package:

```
poetry build {{[-f|--format]}} wheel
```

- Build a source distribution (sdist):

```
poetry build {{[-f|--format]}} sdist
```

- Clean the output directory before building:

```
poetry build --clean
```

- Set the output directory:

```
poetry build {{[-o|--output]}} {{path/to/directory}}
```

poetry cache

Manage Poetry's cache.

See also: **`asdf`**.

More information: https://python-poetry.org/docs/cli/#cache.

- Display Poetry's available caches:

```
poetry cache list
```

- Remove all packages from a cache (e.g.: PyPI):

```
poetry cache clear PyPI --all
```

- Remove a specific package from a cache (Note: must be in format `cache:package:version`):

```
poetry cache clear {{pypi}}:{{requests}}:{{2.24.0}}
```

poetry check

Manage Poetry file validation and consistency.

See also: **`asdf`**.

More information: https://python-poetry.org/docs/cli/#check.

- Check validation and consistency between `pyproject.toml` and `poetry.lock` for Poetry:

```
poetry check
```

- Verify that `poetry.lock` exists:

```
poetry check --lock
```

- Fail if warnings are reported:

```
poetry check --strict
```

poetry config

Edit poetry config settings and repositories.

More information: https://python-poetry.org/docs/cli/#config.

- List current configuration:

```
poetry config --list
```

- Remove the a previously set setting:

```
poetry config {{config_key}} --unset
```

- See the value of a specific setting:

```
poetry config {{config_key}}
```

- Change or add a new configuration setting by passing a value after the setting's name:

```
poetry config {{config_key}} {{config_value}}
```

- Migrate outdated configuration settings:

```
poetry config --migrate
```

- Set/Get settings that are specific to a project:

```
poetry config --local
```

poetry-debug

Debug issues with your Poetry project.

See also: **asdf**.

More information: https://python-poetry.org/docs/cli/#debug.

- Show debug info about Poetry and your project's virtual environment:

```
poetry debug info
```

- Attempt to resolve your project dependencies:

```
poetry debug resolve
```

- Show supported packaging tags for your project's active virtual environment:

```
poetry debug tags
```

poetry-env

Manage virtual environments associated with a Poetry project.

See also: **`asdf`**.

More information: https://python-poetry.org/docs/cli/#env.

- Print the command to activate a virtual environment:

```
poetry env activate
```

- Display information about the current environment:

```
poetry env info
```

- Display the path of the current environment:

```
poetry env info {{[-p|--path]}}
```

- Display the path of the current environment's Python executable:

```
poetry env info {{[-e|--executable]}}
```

- List all virtual environments associated with the current project (optionally showing the full path):

```
poetry env list --full-path
```

- Remove specific or all virtual environments associated with the current project:

```
poetry env remove python {{path/to/executable|
environment_name}} | poetry env remove --all
```

- Activate or create a virtual environment for the project using the specified Python executable:

```
poetry env use python {{path/to/executable}}
```

poetry export

Export Poetry's lock file to other formats.

Provided by the Export Poetry Plugin.

More information: https://github.com/python-poetry/poetry-plugin-export#usage.

- Export dependencies to a `requirements.txt` file:

```
poetry export {{[-o|--output]}} {{requirements.txt}}
```

- Export dependencies including development dependencies:

```
poetry export {{[-o|--output]}} {{requirements-dev.txt}} --dev
```

- Export dependencies without hashes:

```
poetry export {{[-o|--output]}} {{requirements.txt}} --without-hashes
```

- Export dependencies for a specific format:

```
poetry export {{[-o|--output]}} {{requirements.txt}} {{[-f|--format]}} {{requirements.txt}}
```

- Export only specific dependency groups:

```
poetry export {{[-o|--output]}} {{requirements.txt}} --only {{main}}
```

- Display help:

```
poetry export {{[-h|--help]}}
```

poetry help

Display global help, or help for a specific **`poetry`** command.

More information: https://python-poetry.org/docs/cli/#help.

- Display global help:

```
poetry help
```

- Display help for a specific command:

```
poetry help {{config|show|...}}
```

poetry init

Create a basic **`pyproject.toml`** file interactively.

More information: https://python-poetry.org/docs/cli/#init.

- Create a `pyproject.toml` file interactively:

```
poetry init
```

- Create a `pyproject.toml` file with prefilled values:

```
poetry init --name {{package_name}} --author "{{author_name <email@example.com>}}"
```

- Create a `pyproject.toml` file without interaction (using defaults):

```
poetry init {{[-n|--no-interaction]}}
```

- Create a `pyproject.toml` file and add a dependency:

```
poetry init --dependency {{package_name}}
```

- Create a `pyproject.toml` file and add a development dependency:

```
poetry init --dev-dependency {{package_name}}
```

- Display help:

```
poetry init {{[-h|--help]}}
```

poetry install

Install all dependencies for a Python project as defined in the pyproject.toml file.

More information: https://python-poetry.org/docs/cli/#install.

- Install dependencies:

```
poetry install
```

- Skip installing the project itself as a dependency:

```
poetry install --no-root
```

- Install only production dependencies:

```
poetry install --without dev
```

- Install optional dependenciy groups:

```
poetry install --with test,docs
```

poetry list

List available Poetry commands.

See also: **`asdf`**.

More information: https://python-poetry.org/docs/cli/#list.

- List all available Poetry commands:

```
poetry list
```

- List commands for a specific namespace:

```
poetry list {{namespace}}
```

poetry lock

Lock dependencies in **`pyproject.toml`** (without installing them).

See also: **`asdf`**.

More information: https://python-poetry.org/docs/cli/#lock.

- Lock dependencies from the current directory's `pyproject.toml` file:

  ```
  poetry lock
  ```

- Regenerate the existing lock file:

  ```
  poetry lock --regenerate
  ```

poetry-python

Manage Python versions through Poetry.

See also: **`asdf`**.

More information: https://python-poetry.org/docs/cli/#python.

- Install the specified Python version:

```
poetry python install {{3.13.1}}
```

- List all Python versions managed by System or Poetry:

```
poetry python list
```

- List all Python versions managed by Poetry:

```
poetry python list --managed
```

- Remove the specified Python version (if managed by Poetry):

```
poetry python remove {{3.13.1}}
```

poetry-self

Manage the Poetry installation/runtime environment itself.

These commands reference **`pyproject.toml`** and **`poetry.lock`** files in your Poetry configuration directory.

See also: **`asdf`**.

More information: https://python-poetry.org/docs/cli/#self.

- Install a package:

```
poetry self add {{package_name}}
```

- Install dependencies from the Poetry installation's `pyproject.toml` file:

```
poetry self install
```

- Lock dependencies from the Poetry installation's `pyproject.toml` file:

```
poetry self lock
```

- Remove a package:

```
poetry self remove {{package_name}}
```

- List all installed packages:

```
poetry self show
```

- List all installed plugins:

```
poetry self show plugins
```

- Sync the runtime environment with the Poetry installation's `poetry.lock` file:

```
poetry self sync
```

- Update dependencies from the Poetry installation's `pyproject.toml` file:

```
poetry self update
```

poetry shell

Spawn a shell within the virtual environment.

Note: The **`shell`** command was moved to a plugin: **`poetry-plugin-shell`**.

More information: https://python-poetry.org/docs/cli/#shell.

- Spawn a shell within the virtual environment:

  ```
  poetry shell
  ```

poetry show

Show details of packages in your Poetry project.

See also: **`asdf`**.

More information: https://python-poetry.org/docs/cli/#show.

- Display all packages:

```
poetry show
```

- Show details of a specific package:

```
poetry show {{package_name}}
```

- Show details as a dependency tree:

```
poetry show {{[-t|--tree]}}
```

- Only show top-level packages (those explicitly defined in `pyproject.toml`):

```
poetry show {{[-T|--top-level]}}
```

- Show outdated packages:

```
poetry show {{[-o|--outdated]}}
```

- Show the latest versions for all packages:

```
poetry show {{[-l|--latest]}}
```

- Exclude a specific dependency group/s:

```
poetry show --without {{group1,group2,...}}
```

- Only show a specific dependency group/s:

```
poetry show --only {{group1,group2,...}}
```

poetry source

Add source configurations to your Poetry project.

See also: **`asdf`**.

More information: https://python-poetry.org/docs/cli/#source.

- Add a source configuration:

```
poetry source add {{source_name}} {{source_url}}
```

- Set the priority of a source:

```
poetry source add --priority {{primary|supplemental|
explicit}} {{source_name}} {{source_url}}
```

- Display info for all sources:

```
poetry source show
```

- Display info for a specific source:

```
poetry source show {{source_name}}
```

- Remove a source from your `pyproject.toml` file:

```
poetry source remove {{source_name}}
```

poetry sync

Syncs your project's environment with the **`poetry.lock`** file.

More information: https://python-poetry.org/docs/cli/#sync.

- Sync your project's environment with the `poetry.lock` file:

  ```
  poetry sync
  ```

- Exclude one or more dependency groups for the installation:

  ```
  poetry sync --without {{test|docs|...}}
  ```

- Select optional dependency groups:

  ```
  poetry sync --with {{test|docs|...}}
  ```

- Install all dependency groups including optional groups:

  ```
  poetry sync --all-groups
  ```

- Install specific dependency groups:

  ```
  poetry sync --only {{test|docs|...}}
  ```

- Install project without dependencies:

  ```
  poetry sync --only-root
  ```

- Specify extras to install:

  ```
  poetry sync {{[-E|--extras]}}
  ```

- Skip the defaulted package installation for your project:

  ```
  poetry sync --no-root
  ```

poetry version

Manage Poetry project version.

Assumes the following project stages: **`patch`**, **`minor`**, **`major`**, **`prepatch`**, **`preminor`**, **`premajor`**, **`prerelease`**.

See also: **`asdf`**.

More information: https://python-poetry.org/docs/cli/#version.

- Output the current version:

```
poetry version {{[-s|--short]}}
```

- Set project to a specific phase:

```
poetry version {{stage}}
```

- Increment the project to the next prerelease phase:

```
poetry version --next-phase
```

- Test project stage function without writing to `pyproject.toml`:

```
poetry version {{stage}} --dry-run
```

poetry

Manage Python packages and dependencies.

Some subcommands such as **`about`**, **`check`**, **`env`**, etc. have their own usage documentation.

See also: **`asdf`**, **`pipenv`**, **`hatch`**.

More information: https://python-poetry.org/docs/cli/.

- Create a new Poetry project in the directory with a specific name:

```
poetry new {{project_name}}
```

- Install and add a dependency and its sub-dependencies to the `pyproject.toml` file in the current directory:

```
poetry add {{dependency}}
```

- Install the project dependencies using the `pyproject.toml` file in the current directory:

```
poetry install
```

- Interactively (append `-n` for non-interactively) initialize the current directory as a new Poetry project:

```
poetry init
```

- Get the latest version of all dependencies and update `poetry.lock`:

```
poetry update
```

- Execute a command inside the project's virtual environment:

```
poetry run {{command}}
```

- Bump the version of the project in `pyproject.toml`:

```
poetry version {{patch|minor|major|prepatch|preminor|
premajor|prerelease}}
```

- Spawn a shell within the project's virtual environment (for versions below 2.0, use `poetry shell`):

```
eval "$(poetry env activate)"
```

pokeget

Display sprites of Pokemon in your terminal.

More information: https://github.com/talwat/pokeget-rs.

- Print a sprite of a given pokemon:

```
pokeget {{pokemon_name}}
```

- Print Mr. Mime (note the use of `-` instead of spaces):

```
pokeget mr-mime
```

- Print Mega Gengar:

```
pokeget gengar {{[-m|--mega]}}
```

- Print a random shiny Pokemon:

```
pokeget random {{[-s|--shiny]}}
```

- Print Alolan Meowth, without printing the Pokemon's name:

```
pokeget meowth {{[-a|--alolan]}} --hide-name
```

- Print a random Pokemon with 1/4096 chance to be shiny:

```
((RANDOM%4096 == 0)) && pokeget random --shiny || pokeget
random
```

polybar-msg

Control **`polybar`** using inter-process-messaging (IPC).

Note: IPC is disabled by default and can be enabled by setting **`enable-ipc = true`** in the Polybar config.

More information: https://polybar.rtfd.io/en/stable/user/ipc.html.

- Quit the bar:

  ```
  polybar-msg cmd quit
  ```

- Restart the bar in-place:

  ```
  polybar-msg cmd restart
  ```

- Hide the bar (does nothing if the bar is already hidden):

  ```
  polybar-msg cmd hide
  ```

- Show the bar again (does nothing if the bar is not hidden):

  ```
  polybar-msg cmd show
  ```

- Toggle between hidden/visible:

  ```
  polybar-msg cmd toggle
  ```

- Execute a module action (the data string is optional):

  ```
  polybar-msg action "#{{module_name}}.{{action_name}}.
  {{data_string}}"
  ```

- Only send messages to a specific Polybar instance (all instances by default):

  ```
  polybar-msg -p {{pid}} {{cmd|action}} {{payload}}
  ```

polybar

A fast and easy-to-use status bar.

More information: https://github.com/polybar/polybar/wiki.

- Start Polybar (the bar name is optional if only one bar is defined in the config):

```
polybar {{bar_name}}
```

- Start Polybar with the specified config:

```
polybar --config={{path/to/config.ini}} {{bar_name}}
```

- Start Polybar and reload the bar when the configuration file is modified:

```
polybar --reload {{bar_name}}
```

pop

Send emails from your terminal.

More information: https://github.com/charmbracelet/pop.

- Launch the Text-based User Interface:

```
pop
```

- Send an email using the content of a Markdown file as body:

```
pop < {{path/to/message.md}} --from {{me@example.com}} --to
{{you@example.com}} --subject "{{On the Subject of
Ducks...}}" --attach {{path/to/attachment}}
```

- Display help:

```
pop --help
```

popd

Remove a directory placed on the directory stack via the pushd shell built-in.

See also: **`pushd`** to place a directory on the stack and **`dirs`** to display directory stack contents.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-popd.

- Remove the top directory from the stack and cd to it:

```
popd
```

- Remove the Nth directory (starting from zero to the left from the list printed with `dirs`):

```
popd +N
```

- Remove the Nth directory (starting from zero to the right from the list printed with `dirs`):

```
popd -N
```

- Remove the 1st directory (starting from zero to the left from the list printed with `dirs`):

```
popd -n
```

popeye

Report potential issues with Kubernetes deployment manifests.

More information: https://github.com/derailed/popeye.

- Scan the current Kubernetes cluster:

  ```
  popeye
  ```

- Scan a specific namespace:

  ```
  popeye {{[-n|--namespace]}} {{namespace}}
  ```

- Scan specific Kubernetes context:

  ```
  popeye --context {{context}}
  ```

- Use a spinach configuration file for scanning:

  ```
  popeye {{[-f|--file]}} {{spinach.yaml}}
  ```

postcss

Transform styles with JS plugins.

More information: https://postcss.org.

- Parse and transform a CSS file:

```
postcss {{path/to/file}}
```

- Parse and transform a CSS file and output to a specific file:

```
postcss {{path/to/file}} --output {{path/to/file}}
```

- Parse and transform a CSS file and output to a specific directory:

```
postcss {{path/to/file}} --dir {{path/to/directory}}
```

- Parse and transform a CSS file in-place:

```
postcss {{path/to/file}} --replace
```

- Specify a custom PostCSS parser:

```
postcss {{path/to/file}} --parser {{parser}}
```

- Specify a custom PostCSS syntax:

```
postcss {{path/to/file}} --syntax {{syntax}}
```

- Watch for changes to a CSS file:

```
postcss {{path/to/file}} --watch
```

- Display help:

```
postcss --help
```

postgres

Run the PostgreSQL database server.

More information: https://www.postgresql.org/docs/current/app-postgres.html.

- Start the server on a specific port (defaults to 5432):

```
postgres -p {{5433}}
```

- Set a runtime parameter (short form):

```
postgres -c {{shared_buffers=128MB}}
```

- Set a runtime parameter (long form):

```
postgres --{{shared-buffers}}={{128MB}}
```

- Start in single-user mode for a specific database (defaults to the user name):

```
postgres --single -D {{path/to/datadir}} {{my_database}}
```

powershell

This command may be mistaken as the cross-platform version of PowerShell (formerly known as PowerShell Core), which uses **`pwsh`** instead of **`powershell`**.

The original **`powershell`** command in Windows is still available to use the legacy Windows version of PowerShell (version 5.1 and below).

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/about/about_pwsh.

- View the documentation for the command referring to the latest, cross-platform version of PowerShell (version 6 and above):

  ```
  tldr pwsh
  ```

- View the documentation for the command referring to the legacy Windows PowerShell (version 5.1 and below):

  ```
  tldr powershell {{[-p|--platform]}} windows
  ```

ppmbrighten

This command has been superseded by **pambrighten**.

More information: https://netpbm.sourceforge.net/doc/ppmbrighten.html.

- View documentation for the current command:

```
tldr pambrighten
```

ppmchange

Change all pixels of one color in a PPM image to another color.

More information: https://netpbm.sourceforge.net/doc/ppmchange.html.

- Exchange the first color in each `oldcolor` - `newcolor` pair with the second color:

```
ppmchange {{oldcolor1 newcolor1 oldcolor2 newcolor2 ...}}
{{path/to/input.ppm}} > {{path/to/output.ppm}}
```

- Specify how similar colors must be in order to be considered the same:

```
ppmchange -closeness {{percentage}} {{oldcolor1 newcolor1
oldcolor2 newcolor2 ...}} {{path/to/input.ppm}} > {{path/to/
output.ppm}}
```

- Replace all pixels not specified in the arguments by a color:

```
ppmchange {{[-r|-remainder]}} {{color}} {{oldcolor1 newcolor1
oldcolor2 newcolor2 ...}} {{path/to/input.ppm}} > {{path/to/
output.ppm}}
```

ppmcie

Draw a CIE color chart as a PPM image.

More information: https://netpbm.sourceforge.net/doc/ppmcie.html.

- Draw a CIE color chart using the REC709 color system as a PPM image:

```
ppmcie > {{path/to/output.ppm}}
```

- Specify the color system to be used:

```
ppmcie -{{cie|ebu|hdtv|ntsc|smpte}} > {{path/to/output.ppm}}
```

- Specify the location of the individual illuminants:

```
ppmcie -{{red|green|blue}} {{xpos ypos}} > {{path/to/
output.ppm}}
```

- Do not dim the area outside the Maxwell triangle:

```
ppmcie {{[-f|-full]}} > {{path/to/output.ppm}}
```

ppmcolormask

Produce a mask of areas of a certain color in a PPM image.

More information: https://netpbm.sourceforge.net/doc/ppmcolormask.html.

- Produce a mask of areas of a certain color in the specified PPM image:

```
ppmcolormask {{[-c|-color]}} {{red,blue}} {{path/to/
input.ppm}} > {{path/to/output.pbm}}
```

ppmdim

Dim a PPM image.

Partly superseded by **`pamfunc -multiplier`**.

More information: https://netpbm.sourceforge.net/doc/ppmdim.html.

- Dim the specified PPM image by dimfactor:

```
ppmdim {{0.6}} {{path/to/input.ppm}} > {{path/to/output.ppm}}
```

ppmdist

Produce a grayscale version of a PPM image.

More information: https://netpbm.sourceforge.net/doc/ppmdist.html.

- Produce a grayscale version of the specified PPM image:

```
ppmdist {{path/to/input.ppm}} > {{path/to/output.pgm}}
```

- Use the specified method to map colors to graylevels:

```
ppmdist -{{frequency|intensity}} {{path/to/input.ppm}} >
{{path/to/output.pgm}}
```

ppmdither

Reduce the number of colors in an image by applying dithering.

More information: https://netpbm.sourceforge.net/doc/ppmdither.html.

- Read a PPM image, apply dithering and save it to a file:

```
ppmdither {{path/to/image.ppm}} > {{path/to/file.ppm}}
```

- Specify the desired number of shades for each primary color:

```
ppmdither {{[-r|-red]}} {{2}} {{[-g|-green]}} {{3}} {{[-b|-blue]}} {{2}} {{path/to/image.ppm}} > {{path/to/file.ppm}}
```

- Specify the dimensions of the dithering matrix:

```
ppmdither {{[-d|-dim]}} {{2}} {{path/to/image.ppm}} > {{path/to/file.ppm}}
```

ppmdraw

Draw lines, text and more on a PPM image by executing a script.

Documentation on the utilized scripting language can be found by following the link below.

More information: https://netpbm.sourceforge.net/doc/ppmdraw.html.

- Draw on the specified PPM image by executing the supplied script:

```
ppmdraw -script '{{setpos 50 50; text_here
"hello!"; }}' {{path/to/image.pnm}} > {{path/to/output.pnm}}
```

- Draw on the specified PPM image by executing the script in the specified file:

```
ppmdraw -scriptfile {{path/to/script}} {{path/to/image.pnm}}
> {{path/to/output.pnm}}
```

ppmfade

Generate a transition between two PPM images.

More information: https://netpbm.sourceforge.net/doc/ppmfade.html.

- Generate a transition between two PPM images ([f]irst and [l]ast) using the specified effect:

```
ppmfade -f {{path/to/image1.ppm}} -l {{path/to/image2.ppm}} -{{mix|spread|shift|relief|oil|...}}
```

- Generate a transition starting with the specified image and ending in a solid black image:

```
ppmfade -f {{path/to/image.ppm}} -{{mix|spread|shift|relief|oil|...}}
```

- Generate a transition starting with a solid black image and ending with the specified image:

```
ppmfade -l {{path/to/image.ppm}} -{{mix|spread|shift|relief|oil|...}}
```

- Store the resulting images in files named `base.NNNN.ppm` where `NNNN` is a increasing number:

```
ppmfade -f {{path/to/image1.ppm}} -l {{path/to/image2.ppm}} -{{mix|spread|shift|relief|oil|...}} -base {{base}}
```

ppmflash

Brighten a PPM image file.

More information: https://netpbm.sourceforge.net/doc/ppmflash.html.

- Generate a PPM image as output that is `flashfactor` times brighter than the input PPM image:

```
ppmflash {{flashfactor}} {{path/to/file.ppm}} > {{path/to/file.ppm}}
```

- Display version:

```
ppmflash {{[-v|-version]}}
```

ppmforge

Generate fractals resembling clouds, planets and starry skies.

More information: https://netpbm.sourceforge.net/doc/ppmforge.html.

- Generate an image of a planet:

```
ppmforge > {{path/to/image.ppm}}
```

- Generate an image of clouds or the night sky:

```
ppmforge -{{night|clouds}} > {{path/to/image.ppm}}
```

- Use a custom mesh size and dimension for fractal generation and specify the dimensions of the output:

```
ppmforge {{[-m|-mesh]}} {{512}} {{[-d|-dimension]}} {{2.5}} {{[-x|-xsize]}} {{1000}} {{[-y|-ysize]}} {{1000}} > {{path/to/image.ppm}}
```

- Control the tilt and the angle from which the generated planet is illuminated:

```
ppmforge {{[-t|-tilt]}} {{-15}} {{[-ho|-hour]}} {{12}} > {{path/to/image.ppm}}
```

ppmglobe

Generate strips of an image suitable to be glued onto a sphere.

See also: **pnmmercator**.

More information: https://netpbm.sourceforge.net/doc/ppmglobe.html.

- Transform an image to strips that can be cut out and glues onto a sphere:

```
ppmglobe {{number_of_strips}} {{path/to/image.ppm}} > {{path/to/output.ppm}}
```

- Use the specified color for the areas between the strips:

```
ppmglobe {{[-b|-background]}} {{red}} {{number_of_strips}} {{path/to/image.ppm}} > {{path/to/output.ppm}}
```

ppmhist

Print a histogram of the colors present in a PPM image.

See also: **pgmhist**.

More information: https://netpbm.sourceforge.net/doc/ppmhist.html.

- Generate the histogram for human reading:

```
ppmhist {{[-nom|-nomap]}} {{path/to/image.ppm}}
```

- Generate a PPM file of the colormap for the image, with the color histogram as comments:

```
ppmhist {{[-m|-map]}} {{path/to/image.ppm}}
```

- Display version:

```
ppmhist {{[-v|-version]}}
```

ppmlabel

Add text to a PPM image.

More information: https://netpbm.sourceforge.net/doc/ppmlabel.html.

- Add text to a PPM image at the specified location:

```
ppmlabel -x {{pos_x}} -y {{pos_y}} {{[-t|-text]}} {{text}}
{{path/to/input_file.ppm}} > {{path/to/output_file.ppm}}
```

- Add multiple texts at different locations:

```
ppmlabel -x {{pos_x1}} -y {{pos_y1}} {{[-t|-text]}} {{text1}}
-x {{pos_x2}} -y {{pos_y2}} {{[-t|-text]}} {{text2}} {{path/
to/input_file.ppm}} > {{path/to/output_file.ppm}}
```

- Specify the line color, the background color, the tilt and the size of the added text:

```
ppmlabel -x {{pos_x}} -y {{pos_y}} {{[-c|-color]}}
{{line_color}} {{[-b|-background]}} {{background_color}} {{[-
a|-angle]}} {{tilt}} {{[-s|-size]}} {{size}} {{[-t|-text]}}
{{text}} {{path/to/input_file.ppm}} > {{path/to/
output_file.ppm}}
```

ppmmake

Create a PPM image of a specified color and dimensions.

More information: https://netpbm.sourceforge.net/doc/ppmmake.html.

- Create a PPM image of the specified color and dimensions:

```
ppmmake {{color}} {{width}} {{height}} > {{path/to/
output_file.ppm}}
```

ppmmix

Blend together two PPM images.

More information: https://netpbm.sourceforge.net/doc/ppmmix.html.

- Blend the specified PPM images using fadefactor to control the weight of each image:

```
ppmmix {{fadefactor}} {{path/to/input_file1.ppm path/to/
input_file2.ppm ...}} > {{path/to/output_file.ppm}}
```

ppmnorm

This command has been superseded by **pnmnorm**.

More information: https://netpbm.sourceforge.net/doc/ppmnorm.html.

- View documentation for the current command:

```
tldr pnmnorm
```

ppmntsc

Make the RGB colors in a PPM image compatible with NTSC or PAL color systems.

More information: https://netpbm.sourceforge.net/doc/ppmntsc.html.

- Make the RGB colors in a PPM image compatible with NTSC color systems:

```
ppmntsc {{path/to/input_file.ppm}} > {{path/to/
output_file.ppm}}
```

- Make the RGB colors in a PPM image compatible with PAL color systems:

```
ppmntsc --pal {{path/to/input_file.ppm}} > {{path/to/
output_file.ppm}}
```

- Print the number of illegal pixels in the input image to `stderr`:

```
ppmntsc {{[--verb|--verbose]}} {{path/to/input_file.ppm}} >
{{path/to/output_file.ppm}}
```

- Output only legal/illegal/corrected pixels, set other pixels to black:

```
ppmntsc --{{legalonly|illegalonly|correctedonly}} {{path/to/
input_file.ppm}} > {{path/to/output_file.ppm}}
```

ppmpat

Produce a PPM image with a pattern.

More information: https://netpbm.sourceforge.net/doc/ppmpat.html.

- Produce a PPM file of the specified pattern with the specified dimensions:

```
ppmpat -{{gingham2|gingham3|madras|tartan|poles|...}}
{{width}} {{height}} > {{path/to/file.ppm}}
```

- Produce a PPM file of a camo pattern using the specified colors:

```
ppmpat {{[-ca|-camo]}} {{[-co|-color]}} {{color1,color2,...}}
{{width}} {{height}} > {{path/to/file.ppm}}
```

ppmquant

This command has been replaced by **`pnmquant`** and **`pnmremap`**.

More information: https://netpbm.sourceforge.net/doc/ppmquant.html.

- View documentation for `pnmquant`:

```
tldr pnmquant
```

- View documentation for `pnmremap`:

```
tldr pnmremap
```

ppmquantall

This command has been superseded by **`pnmquantall`**.

More information: https://netpbm.sourceforge.net/doc/ppmquantall.html.

- View documentation for the current command:

```
tldr pnmquantall
```

ppmrainbow

Generate a rainbow.

More information: https://netpbm.sourceforge.net/doc/ppmrainbow.html.

- Generate a rainbow consisting of the specified colors:

```
ppmrainbow {{color1 color2 ...}} > {{path/to/
output_file.ppm}}
```

- Specify the size of the output in pixels:

```
ppmrainbow {{[-w|-width]}} {{width}} {{[-h|-height]}}
{{height}} {{color1 color2 ...}} > {{path/to/
output_file.ppm}}
```

- End the rainbow with the last color specified, do not repeat the first color:

```
ppmrainbow {{[-n|-norepeat]}} {{color1 color2 ...}} > {{path/
to/output_file.ppm}}
```

ppmrelief

Produce a relief of a PPM image.

More information: https://netpbm.sourceforge.net/doc/ppmrelief.html.

- Produce a relief of the specified PPM image:

```
ppmrelief {{path/to/input_file.ppm}} > {{path/to/
output_file.ppm}}
```

ppmshadow

Add simulated shadows to a PPM image.

More information: https://netpbm.sourceforge.net/doc/ppmshadow.html.

- Add simulated shadows to a PPM image:

```
ppmshadow {{path/to/input_file.ppm}} > {{path/to/
output_file.ppm}}
```

- [b]lur the image by the specified number of pixels:

```
ppmshadow -b {{n}} {{path/to/input_file.ppm}} > {{path/to/
output_file.ppm}}
```

- Specify the displacement of the simulated light source to the left and the top of the image:

```
ppmshadow -x {{left_offset}} -y {{top_offset}} {{path/to/
input_file.ppm}} > {{path/to/output_file.ppm}}
```

ppmshift

Shift the lines in a PPM image by a randomized amount.

More information: https://netpbm.sourceforge.net/doc/ppmshift.html.

- Shift the lines in the input image by a randomized amount not exceeding s to the left or to the right:

```
ppmshift {{s}} {{path/to/input_file.ppm}} > {{path/to/
output_file.ppm}}
```

ppmspread

Displace the pixels in a PPM image by a randomized amount.

More information: https://netpbm.sourceforge.net/doc/ppmspread.html.

- Displace the pixels in a PPM image by a randomized amount that is at most a:

```
ppmspread {{a}} {{path/to/input_file.ppm}} > {{path/to/
output_file.ppm}}
```

- Specify a seed to a the pseudo-random number generator:

```
ppmspread {{a}} {{path/to/input_file.ppm}} {{[-r|-
randomseed]}} {{seed}} > {{path/to/output_file.ppm}}
```

ppmtoacad

Convert a PPM image to an AutoCAD database or slide.

More information: https://netpbm.sourceforge.net/doc/ppmtoacad.html.

- Convert a PPM image to an AutoCAD slide:

```
ppmtoacad {{path/to/file.ppm}} > {{path/to/file.acad}}
```

- Convert a PPM image to an AutoCAD binary database import file:

```
ppmtoacad {{[-d|-dxb]}} {{path/to/file.ppm}} > {{path/to/file.dxb}}
```

- Restrict the colors in the output to 8 RGB shades:

```
ppmtoacad -8 {{path/to/file.ppm}} > {{path/to/file.dxb}}
```

ppmtoapplevol

Convert a PPM image into an Apple volume label image.

More information: https://netpbm.sourceforge.net/doc/ppmtoapplevol.html.

- Convert a PPM image into an Apple volume label image:

```
ppmtoapplevol {{path/to/image.ppm}} > {{path/to/output}}
```

ppmtoarbtxt

Convert a PPM image to an arbitrary text format according to a template.

More information: https://netpbm.sourceforge.net/doc/ppmtoarbtxt.html.

- Convert a PPM image to text as specified by the given template:

```
ppmtoarbtxt {{path/to/template}} {{path/to/image.ppm}} >
{{path/to/output_file.txt}}
```

- Convert a PPM image to text as specified by the given template, prepend the contents of the specified head template:

```
ppmtoarbtxt {{path/to/template}} -hd {{path/to/
head_template}} {{path/to/image.ppm}} > {{path/to/
output_file.txt}}
```

- Convert a PPM image to text as specified by the given template, append the contents of the specified tail template:

```
ppmtoarbtxt {{path/to/template}} -hd {{path/to/
tail_template}} {{path/to/image.ppm}} > {{path/to/
output_file.txt}}
```

- Display version:

```
ppmtoarbtxt {{[-v|-version]}}
```

ppmtoascii

Convert a PPM image to an ASCII image using ANSI terminal color codes.

See also: **ppmtoterm**, **pbmtoascii**, **pbmto4425**.

More information: https://netpbm.sourceforge.net/doc/ppmtoascii.html.

- Convert a PPM image to an ASCII image, combining an area of 1x2 pixels into a character:

```
ppmtoascii {{path/to/input.ppm}} > {{path/to/output.txt}}
```

- Convert a PPM image to an ASCII image, combining an area of 2x4 pixels into a character:

```
ppmtoascii -2x4 {{path/to/input.ppm}} > {{path/to/
output.txt}}
```

ppmtobmp

Convert a PPM image to a BMP file.

More information: https://netpbm.sourceforge.net/doc/ppmtobmp.html.

- Convert a PPM image to a BMP file:

```
ppmtobmp {{path/to/file.ppm}} > {{path/to/file.bmp}}
```

- Explicitly specify whether or not a Windows BMP file or an OS/2 BMP file should be created:

```
ppmtobmp -{{windows|os2}} {{path/to/file.ppm}} > {{path/to/file.bmp}}
```

- Use a specific number of bits for each pixel:

```
ppmtobmp {{[-b|-bbp]}} {{1|4|8|24}} {{path/to/file.ppm}} > {{path/to/file.bmp}}
```

ppmtoeyuv

Convert a PPM image to an Berkeley YUV file.

More information: https://netpbm.sourceforge.net/doc/ppmtoeyuv.html.

- Read a PPM image from the input file, convert it to a Berkeley YUV image and store it in the specified output file:

```
ppmtoeyuv {{path/to/input_file.ppm}} > {{path/to/
output_file.eyuv}}
```

ppmtogif

This command has been superseded by **`pamtogif`**.

More information: https://netpbm.sourceforge.net/doc/ppmtogif.html.

- View documentation for the current command:

```
tldr pamtogif
```

ppmtoicr

Convert a PPM image to NCSA ICR format.

More information: https://netpbm.sourceforge.net/doc/ppmtoicr.html.

- Convert a PPM image to a ICR file:

```
ppmtoicr {{path/to/file.ppm}} > {{path/to/file.icr}}
```

- Display the output in name:

```
ppmtoicr {{[-w|-windowname]}} {{name}} {{path/to/file.ppm}} >
{{path/to/file.icr}}
```

- Expand the image by the specified factor:

```
ppmtoicr {{[-e|-expand]}} {{factor}} {{path/to/file.ppm}} >
{{path/to/file.icr}}
```

- Display the output on the screen with the specified number:

```
ppmtoicr {{[-d|-display]}} {{number}} {{path/to/file.ppm}} >
{{path/to/file.icr}}
```

ppmtoilbm

Convert a PPM image to an ILBM file.

More information: https://netpbm.sourceforge.net/doc/ppmtoilbm.html.

- Convert a PPM image to an ILBM file:

```
ppmtoilbm {{path/to/file.ppm}} > {{path/to/file.ilbm}}
```

- Write a maximum of n planes to the ILBM file and produce a HAM/24bit/direct color file if this number is exceeded:

```
ppmtoilbm {{[-mp|-maxplanes]}} {{n}} -{{hamif|24if|dcif}}
{{path/to/file.ppm}} > {{path/to/file.ilbm}}
```

- Produce a ILBM file with exactly n planes:

```
ppmtoilbm {{[-fp|-fixplanes]}} {{n}} {{path/to/file.ppm}} >
{{path/to/file.ilbm}}
```

- Select the compression method to be used:

```
ppmtoilbm -{{compress|nocompress|savemem}} {{path/to/
file.ppm}} > {{path/to/file.ilbm}}
```

ppmtojpeg

This command has been superseded by **pnmtojpeg**.

More information: https://netpbm.sourceforge.net/doc/ppmtojpeg.html.

- View documentation for the current command:

```
tldr pnmtojpeg
```

ppmtoleaf

Convert a PPM image to the Interleaf image format.

More information: https://netpbm.sourceforge.net/doc/ppmtoleaf.html.

- Convert a PPM image to a LEAF file:

```
ppmtoleaf {{path/to/file.ppm}} > {{path/to/file.leaf}}
```

ppmtolj

Convert a PPM file to an HP LaserJet PCL 5 Color file.

More information: https://netpbm.sourceforge.net/doc/ppmtolj.html.

- Convert a PPM file to an HP LaserJet PCL 5 Color file:

```
ppmtolj {{path/to/input.ppm}} > {{path/to/output.lj}}
```

- Apply a gamma correction using the specified gamma value:

```
ppmtolj {{[-g|-gamma]}} {{gamma}} {{path/to/input.ppm}} >
{{path/to/output.lj}}
```

- Specify the required resolution:

```
ppmtolj {{[-r|-resolution]}} {{75|100|150|300|600}} {{path/
to/input.ppm}} > {{path/to/output.lj}}
```

ppmtomap

This command has been superseded by **`pnmcolormap`**.

More information: https://netpbm.sourceforge.net/doc/ppmtomap.html.

- View documentation for the current command:

```
tldr pnmcolormap
```

ppmtomitsu

Convert a PPM image to a Mitsubishi S340-10 file.

More information: https://netpbm.sourceforge.net/doc/ppmtomitsu.html.

- Convert a PPM image to a MITSU file:

```
ppmtomitsu {{path/to/file.ppm}} > {{path/to/file.mitsu}}
```

- Enlarge the image by the specified factor, use the specified sharpness and produce n copies:

```
ppmtomitsu {{[-e|-enlarge]}} {{1|2|3}} {{[-s|-sharpness]}}
{{1|2|3|4}} {{[-c|-copy]}} {{n}} {{path/to/file.ppm}} >
{{path/to/file.mitsu}}
```

- Use the given medium for the printing process:

```
ppmtomitsu {{[-m|-media]}} {{A|A4|AS|A4S}} {{path/to/
file.ppm}} > {{path/to/file.mitsu}}
```

ppmtompeg

Encode an MPEG-1 stream.

More information: https://netpbm.sourceforge.net/doc/ppmtompeg.html.

- Produce an MPEG-1 stream using the parameter file to specify inputs and outputs:

```
ppmtompeg {{path/to/parameter_file}}
```

- Encode the GOP with the specified number only:

```
ppmtompeg {{[-g|-gop]}} {{gop_num}} {{path/to/
parameter_file}}
```

- Specify the first and last frame to encode:

```
ppmtompeg {{[-fr|-frames]}} {{first_frame}} {{last_frame}}
{{path/to/parameter_file}}
```

- Combine multiple MPEG frames into a single MPEG-1 stream:

```
ppmtompeg -combine_frames {{path/to/parameter_file}}
```

ppmtoneo

Convert a PPM image to an Atari Neochrome file.

More information: https://netpbm.sourceforge.net/doc/ppmtoneo.html.

- Convert a PPM image to a NEO file:

```
ppmtoneo {{path/to/file.ppm}} > {{path/to/file.neo}}
```

ppmtopcx

Convert a PPM image to a PCX file.

More information: https://netpbm.sourceforge.net/doc/ppmtopcx.html.

- Convert a PPM image to a PCX file:

```
ppmtopcx {{path/to/file.ppm}} > {{path/to/file.pcx}}
```

- Produce a PCX file with the specified color depth:

```
ppmtopcx -{{8bit|24bit}} {{path/to/file.ppm}} > {{path/to/file.pcx}}
```

ppmtopgm

Convert a PPM image to a PGM image.

More information: https://netpbm.sourceforge.net/doc/ppmtopgm.html.

- Convert PPM image to PGM image:

```
ppmtopgm {{path/to/file.ppm}} > {{path/to/file.pgm}}
```

- Display version:

```
ppmtopgm {{[-v|-version]}}
```

ppmtopi1

Convert a PPM image to an Atari Degas PI1 image.

See also: **`pi1toppm`**.

More information: https://netpbm.sourceforge.net/doc/ppmtopi1.html.

- Convert a PPM image into an Atari Degas PI1 image:

```
ppmtopi1 {{path/to/image.ppm}} > {{path/to/output_image.pi1}}
```

ppmtopict

Convert a PPM image to a Macintosh PICT file.

More information: https://netpbm.sourceforge.net/doc/ppmtopict.html.

- Convert a PPM image to a PICT file:

```
ppmtopict {{path/to/file.ppm}} > {{path/to/file.pict}}
```

ppmtopj

Convert a PPM file to an HP PaintJet file.

More information: https://netpbm.sourceforge.net/doc/ppmtopj.html.

- Convert a PPM file to an HP PaintJet file:

```
ppmtopj {{path/to/input.ppm}} > {{path/to/output.pj}}
```

- Move the image in the x and y direction:

```
ppmtopj {{[-x|-xpos]}} {{dx}} {{[-y|-ypos]}} {{dy}} {{path/
to/input.ppm}} > {{path/to/output.pj}}
```

- Explicitly specify a gamma value:

```
ppmtopj {{[-g|-gamma]}} {{gamma}} {{path/to/input.ppm}} >
{{path/to/output.pj}}
```

ppmtopjxl

Convert a PPM image into an HP PaintJet XL PCL file.

More information: https://netpbm.sourceforge.net/doc/ppmtopjxl.html.

- Convert a PPM image into an PJXL file:

```
ppmtopjxl {{path/to/image.ppm}} > {{path/to/output.pjxl}}
```

- Resize the input image:

```
ppmtopjxl {{[-xsi|-xsize]}} {{10cm}} {{[-ysi|-ysize]}}
{{5cm}} {{path/to/image.ppm}} > {{path/to/output.pjxl}}
```

- Shift the input image:

```
ppmtopjxl {{[-xsh|-xshift]}} {{10pt}} {{[-ysh|-yshift]}}
{{5pt}} {{path/to/image.ppm}} > {{path/to/output.pjxl}}
```

- Do not use the normal TIFF 4.0 compression method:

```
ppmtopjxl {{[-n|-nopack]}} {{path/to/image.ppm}} > {{path/to/
output.pjxl}}
```

ppmtoppm

Copy a PPM image.

More information: https://netpbm.sourceforge.net/doc/ppmtoppm.html.

- Copy a PPM image (i.e. a PBM, PGM or PPM image) from `stdin` to `stdout`:

```
ppmtoppm < {{path/to/image.ppm}} > {{path/to/output.ppm}}
```

- Display version:

```
ppmtoppm {{[-v|-version]}}
```

ppmtopuzz

Convert a PPM image to an X11 puzzle file.

More information: https://netpbm.sourceforge.net/doc/ppmtopuzz.html.

- Convert a PPM image to an X11 puzzle file:

```
ppmtopuzz {{path/to/file.ppm}} > {{path/to/file.puzz}}
```

ppmtorgb3

Separate the color components of a PPM file into three separate PGM files.

See also: **`rgb3toppm`**.

More information: https://netpbm.sourceforge.net/doc/ppmtorgb3.html.

- Separate the color components of a PPM file, saving the outputs to `file.red`, `file.grn` and `file.blu`:

```
ppmtorgb3 {{path/to/file.ppm}}
```

ppmtosixel

Convert a PPM image to DEC sixel format.

More information: https://netpbm.sourceforge.net/doc/ppmtosixel.html.

- Convert a PPM image to DEC sixel format:

```
ppmtosixel {{path/to/file.ppm}} > {{path/to/file.sixel}}
```

- Produce an uncompressed SIXEL file that is much slower to print:

```
ppmtosixel {{[-r|-raw]}} {{path/to/file.ppm}} > {{path/to/
file.sixel}}
```

- Add a left margin of 1.5 inches:

```
ppmtosixel {{[-m|-margin]}} {{path/to/file.ppm}} > {{path/to/
file.sixel}}
```

- Encode control codes in a more portable (although less space-efficient) way:

```
ppmtosixel -7bit {{path/to/file.ppm}} > {{path/to/
file.sixel}}
```

ppmtospu

Convert a PPM file to an Atari Spectrum 512 image.

More information: https://netpbm.sourceforge.net/doc/ppmtospu.html.

- Convert a PPM file to an Atari Spectrum 512 image:

```
ppmtospu {{path/to/input.ppm}} > {{path/to/output.spu}}
```

- Use a dithering matrix of the specified size (0 means no dithering):

```
ppmtospu -d{{0|2|4}} {{path/to/input.ppm}} > {{path/to/
output.spu}}
```

ppmtoterm

Convert a PPM image to an ANSI ISO 6429 ASCII image.

See also: **`ppmtoascii`**, **`pbmtoascii`**, **`pbmto4425`**.

More information: https://netpbm.sourceforge.net/doc/ppmtoterm.html.

- Convert a PPM image to an ANSI ISO 6429 ASCII image, mapping each pixel to an individual character:

```
ppmtoterm {{path/to/input.ppm}} > {{path/to/output.txt}}
```

ppmtotga

This command has been superseded by **pamtotga**.

More information: https://netpbm.sourceforge.net/doc/ppmtotga.html.

- View documentation for the current command:

```
tldr pamtotga
```

ppmtouil

This command has been superseded by **pamtouil**.

More information: https://netpbm.sourceforge.net/doc/ppmtouil.html.

- View documentation for the current command:

```
tldr pamtouil
```

ppmtowinicon

This command has been superseded by **pamtowinicon**.

More information: https://netpbm.sourceforge.net/doc/ppmtowinicon.html.

- View documentation for the current command:

```
tldr pamtowinicon
```

ppmtoxpm

Convert a PPM image to an X11 version 3 pixmap.

More information: https://netpbm.sourceforge.net/doc/ppmtoxpm.html.

- Convert a PPM image to a XPM image:

```
ppmtoxpm {{path/to/input_file.ppm}} > {{path/to/
output_file.xpm}}
```

- Specify the prefix string in the output XPM image:

```
ppmtoxpm {{[-n|-name]}} {{prefix_string}} {{path/to/
input_file.ppm}} > {{path/to/output_file.xpm}}
```

- In the output XPM file, specify colors by their hexadecimal code instead of their name:

```
ppmtoxpm {{[-h|-hexonly]}} {{path/to/input_file.ppm}} >
{{path/to/output_file.xpm}}
```

- Use the specified PGM file as a transparency mask:

```
ppmtoxpm {{[-a|-alphamask]}} {{path/to/alpha_file.pgm}}
{{path/to/input_file.ppm}} > {{path/to/output_file.xpm}}
```

ppmtoyuv

Convert a PPM image to an Abekas YUV file.

More information: https://netpbm.sourceforge.net/doc/ppmtoyuv.html.

- Read a PPM image from the input file, convert it to an Abekas YUV image and store it in the specified output file:

```
ppmtoyuv {{path/to/input_file.ppm}} > {{path/to/
output_file.yuv}}
```

ppmtoyuvsplit

Convert a PPM image to three subsampled Abekas YUV files.

More information: https://netpbm.sourceforge.net/doc/ppmtoyuvsplit.html.

- Read a PPM image from the input file, convert it to three subsampled Abekas YUV image and store these images to output files starting with the specified basename:

```
ppmtoyuvsplit {{basename}} {{path/to/input_file.ppm}}
```

ppmtv

Make a PPM Image look like taken from an American TV.

Dim every other row of image data down by the specified dim factor (a number between 0 and 1).

More information: https://netpbm.sourceforge.net/doc/ppmtv.html.

- Give the PPM image an American TV appearance:

```
ppmtv {{dim_factor}} {{path/to/file.ppm}} > {{path/to/output.ppm}}
```

- Suppress all informational messages:

```
ppmtv {{[-q|-quiet]}}
```

- Display version:

```
ppmtv {{[-v|-version]}}
```

ppmwheel

Generate a PPM image of a color wheel.

More information: https://netpbm.sourceforge.net/doc/ppmwheel.html.

- Generate a color wheel of type `Ppmcirc`:

```
ppmwheel {{diameter}} > {{path/to/output.ppm}}
```

- Generate a color wheel of type `Hue-value`:

```
ppmwheel {{[-huev|-huevalue]}} {{diameter}} > {{path/to/
output.ppm}}
```

- Generate a color wheel of type `Hue-saturation`:

```
ppmwheel {{[-hues|-huesaturation]}} {{diameter}} > {{path/to/
output.ppm}}
```

pppd

Establish Point-to-Point connection to another computer.

It should not be invoked manually.

More information: https://ppp.samba.org/pppd.html.

- Start the daemon:

```
pppd
```

pprof

Visualize and analyze profiling data.

More information: https://github.com/google/pprof.

- Generate a text report from a specific profiling file, on fibbo binary:

```
pprof -top {{./fibbo}} {{./fibbo-profile.pb.gz}}
```

- Generate a graph and open it on a web browser:

```
pprof -svg {{./fibbo}} {{./fibbo-profile.pb.gz}}
```

- Run pprof in interactive mode to be able to manually launch `pprof` on a file:

```
pprof {{./fibbo}} {{./fibbo-profile.pb.gz}}
```

- Run a web server that serves a web interface on top of `pprof`:

```
pprof -http={{localhost:8080}} {{./fibbo}} {{./fibbo-profile.pb.gz}}
```

- Fetch a profile from an HTTP server and generate a report:

```
pprof {{http://localhost:8080/debug/pprof}}
```

pr

Paginate or columnate files for printing.

More information: https://www.gnu.org/software/coreutils/manual/html_node/pr-invocation.html.

- Print multiple files with a default header and footer:

```
pr {{path/to/file1 path/to/file2 ...}}
```

- Print with a custom centered header:

```
pr {{[-h|--header]}} "{{header}}" {{path/to/file1 path/to/file2 ...}}
```

- Print with numbered lines and a custom date format:

```
pr {{[-n|--number-lines]}} {{[-D|--date-format]}} "{{format}}" {{path/to/file1 path/to/file2 ...}}
```

- Print all files together, one in each column, without a header or footer:

```
pr {{[-m|--merge]}} {{[-T|--omit-pagination]}} {{path/to/file1 path/to/file2 ...}}
```

- Print, beginning at page 2 up to page 5, with a given page length (including header and footer):

```
pr +2:5 {{[-l|--length]}} {{page_length}} {{path/to/file1 path/to/file2 ...}}
```

- Print with an offset for each line and a truncating custom page width:

```
pr {{[-o|--indent]}} {{offset}} {{[-W|--page_width]}} {{width}} {{path/to/file1 path/to/file2 ...}}
```

pre-commit

Create Git hooks that get run before a commit.

More information: https://pre-commit.com/#command-line-interface.

- Install pre-commit into your Git hooks:

```
pre-commit install
```

- Run pre-commit hooks on all staged files:

```
pre-commit run
```

- Run pre-commit hooks on all files, staged or unstaged:

```
pre-commit run --all-files
```

- Clean pre-commit cache:

```
pre-commit clean
```

- Update pre-commit configuration file to the latest repos' versions:

```
pre-commit autoupdate
```

prename

This command is an alias of **rename**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common rename
```

prettier

An opinionated code formatter for JavaScript, JSON, CSS, YAML, and more.

More information: https://prettier.io/docs/cli.

- Format a file and print the result to `stdout`:

  ```
  prettier {{path/to/file}}
  ```

- Check if a specific file has been formatted:

  ```
  prettier --check {{path/to/file}}
  ```

- Run with a specific configuration file:

  ```
  prettier --config {{path/to/config_file}} {{path/to/file}}
  ```

- Format a file or directory, replacing the original:

  ```
  prettier --write {{path/to/file_or_directory}}
  ```

- Format files or directories recursively using single quotes and no trailing commas:

  ```
  prettier --single-quote --trailing-comma {{none}} --write {{path/to/file_or_directory}}
  ```

- Format JavaScript and TypeScript files recursively, replacing the original:

  ```
  prettier --write "**/*.{js,jsx,ts,tsx}"
  ```

pretty-bytes

Convert bytes to a human-readable string.

More information: https://github.com/sindresorhus/pretty-bytes-cli.

- Convert numeric bytes value to a human-readable string:

```
pretty-bytes {{1337}}
```

- Convert numeric bytes value from `stdin` to a human-readable string:

```
echo {{1337}} | pretty-bytes
```

- Display help:

```
pretty-bytes --help
```

print

print can refer to multiple commands with the same name.

- View documentation for the Zsh builtin:

  ```
  tldr print.zsh
  ```

- View documentation for the `print` alias of `run-mailcap`:

  ```
  tldr {{[-p|--platform]}} linux print.runmailcap
  ```

- View documentation for printing text files on Windows:

  ```
  tldr {{[-p|--platform]}} windows print.win
  ```

print

Z Shell (**zsh**) builtin. Prints arguments, similar to **echo**.

See also: **echo**, **printf**, **zsh**.

More information: https://zsh.sourceforge.io/Doc/Release/Shell-Builtin-Commands.html.

- Print input:

```
print "Hello" "World"
```

- Print separated by newline(s):

```
print -l "Line1" "Line 2" "Line3"
```

- Print without trailing newline:

```
print -n "Hello"; print "World"
```

- Enable backslash escapes:

```
print -e "Line 1\nLine2"
```

- Print arguments as described by `printf` (for greater portability across shells, consider using the `printf` command instead):

```
print -f "%s is %d years old.\n" "Alice" 30
```

printenv

Print values of all or environment variables.

More information: https://www.gnu.org/software/coreutils/manual/html_node/printenv-invocation.html.

- Display key-value pairs of all environment variables:

```
printenv
```

- Display the value of a specific variable:

```
printenv {{HOME}}
```

- Display the value of a variable and end with NUL instead of newline:

```
printenv {{[-0|--null]}} {{HOME}}
```

printf

Format and print text.

See also: **echo**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/printf-invocation.html.

- Print a text message:

```
printf "{{%s\n}}" "{{Hello world}}"
```

- Print an integer in bold blue:

```
printf "{{\e[1;34m%.3d\e[0m\n}}" {{42}}
```

- Print a float number with the Unicode Euro sign:

```
printf "{{\u20AC %.2f\n}}" {{123.4}}
```

- Print a text message composed with environment variables:

```
printf "{{var1: %s\tvar2: %s\n}}" "{{$VAR1}}" "{{$VAR2}}"
```

- Store a formatted message in a variable (does not work on Zsh):

```
printf -v {{myvar}} {{"This is %s = %d\n" "a year" 2016}}
```

- Print a hexadecimal, octal and scientific number:

```
printf "{{hex=%x octal=%o scientific=%e}}" 0x{{FF}} 0{{377}} {{100000}}
```

procs

Display information about the active processes.

More information: https://github.com/dalance/procs.

- List all processes showing the PID, user, CPU usage, memory usage, and the command which started them:

```
procs
```

- List all processes as a tree:

```
procs --tree
```

- List information about processes, if the commands which started them contain Zsh:

```
procs {{zsh}}
```

- List information about all processes sorted by CPU time in [a]scending or [d]escending order:

```
procs {{--sorta|--sortd}} cpu
```

- List information about processes with either a PID, command, or user containing `41` or `firefox`:

```
procs --or {{PID|command|user}} {{41}} {{firefox}}
```

- List information about processes with both PID `41` and a command or user containing `zsh`:

```
procs --and {{41}} {{zsh}}
```

progpilot

A PHP static analysis tool for detecting security vulnerabilities.

More information: https://github.com/designsecurity/progpilot.

- Analyze the current directory:

```
progpilot
```

- Analyze a specific file or directory:

```
progpilot {{path/to/file_or_directory}}
```

- Specify a custom configuration file:

```
progpilot --configuration {{path/to/configuration.yml}}
```

progress

Display/Monitor the progress of running coreutils.

More information: https://github.com/Xfennec/progress.

- Show the progress of running coreutils:

```
progress
```

- Monitor all running coreutils:

```
progress {{[-m|--monitor]}}
```

- Show the progress of running coreutils in quiet mode:

```
progress {{[-q|--quiet]}}
```

- Launch and monitor a single long-running command:

```
{{command}} & progress {{[-m|--monitor]}} {{[-p|--pid]}} $!
```

- Include an estimate of time remaining for completion:

```
progress {{[-w|--wait]}} {{[-c|--command]}} {{firefox}}
```

Projucer

A project manager for JUCE framework applications.

More information: https://docs.juce.com/master/projucer_manual.html#projucer_manual_tools_command_line_tools.

- Display information about a project:

```
Projucer --status {{path/to/project_file}}
```

- Resave all files and resources in a project:

```
Projucer --resave {{path/to/project_file}}
```

- Update the version number in a project:

```
Projucer --set-version {{version_number}} {{path/to/project_file}}
```

- Generate a JUCE project from a PIP file:

```
Projucer --create-project-from-pip {{path/to/PIP}} {{path/to/output}}
```

- Remove all JUCE-style comments (`//=====`, `//-----` or `///////`):

```
Projucer --tidy-divider-comments {{path/to/target_folder}}
```

- Display help:

```
Projucer --help
```

promtool

Tooling for the Prometheus monitoring system.

More information: https://prometheus.io/docs/prometheus/latest/getting_started/.

- Check if the configuration files are valid or not (if present report errors):

  ```
  promtool check config {{config_file.yml}}
  ```

- Check if the rule files are valid or not (if present report errors):

  ```
  promtool check rules {{rules_file.yml}}
  ```

- Pass Prometheus metrics over `stdin` to check them for consistency and correctness:

  ```
  curl --silent {{http://example.com:9090/metrics/}} | promtool check metrics
  ```

- Unit tests for rules config:

  ```
  promtool test rules {{test_file.yml}}
  ```

propelauth

Set up PropelAuth authentication as quickly and easily as possible.

More information: https://docs.propelauth.com/reference/api/cli.

- Login to PropelAuth using an API Key generated from https://auth.propelauth.com/api_keys/personal:

  ```
  propelauth login
  ```

- Set the default PropelAuth Project for the CLI. If no default project is set, the system will prompt for selecting a Project each time certain commands are run:

  ```
  propelauth set-default-project
  ```

- Install PropelAuth authentication in an application. If no directory is provided, the current directory is used:

  ```
  propelauth setup {{[-f|--framework]}} {{path/to/directory}}
  ```

- Log the CLI out of PropelAuth:

  ```
  propelauth logout
  ```

prosodyctl

The control tool for the Prosody XMPP server.

Note: Process management through **`prosodyctl`** is discouraged. Instead, use the tools provided by your system (e.g. **`systemctl`**).

More information: https://prosody.im/doc/prosodyctl.

- Show the status of the Prosody server:

  ```
  sudo prosodyctl status
  ```

- Reload the server's configuration files:

  ```
  sudo prosodyctl reload
  ```

- Add a user to the Prosody XMPP server:

  ```
  sudo prosodyctl adduser {{user@example.com}}
  ```

- Set a user's password:

  ```
  sudo prosodyctl passwd {{user@example.com}}
  ```

- Permanently delete a user:

  ```
  sudo prosodyctl deluser {{user@example.com}}
  ```

protector

Protect or unprotect branches on GitHub repositories.

More information: https://github.com/jcgay/protector.

- Protect branches of a GitHub repository (create branch protection rules):

```
protector {{branches_regex}} -repos {{organization/repository}}
```

- Use the dry run to see what would be protected (can also be used for freeing):

```
protector -dry-run {{branches_regex}} -repos {{organization/repository}}
```

- Free branches of a GitHub repository (delete branch protection rules):

```
protector -free {{branches_regex}} -repos {{organization/repository}}
```

protoc

Parse Google Protobuf **`.proto`** files and generate output in the specified language.

More information: https://developers.google.com/protocol-buffers.

- Generate Python code from a `.proto` file:

```
protoc --python_out={{path/to/output_directory}}
{{input_file.proto}}
```

- Generate Java code from a `.proto` file that imports other `.proto` files:

```
protoc --java_out={{path/to/output_directory}} --
proto_path={{path/to/import_search_path}}
{{input_file.proto}}
```

- Generate code for multiple languages:

```
protoc --csharp_out={{path/to/c#_output_directory}} --
js_out={{path/to/js_output_directory}} {{input_file.proto}}
```

- Encode a text-format message into a protocol message from a `.proto` file:

```
protoc < {{message.txt}} --encode={{TypeName}}
{{input_file.proto}}
```

- Decode a protocol message into text-format from a `.proto` file:

```
protoc < {{message.bin}} --decode={{TypeName}}
{{input_file.proto}}
```

- Decode a protocol message into raw tag/value pairs:

```
protoc < {{message.bin}} --decode_raw
```

prowler aws

Assess AWS security best practices, perform audits, compliance checks, and generate reports.

See also: **`prowler`**, **`prowler-azure`**, **`prowler-gcp`**, **`prowler-kubernetes`**, **`prowler-m365`**, **`prowler-github`**.

More information: https://docs.prowler.com/projects/prowler-open-source/en/latest/.

- Run the default set of checks on the AWS account:

```
prowler aws
```

- Use a custom AWS profile and filter audited regions:

```
prowler aws {{[-p|--profile]}} {{custom-profile}} {{[-f|--filter-region]}} {{us-east-1 eu-south-2 ...}}
```

- Run checks for selected AWS services:

```
prowler aws {{[-s|--services]}} {{s3 ec2 ...}}
```

- Run a specific AWS check:

```
prowler aws {{[-c|--checks]}} {{s3_bucket_public_access}}
```

- Exclude specific checks or services:

```
prowler aws {{[-e|--excluded-checks]}} {{s3_bucket_public_access}} --exclude-services {{s3 ec2 ...}}
```

prowler azure

Assess Azure security best practices, perform audits, compliance checks, and generate reports.

See also: **`prowler`**, **`prowler-aws`**, **`prowler-gcp`**, **`prowler-kubernetes`**, **`prowler-m365`**, **`prowler-github`**.

More information: https://docs.prowler.com/projects/prowler-open-source/en/latest/.

- Run the default set of checks on the current Azure account using Azure CLI authentication:

```
prowler azure --az-cli-auth
```

- Run checks for specific Azure subscriptions:

```
prowler azure --az-cli-auth --subscription-ids
{{subscription_id1 subscription_id2 ...}}
```

- Authenticate using a service principal via environment variables:

```
prowler azure --sp-env-auth
```

- Authenticate using browser login and specify a tenant ID:

```
prowler azure --browser-auth --tenant-id "{{XXXXXXXX}}"
```

- Authenticate using a managed identity (e.g. for Azure VM):

```
prowler azure --managed-identity-auth
```

- Run checks for selected Azure services:

```
prowler azure {{[-s|--services]}} {{defender iam ...}}
```

- Run a specific Azure check:

```
prowler azure {{[-c|--checks]}}
{{storage_blob_public_access_level_is_disabled}}
```

- Exclude specific checks or services:

```
prowler azure {{[-e|--excluded-checks]}}
{{storage_blob_public_access_level_is_disabled}} --exclude-
services {{defender iam ...}}
```

prowler gcp

Assess Google Cloud Platform (GCP) security best practices, audits, and compliance checks.

See also: **`prowler`**, **`prowler-aws`**, **`prowler-azure`**, **`prowler-kubernetes`**, **`prowler-m365`**, **`prowler-github`**.

More information: https://docs.prowler.com/projects/prowler-open-source/en/latest/.

- Run the default set of checks on all accessible GCP projects using default user credentials:

```
prowler gcp
```

- Authenticate using a service account credentials file:

```
prowler gcp --credentials-file {{path/to/credentials.json}}
```

- Scan specific GCP projects by ID:

```
prowler gcp --project-ids {{project_id1 project_id2 ...}}
```

- Run checks for selected GCP services:

```
prowler gcp {{[-s|--services]}} {{iam compute ...}}
```

- Run a specific GCP check:

```
prowler gcp {{[-c|--checks]}}
{{gcp_storage_bucket_logging_enabled}}
```

- Exclude specific checks or services:

```
prowler gcp {{[-e|--excluded-checks]}}
{{gcp_storage_bucket_logging_enabled}} --exclude-services
{{iam compute ...}}
```

prowler github

Assess GitHub account, repository, and organization security best practices.

See also: **`prowler`**, **`prowler-aws`**, **`prowler-azure`**, **`prowler-gcp`**, **`prowler-kubernetes`**, **`prowler-m365`**.

More information: https://docs.prowler.com/projects/prowler-open-source/en/latest/.

- Run all default GitHub security checks:

```
prowler github
```

- Authenticate using a GitHub Personal Access Token:

```
prowler github --personal-access-token {{pat}}
```

- Authenticate using a GitHub OAuth App Token:

```
prowler github --oauth-app-token {{oauth_token}}
```

- Authenticate using a GitHub App ID and private key:

```
prowler github --github-app-id {{app_id}} --github-app-key {{app_key}}
```

prowler kubernetes

Assess Kubernetes cluster security best practices and configurations.

See also: **`prowler`**, **`prowler-aws`**, **`prowler-azure`**, **`prowler-gcp`**, **`prowler-m365`**, **`prowler-github`**.

More information: https://docs.prowler.com/projects/prowler-open-source/en/latest/.

- Run the default checks using the default kubeconfig location:

```
prowler kubernetes
```

- Specify a custom kubeconfig file for scanning:

```
prowler kubernetes --kubeconfig-file {{path/to/kubeconfig}}
```

- Specify a specific Kubernetes context to scan:

```
prowler kubernetes --context {{my-context}}
```

- Scan specific namespaces only:

```
prowler kubernetes --namespaces {{default}} {{kube-system}}
```

- Run checks for selected Kubernetes services:

```
prowler kubernetes {{[-s|--services]}} {{ietcd
apiserver ...}}
```

- Run a specific Kubernetes check:

```
prowler kubernetes {{[-c|--checks]}} {{etcd_encryption}}
```

- Exclude specific checks or services:

```
prowler kubernetes {{[-e|--excluded-checks]}}
{{etcd_encryption}} --exclude-services {{ietcd
apiserver ...}}
```

prowler m365

Assess Microsoft 365 (M365) security configurations and best practices.

See also: **`prowler`**, **`prowler-aws`**, **`prowler-azure`**, **`prowler-gcp`**, **`prowler-kubernetes`**, **`prowler-github`**.

More information: https://docs.prowler.com/projects/prowler-open-source/en/latest/.

- Run Prowler with combined service principal and user credentials:

```
prowler m365 --env-auth
```

- Authenticate using a service principal:

```
prowler m365 --sp-env-auth
```

- Authenticate using the Azure CLI:

```
prowler m365 --az-cli-auth
```

- Authenticate using a browser and specify the tenant ID:

```
prowler m365 --browser-auth --tenant-id "{{XXXXXXXX}}"
```

- Run a specific Microsoft 365 check:

```
prowler m365 {{[-c|--checks]}}
{{etcd_enm365_onedrive_sharing_enabledcryption}}
```

- Exclude specific checks:

```
prowler m365 {{[-e|--excluded-checks]}}
{{m365_onedrive_sharing_enabled}}
```

prowler

Performs security best practices assessments, audits and compliance checks across AWS, Azure, Google Cloud, and Kubernetes.

See also: **`prowler-aws`**, **`prowler-azure`**, **`prowler-gcp`**, **`prowler-kubernetes`**, **`prowler-m365`**, **`prowler-github`**.

More information: https://docs.prowler.com/projects/prowler-open-source/en/latest/.

- Run an AWS, Azure, GCP, Kubernetes - as provider - audit with default checks:

```
prowler {{provider}}
```

- Show all available checks for a specific provider:

```
prowler {{provider}} {{[-l|--list-checks]}}
```

- Show all available services for a specific provider:

```
prowler {{provider}} --list-services
```

- Generate output in multiple formats, including JSON-ASFF for AWS Security Hub:

```
prowler {{provider}} --output-modes {{csv,json-asff,html,...}}
```

- Execute in verbose mode:

```
prowler {{provider}} --verbose
```

- Filter findings by status:

```
prowler {{provider}} --status {{PASS,FAIL,MANUAL}}
```

- Display help:

```
prowler --help
```

- Display version:

```
prowler {{[-v|--version]}}
```

proxify

A versatile and portable proxy for capturing, manipulating, and replaying HTTP/HTTPS traffic on the go.

See also: **`mitmproxy`**.

More information: https://github.com/projectdiscovery/proxify.

- Start a HTTP proxy (on the loopback network interface `127.0.0.1` and port `8888`):

```
proxify
```

- Start a HTTP proxy on a custom network interface and port (may require `sudo` for a port number lower than `1024`):

```
proxify {{[-ha|-http-addr]}} "{{ip_address}}:{{port_number}}"
```

- Specify output format and output file:

```
proxify {{[-of|-output-format]}} {{jsonl|yaml}} {{[-o|-output]}} {{path/to/file}}
```

- Display help:

```
proxify -h
```

prqlc

PRQL compiler.

PRQL is a modern language for transforming data - a simple, powerful, pipelined SQL replacement.

More information: https://prql-lang.org.

- Run the compiler interactively:

```
prqlc compile
```

- Compile a specific `.prql` file to `stdout`:

```
prqlc compile {{path/to/file.prql}}
```

- Compile a `.prql` file to a `.sql` file:

```
prqlc compile {{path/to/source.prql}} {{path/to/target.sql}}
```

- Compile a query:

```
echo "{{from employees | filter has_dog | select salary}}" |
prqlc compile
```

- Watch a directory and compile on file modification:

```
prqlc watch {{path/to/directory}}
```

ps-nvm

PowerShell-based utility to manage multiple Node.js versions, inspired by **`nvm`**.

This tool provides multiple commands that all can only be run through PowerShell.

More information: https://github.com/aaronpowell/ps-nvm.

- View documentation for `Get-NodeInstallLocation`, a tool to get the current Node.js install location:

  ```
  tldr get-nodeinstalllocation
  ```

- View documentation for `Get-NodeVersions`, a tool to list all available and currently-installed Node.js versions:

  ```
  tldr get-nodeversions
  ```

- View documentation for `Install-NodeVersion`, a tool to install Node.js runtime versions:

  ```
  tldr install-nodeversion
  ```

- View documentation for `Remove-NodeVersion`, a tool to uninstall an existing Node.js version:

  ```
  tldr remove-nodeversion
  ```

- View documentation for `Set-NodeInstallLocation`, a tool to set the Node.js install location:

  ```
  tldr set-nodeinstalllocation
  ```

- View documentation for `Set-NodeVersion`, a tool to set the default version of Node.js:

  ```
  tldr set-nodeversion
  ```

ps

Information about running processes.

More information: https://manned.org/ps.

- List all running processes:

```
ps aux
```

- List all running processes including the full command string:

```
ps auxww
```

- Search for a process that matches a string (the brackets will prevent `grep` from matching itself):

```
ps aux | grep {{[s]tring}}
```

- List all processes of the current user in extra full format:

```
ps {{[-u|--user]}} $(id {{[-u|--user]}}) -F
```

- List all processes of the current user as a tree:

```
ps {{[-u|--user]}} $(id {{[-u|--user]}}) f
```

- Get the parent PID of a process:

```
ps {{[-o|--format]}} ppid= {{[-p|--pid]}} {{pid}}
```

- Sort processes by memory consumption:

```
ps --sort size
```

psalm

A static analysis tool for finding errors in PHP applications.

More information: https://psalm.dev.

- Generate a Psalm configuration:

```
psalm --init
```

- Analyze the current working directory:

```
psalm
```

- Analyze a specific directory or file:

```
psalm {{path/to/file_or_directory}}
```

- Analyze a project with a specific configuration file:

```
psalm --config {{path/to/psalm.xml}}
```

- Include informational findings in the output:

```
psalm --show-info
```

- Analyze a project and display statistics:

```
psalm --stats
```

- Analyze a project in parallel with 4 threads:

```
psalm --threads {{4}}
```

psexec.py

Execute commands on a remote Windows machine using **`RemComSvc`**, providing PsExec-like functionality.

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- Spawn an interactive shell on a remote target:

```
psexec.py {{domain}}/{{username}}:{{password}}@{{target}}
```

- Execute a specific command on a remote target:

```
psexec.py {{domain}}/{{username}}:{{password}}@{{target}}
{{command}}
```

- Copy the filename for later execution, arguments are passed in the command:

```
psexec.py -c {{filename}} {{domain}}/{{username}}:
{{password}}@{{target}} {{command}}
```

- Execute a command from a specific path on a remote target:

```
psexec.py -path {{path}} {{domain}}/{{username}}:{{password}}
@{{target}} {{command}}
```

- Authenticate using pass-the-hash authentication instead of a password:

```
psexec.py -hashes {{LM_Hash}}:{{NT_Hash}} {{domain}}/
{{username}}@{{target}}
```

- Use Kerberos authentication for the target:

```
psexec.py -k -no-pass {{domain}}/{{username}}@{{target}}
```

- Specify the IP address of the domain controller:

```
psexec.py -dc-ip {{domain_controller_ip}} {{domain}}/
{{username}}:{{password}}@{{target}}
```

psgrep

Search running processes with **`grep`**.

More information: https://jvz.github.io/psgrep.

- Find process lines containing a specific string:

```
psgrep {{process_name}}
```

- Find process lines containing a specific string, excluding headers:

```
psgrep -n {{process_name}}
```

- Search using a simplified format (PID, user, command):

```
psgrep -s {{process_name}}
```

psidtopgm

Convert PostScript image data to a PGM image.

More information: https://netpbm.sourceforge.net/doc/psidtopgm.html.

- Convert the image data in a PS file to a PGM image of the specified dimensions and quality:

```
psidtopgm {{width}} {{height}} {{bits_per_sample}} {{path/to/file.ps}} > {{path/to/image.pgm}}
```

psql

PostgreSQL client.

More information: https://www.postgresql.org/docs/current/app-psql.html.

- Connect to the database. By default, it connects to the local socket using port 5432 with the currently logged in user:

```
psql {{database}}
```

- Connect to the database on given server host running on given port with given username, without a password prompt:

```
psql {{[-h|--host]}} {{host}} {{[-p|--port]}} {{port}} {{[-U|--username]}} {{username}} {{database}}
```

- Connect to the database; user will be prompted for password:

```
psql {{[-h|--host]}} {{host}} {{[-p|--port]}} {{port}} {{[-U|--username]}} {{username}} {{[-W|--password]}} {{database}}
```

- Execute a single SQL query or PostgreSQL command on the given database (useful in shell scripts):

```
psql {{[-c|--command]}} '{{query}}' {{database}}
```

- Execute commands from a file on the given database:

```
psql {{database}} {{[-f|--file]}} {{path/to/file.sql}}
```

pssh

Parallel SSH program.

More information: https://manned.org/pssh.

- Run a command on two hosts, and print its output on each server inline:

```
pssh {{[-i|--inline]}} {{[-H|--host]}} "{{host1}}
{{host2}}" {{hostname --ip-addresses}}
```

- Run a command and save the output to separate files:

```
pssh {{[-H|--host]}} {{host1}} {{[-H|--host]}} {{host2}} {{[-
o|--outdir]}} {{path/to/output_directory}} {{hostname --ip-
addresses}}
```

- Run a command on multiple hosts, specified in a new-line separated file:

```
pssh {{[-i|--inline]}} {{[-h|--hosts]}} {{path/to/
hosts_file}} {{hostname --ip-addresses}}
```

- Run a command as root (this asks for the root password):

```
pssh {{[-i|--inline]}} {{[-h|--hosts]}} {{path/to/
hosts_file}} {{[-A|--askpass]}} {{[-l|--user]}}
{{root_username}} {{hostname --ip-addresses}}
```

- Run a command with extra SSH arguments:

```
pssh {{[-i|--inline]}} {{[-h|--hosts]}} {{path/to/
hosts_file}} {{[-x|--extra-arg]}} "{{-O
VisualHostKey=yes}}" {{hostname --ip-addresses}}
```

- Run a command limiting the number of parallel connections to 10:

```
pssh {{[-i|--inline]}} {{[-h|--hosts]}} {{path/to/
hosts_file}} {{[-p|-par]}} {{10}} '{{cd dir; ./script.sh;
exit}}'
```

pstopnm

Convert a PostScript file to a PNM image.

More information: https://netpbm.sourceforge.net/doc/pstopnm.html.

- Convert a PS file to PNM images, storing page N of the input to `path/to/fileN.ppm`:

```
pstopnm {{path/to/file.ps}}
```

- Explicitly specify the output format:

```
pstopnm -{{pbm|pgm|ppm}} {{path/to/file.ps}}
```

- Specify the resolution of the output in dots per inch:

```
pstopnm -dpi {{n}} {{path/to/file.ps}}
```

psysh

A runtime developer console, interactive debugger and REPL for PHP.

More information: https://psysh.org.

- Open a shell in the current directory:

```
psysh
```

- Open a shell in a specific directory:

```
psysh --cwd {{path/to/directory}}
```

- Use a specific configuration file:

```
psysh --config {{path/to/file}}
```

pt

Platinum Searcher.

A code search tool similar to **ag**.

More information: https://github.com/monochromegane/the_platinum_searcher.

- Find files containing "foo" and print the files with highlighted matches:

```
pt {{foo}}
```

- Find files containing "foo" and display count of matches in each file:

```
pt {{[-c|--count]}} {{foo}}
```

- Find files containing "foo" as a whole word and ignore its case:

```
pt {{[-wi|--word-regexp --ignore-case]}} {{foo}}
```

- Find "foo" in files with a given extension using a `regex`:

```
pt {{[-G|--file-search-regexp]}}='{{\.bar$}}' {{foo}}
```

- Find files whose contents match the `regex`, up to 2 directories deep:

```
pt --depth={{2}} -e '{{^ba[rz]*$}}'
```

ptargrep

Find **regex** patterns in tar archive files.

More information: https://manned.org/ptargrep.

- Search for a pattern within one or more tar archives:

```
ptargrep "{{search_pattern}}" {{path/to/file1 path/to/file2 ...}}
```

- Extract to the current directory using the basename of the file from the archive:

```
ptargrep {{[-b|--basename]}} "{{search_pattern}}" {{path/to/file}}
```

- Search for a case-insensitive pattern matching within a tar archive:

```
ptargrep {{[-i|--ignore-case]}} "{{search_pattern}}" {{path/to/file}}
```

ptpython

A better Python REPL.

More information: https://github.com/prompt-toolkit/ptpython.

- Start a REPL (interactive shell):

```
ptpython
```

- Execute a specific Python file:

```
ptpython {{path/to/file.py}}
```

- Execute a specific Python file and start a REPL:

```
ptpython {{[-i|--interactive]}} {{path/to/file.py}}
```

- Open the menu:

```
<F2>
```

- Open the history page:

```
<F3>
```

- Toggle paste mode:

```
<F6>
```

- Quit:

```
<Ctrl d>
```

ptpython3

This command is an alias of **ptpython**.

- View documentation for the original command:

```
tldr ptpython
```

pueue add

Enqueue a task for execution.

More information: https://github.com/Nukesor/pueue.

- Add any command to the default queue:

```
pueue add {{command}}
```

- Pass a list of flags or arguments to a command when enqueuing:

```
pueue add -- {{command --arg -f}}
```

- Add a command but do not start it if it's the first in a queue:

```
pueue add {{[-s|--stashed]}} -- {{rsync --archive --
compress /local/directory /remote/directory}}
```

- Add a command to a group and start it immediately, see `pueue group` to manage groups:

```
pueue add {{[-i|--immediate]}} {{[-g|--group]}}
"{{CPU_intensive}}" -- {{ffmpeg -i input.mp4 frame_%d.png}}
```

- Add a command and start it after commands 9 and 12 finish successfully:

```
pueue add {{[-a|--after]}} {{9}} {{12}} {{[-g|--group]}}
"{{torrents}}" -- {{transmission-cli torrent_file.torrent}}
```

- Add a command with a label after some delay has passed, see `pueue enqueue` for valid datetime formats:

```
pueue add {{[-l|--label]}} "{{compressing large file}}" {{[-
d|--delay]}} "{{wednesday 10:30pm}}" -- "{{7z a
compressed_file.7z large_file.xml}}"
```

pueue clean

Remove all finished tasks from the list and clear logs.

More information: https://github.com/Nukesor/pueue.

- Remove finished tasks and clear logs:

```
pueue clean
```

- Only clean commands that finished successfully:

```
pueue clean {{[-s|--successful-only]}}
```

pueue completions

Generate shell completion files for Bash, Elvish, fish, PowerShell, and Zsh.

More information: https://github.com/Nukesor/pueue.

- Generate completions for Bash:

```
sudo pueue completions bash {{/usr/share/bash-completion/completions/pueue.bash}}
```

- Generate completions for Zsh:

```
sudo pueue completions zsh {{/usr/share/zsh/site-functions}}
```

- Generate completions for fish:

```
sudo pueue completions fish {{/usr/share/fish/completions}}
```

pueue edit

Edit the command or path of a stashed or queued task.

More information: https://github.com/Nukesor/pueue.

- Edit a task, see `pueue status` to get the task ID:

```
pueue edit {{task_id}}
```

- Edit the path from which a task is executed:

```
pueue edit {{task_id}} --path
```

- Edit a command with the specified editor:

```
EDITOR={{nano}} pueue edit {{task_id}}
```

pueue enqueue

Enqueue stashed tasks.

See also: **`pueue stash`**.

More information: https://github.com/Nukesor/pueue.

- Enqueue multiple stashed tasks at once:

```
pueue enqueue {{task_id}} {{task_id}}
```

- Enqueue a stashed task after 60 seconds:

```
pueue enqueue {{[-d|--delay]}} {{60}} {{task_id}}
```

- Enqueue a stashed task next Wednesday:

```
pueue enqueue {{[-d|--delay]}} {{wednesday}} {{task_id}}
```

- Enqueue a stashed task after four months:

```
pueue enqueue {{[-d|--delay]}} "4 months" {{task_id}}
```

- Enqueue a stashed task on 2021-02-19:

```
pueue enqueue {{[-d|--delay]}} {{2021-02-19}} {{task_id}}
```

- List all available date/time formats:

```
pueue enqueue {{[-h|--help]}}
```

pueue follow

Follow the output of a currently running task.

See also: **`pueue log`**.

More information: https://github.com/Nukesor/pueue.

- Follow the output of a task (`stdout` + `stderr`):

```
pueue follow {{task_id}}
```

- Follow `stderr` of a task:

```
pueue follow --err {{task_id}}
```

pueue group

Display, add or remove groups.

More information: https://github.com/Nukesor/pueue.

- Show all groups with their statuses and number of parallel jobs:

```
pueue group
```

- Add a custom group:

```
pueue group add "{{group_name}}"
```

- Remove a group and move its tasks to the default group:

```
pueue group remove "{{group_name}}"
```

pueue help

Display help for subcommands.

More information: https://github.com/Nukesor/pueue.

- Show all available subcommands and flags:

```
pueue help
```

- Display help for a specific subcommand:

```
pueue help {{subcommand}}
```

pueue kill

Kill running tasks or whole groups.

More information: https://github.com/Nukesor/pueue.

- Kill all tasks in the default group:

```
pueue kill
```

- Kill a specific task:

```
pueue kill {{task_id}}
```

- Kill a task and terminate all its child processes:

```
pueue kill --children {{task_id}}
```

- Kill all tasks in a group and pause the group:

```
pueue kill {{[-g|--group]}} {{group_name}}
```

- Kill all tasks across all groups and pause all groups:

```
pueue kill {{[-a|--all]}}
```

pueue log

Display the log output of 1 or more tasks.

See also: **`pueue status`**.

More information: https://github.com/Nukesor/pueue.

- Show the last few lines of output from all tasks:

```
pueue log
```

- Show the full output of a task:

```
pueue log {{task_id}}
```

- Show the last few lines of output from several tasks:

```
pueue log {{task_id}} {{task_id}}
```

- Print a specific number of lines from the tail of output:

```
pueue log {{[-l|--lines]}} {{number_of_lines}} {{task_id}}
```

pueue parallel

Set the amount of allowed parallel tasks.

More information: https://github.com/Nukesor/pueue.

- Set the maximum number of tasks allowed to run in parallel, in the default group:

```
pueue parallel {{max_number_of_parallel_tasks}}
```

- Set the maximum number of tasks allowed to run in parallel, in a specific group:

```
pueue parallel {{[-g|--group]}} {{group_name}} {{maximum_number_of_parallel_tasks}}
```

pueue pause

Pause running tasks or groups.

See also: **`pueue start`**.

More information: https://github.com/Nukesor/pueue.

- Pause all tasks in the default group:

```
pueue pause
```

- Pause a running task:

```
pueue pause {{task_id}}
```

- Pause a running task and stop all its direct children:

```
pueue pause --children {{task_id}}
```

- Pause all tasks in a group and prevent it from starting new tasks:

```
pueue pause {{[-g|--group]}} {{group_name}}
```

- Pause all tasks and prevent all groups from starting new tasks:

```
pueue pause {{[-a|--all]}}
```

pueue remove

Remove tasks from the list. Running or paused tasks need to be killed first.

More information: https://github.com/Nukesor/pueue.

- Remove a killed or finished task:

```
pueue remove {{task_id}}
```

- Remove multiple tasks at once:

```
pueue remove {{task_id}} {{task_id}}
```

pueue reset

Kill everything and reset.

More information: https://github.com/Nukesor/pueue.

- Kill all tasks and remove everything (logs, status, groups, task IDs):

```
pueue reset
```

- Kill all tasks, terminate their children, and reset everything:

```
pueue reset --children
```

- Reset without asking for confirmation:

```
pueue reset {{[-f|--force]}}
```

pueue restart

Restart tasks.

More information: https://github.com/Nukesor/pueue.

- Restart a specific task:

```
pueue restart {{task_id}}
```

- Restart multiple tasks at once, and start them immediately (do not enqueue):

```
pueue restart {{[-k|--immediately]}} {{task_id}} {{task_id}}
```

- Restart a specific task from a different path:

```
pueue restart --edit-path {{task_id}}
```

- Edit a command before restarting:

```
pueue restart {{[-e|--edit]}} {{task_id}}
```

- Restart a task in-place (without enqueuing as a separate task):

```
pueue restart {{[-i|--in-place]}} {{task_id}}
```

- Restart all failed tasks and stash them:

```
pueue restart {{[-a|--all-failed]}} --stashed
```

pueue send

Send input to a task.

More information: https://github.com/Nukesor/pueue.

- Send input to a running command:

```
pueue send {{task_id}} "{{input}}"
```

- Send confirmation to a task expecting y/N (e.g. APT, cp):

```
pueue send {{task_id}} {{y}}
```

pueue shutdown

Remotely shut down the daemon.

Only use this subcommand if the daemon isn't started by a service manager.

More information: https://github.com/Nukesor/pueue.

- Shutdown the daemon without a service manager:

```
pueue shutdown
```

pueue start

Resume operation of tasks or groups of tasks.

See also: **pueue pause**.

More information: https://github.com/Nukesor/pueue.

- Resume all tasks in the default group:

```
pueue start
```

- Resume a specific task:

```
pueue start {{task_id}}
```

- Resume multiple tasks at once:

```
pueue start {{task_id}} {{task_id}}
```

- Resume all tasks and start their children:

```
pueue start {{[-a|--all]}} --children
```

- Resume all tasks in a specific group:

```
pueue start group {{group_name}}
```

pueue stash

Stash tasks to prevent them starting automatically.

See also: **`pueue start`**, **`pueue enqueue`**.

More information: https://github.com/Nukesor/pueue.

- Stash an enqueued task:

```
pueue stash {{task_id}}
```

- Stash multiple tasks at once:

```
pueue stash {{task_id}} {{task_id}}
```

- Start a stashed task immediately:

```
pueue start {{task_id}}
```

- Enqueue a task to be executed when preceding tasks finish:

```
pueue enqueue {{task_id}}
```

pueue status

Display the current status of all tasks.

More information: https://github.com/Nukesor/pueue.

- Show the status of all tasks:

```
pueue status
```

- Show the status of a specific group:

```
pueue status {{[-g|--group]}} {{group_name}}
```

pueue switch

Switches the queue position of two enqueued or stashed commands.

More information: https://github.com/Nukesor/pueue.

- Switch the priority of two tasks:

```
pueue switch {{task_id1}} {{task_id2}}
```

pueue

A task management tool for sequential and parallel execution of long-running tasks.

Some subcommands such as **add** have their own usage documentation.

More information: https://github.com/Nukesor/pueue#how-to-use-it.

- Add a command to the task list:

```
pueue add -- "{{command}}"
```

- List tasks in the task list:

```
pueue status
```

- Send data to a task's `stdin`:

```
pueue send {{task_id}} "{{hello}}"
```

- View a task's `stdout` and `stderr`, as well as basic information about that task:

```
pueue log {{task_id}}
```

- Create a task group:

```
pueue group --add {{group_name}}
```

- Kill a task:

```
pueue kill {{task_id}}
```

- Set maximum amount of parallel tasks (queued tasks are started as needed to meet this limit):

```
pueue parallel {{number_of_parallel_tasks}}
```

- Edit the command line of a stopped task in the default editor (as specified by `$EDITOR`):

```
pueue edit {{task_id}}
```

pulumi about

Print information about the Pulumi environment.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_about/.

- Print information about the Pulumi environment:

```
pulumi about
```

- Print information about the Pulumi environment in JSON format:

```
pulumi about {{[-j|--json]}}
```

- Print information about the Pulumi environment of a specific stack:

```
pulumi about {{[-s|--stack]}} {{stack_name}}
```

- Display help:

```
pulumi about {{[-h|--help]}}
```

pulumi cancel

Cancel a stack's currently running update, if any.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_cancel/.

- Cancel a stack's currently running update, if any:

```
pulumi cancel {{stack_name}}
```

- Skip confirmation prompts, and proceed with cancellation anyway:

```
pulumi cancel {{[-y|--yes]}}
```

- Display help:

```
pulumi cancel {{[-h|--help]}}
```

pulumi config

Manage configuration of a Pulumi stack.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_config/.

- View current configuration in JSON format:

```
pulumi config {{[-j|--json]}}
```

- View configuration for a specified stack:

```
pulumi config {{[-s|--stack]}} {{stack_name}}
```

- Get the value of a configuration key:

```
pulumi config get {{key}}
```

- Remove a configuration value:

```
pulumi config rm {{key}}
```

- Set a value for a configuration key from a file:

```
cat {{path/to/file}} | pulumi config set {{key}}
```

- Set a secret value (e.g. API key) for a configuration key and store/display as ciphertext:

```
pulumi config set --secret {{key}} {{S3cr37_value}}
```

- Remove multiple configuration values from a specified configuration file:

```
pulumi config --config-file {{path/to/file}} rm-all {{key1 key2 ...}}
```

pulumi console

Open a stack in the Pulumi Console (cloud or self-hosted).

It gives you instructions to migrate if not available for your backend.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_console/.

- Open the current stack:

  ```
  pulumi console
  ```

- Open the console for a specific stack:

  ```
  pulumi console {{[-s|--stack]}} {{stack_name}}
  ```

- Display help:

  ```
  pulumi console {{[-h|--help]}}
  ```

pulumi destroy

Destroy all existing resources in a stack.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_destroy/.

- Destroy all resources in the current stack:

  ```
  pulumi destroy
  ```

- Destroy all resources in a specific stack:

  ```
  pulumi destroy {{[-s|--stack]}} {{stack}}
  ```

- Automatically approve and destroy resources after previewing:

  ```
  pulumi destroy {{[-y|--yes]}}
  ```

- Exclude protected resources from being destroyed:

  ```
  pulumi destroy --exclude-protected
  ```

- Remove the stack and its configuration file after all resources in the stack are deleted:

  ```
  pulumi destroy --remove
  ```

- Continue destroying the resources, even if an error is encountered:

  ```
  pulumi destroy --continue-on-error
  ```

pulumi down

This command is an alias of **`pulumi destroy`**.

- View documentation for the original command:

```
tldr pulumi destroy
```

pulumi env

Manage Pulumi environments.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_env/.

- List all environments:

```
pulumi env ls
```

- Create an environment:

```
pulumi env init {{environment_name}}
```

- Set a value in an environment:

```
pulumi env set {{environment_name}} {{key}} {{value}}
```

- Edit an environment definition:

```
pulumi env edit {{environment_name}}
```

- Delete a value from an environment:

```
pulumi env rm {{environment_name}} {{key}}
```

- Delete an environment entirely:

```
pulumi env rm {{environment_name}}
```

- Display help:

```
pulumi env {{[-h|--help]}}
```

pulumi gen-completion

Generate completion scripts for the Pulumi CLI.

Supported shells are Bash, Zsh, fish.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_gen-completion/.

- Generate completion scripts:

```
pulumi gen-completion {{bash|zsh|fish}}
```

pulumi import

Import resources into an existing stack.

Read syntax for your cloud provider: https://www.pulumi.com/registry/.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_import/.

- Generate the resource definition with a given name for an existing provider resource:

```
pulumi import {{type_token}} {{name}} {{id}}
```

- Import an existing AWS user as a `pulumi` resource:

```
pulumi import aws:iam/user:User {{my_user_resource}} {{id}}
```

- Import an existing Cloudflare worker:

```
pulumi import cloudflare:index/workersScript:WorkersScript {{my_worker_script}} {{account_id/script_name}}
```

- Import from a JSON file for bulk import operations and output to a file instead of `stdout`:

```
pulumi import --file {{path/to/file.json}} --out {{path/to/file}}
```

pulumi install

Install packages and plugins for the current program or policy pack.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_install.

- Install packages and plugins:

  ```
  pulumi install
  ```

- Install packages and plugins but skip dependencies:

  ```
  pulumi install --no-dependencies
  ```

- Install packages and reinstall existing plugins:

  ```
  pulumi install --reinstall
  ```

pulumi login

Log in to the Pulumi cloud.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_login/.

- Log in to the managed Pulumi Cloud backend, defaults to `https://api.pulumi.com`:

  ```
  pulumi login
  ```

- Log in to a self-hosted Pulumi Cloud backend on a specified URL:

  ```
  pulumi login {{url}}
  ```

- Use Pulumi locally, independent of a cloud backend:

  ```
  pulumi login {{[-l|--local]}}
  ```

- Display help:

  ```
  pulumi login {{[-h|--help]}}
  ```

pulumi logout

Log out of the Pulumi cloud.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_logout/.

- Log out of a specific cloud backend:

```
pulumi logout {{url}}
```

- Log out of all backends simultaneously:

```
pulumi logout --all
```

- Log out of using local mode:

```
pulumi logout {{[-l|--local]}}
```

- Display help:

```
pulumi logout {{[-h|--help]}}
```

pulumi new

Create a new Pulumi project.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_new/.

- Choose a template interactively:

  ```
  pulumi new
  ```

- Create a project from a specific template (e.g `azure-python`):

  ```
  pulumi new {{provided-template}}
  ```

- Create a project from a local file:

  ```
  pulumi new {{path/to/templates/aws-typescript}}
  ```

- Create a project from a Git repository:

  ```
  pulumi new {{url}}
  ```

- Use the specified secrets provider with the backend:

  ```
  pulumi new --secrets-provider {{passphrase}}
  ```

pulumi org

Manage Pulumi Organization configuration.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_org/.

- Display the default organization and current backend:

  ```
  pulumi org
  ```

- Display the default organization:

  ```
  pulumi org get-default
  ```

- Set the default organization:

  ```
  pulumi org set-default {{organization_name}}
  ```

- Search for resources in Pulumi Cloud using Pulumi AI with a plaintext natural language query:

  ```
  pulumi org search ai {{[-q|--query]}} "{{show me all load balancers in my organization}}"
  ```

- Display help:

  ```
  pulumi org {{[-h|--help]}}
  ```

pulumi plugin

Manage language and resource provider plugins manually.

Other commands manage these automatically.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_plugin/.

- List all plugins on the downloaded cache:

  ```
  pulumi plugin ls
  ```

- List plugins being used by the current project in JSON format:

  ```
  pulumi plugin {{[-p|--project]}} {{[-j|--json]}}
  ```

- Install a plugin kind (e.g resource) with the latest version or a specific one:

  ```
  pulumi plugin install {{kind}} {{name}} {{version}}
  ```

- Remove a plugin kind (e.g. resource) and interactively pick a version or provide a specific one:

  ```
  pulumi plugin rm {{kind}} {{name}} {{version}}
  ```

- Display help:

  ```
  pulumi plugin {{[-h|--help]}}
  ```

pulumi policy

Manage resource policies on Pulumi Cloud (Business Critical) or locally (no organization sub-commands).

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_policy/.

- Create a new Pulumi Policy Pack from a template or URL:

```
pulumi policy new --dir {{path/to/directory}} {{template|
url}}
```

- Validate syntax of a policy. To check a policy against a project use `pulumi preview`:

```
pulumi policy validate-config {{organization_name}}/
{{policy_pack_name}} {{version}}
```

- List all policies for an organization:

```
pulumi policy ls {{[-j|--json]}} {{organization_name}}
```

- Publish a policy to the Pulumi Cloud:

```
pulumi policy publish {{organization_name}}
```

- Enable a policy with a specific version:

```
pulumi policy enable {{organization_name}}/
{{policy_pack_name}} {{latest|version}}
```

- Disable a policy with a specific version (defaults to all versions):

```
pulumi policy disable {{organization_name}}/
{{policy_pack_name}} --version {{version}}
```

- Display help:

```
pulumi policy {{[-h|--help]}}
```

pulumi preview

Show a preview of updates to a stack's resources.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_preview/.

- Show a preview of updates to a stack's resources:

```
pulumi preview
```

- Show a preview of updates to a stack's resources in JSON format:

```
pulumi preview {{[-j|--json]}}
```

- Preview updates as a rich diff showing overall changes:

```
pulumi preview --diff
```

- Preview updates using a Policy Pack (without Pulumi Cloud, best on CI/CD):

```
pulumi preview --policy-pack {{path/to/directory}}
```

- Display help:

```
pulumi preview {{[-h|--help]}}
```

pulumi refresh

Refresh the resources in a stack.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_refresh/.

- Compare the current stack's state with the state in the cloud provider and adopt any changes into the current stack:

```
pulumi refresh
```

- Refresh resources in the current stack and show the operation as a rich diff:

```
pulumi refresh --diff
```

- Refresh resources in the current stack and return an error if any changes occur during the refresh:

```
pulumi refresh --expect-no-changes
```

- Only show a preview of the refresh, but don't perform the refresh itself:

```
pulumi refresh --preview-only
```

- The name of the stack to operate on (defaults to the current stack):

```
pulumi refresh {{[-s|--stack]}} {{stack_name}}
```

- Display help:

```
pulumi refresh {{[-h|--help]}}
```

pulumi schema

Check a Pulumi package schema for errors.

Schema reference: https://www.pulumi.com/docs/iac/extending-pulumi/schema/.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_schema/.

- Check a package schema:

  ```
  pulumi schema check {{path/to/file}}
  ```

- Check a package schema without failing if the reference to a type is missing:

  ```
  pulumi schema check --allow-dangling-references {{path/to/file}}
  ```

- Display help:

  ```
  pulumi schema check {{[-h|--help]}}
  ```

pulumi stack hist

This command is an alias of **`pulumi stack history`**.

- View documentation for the original command:

```
tldr pulumi stack history
```

pulumi stack history

Display history for a stack.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_stack_history/.

- Display history for the current stack:

```
pulumi stack {{[hist|history]}}
```

- Display history for the current stack showing full dates instead of relative dates:

```
pulumi stack {{[hist|history]}} --full-dates
```

- Display history for the current stack in JSON format:

```
pulumi stack {{[hist|history]}} {{[-j|--json]}}
```

- Display history for a specific stack:

```
pulumi stack {{[hist|history]}} {{[-s|--stack]}} {{stack_name}}
```

- Display help:

```
pulumi stack {{[hist|history]}} {{[-h|--help]}}
```

pulumi stack

Manage stacks and view stack state.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_stack/.

- Create a new stack:

```
pulumi stack init {{stack_name}}
```

- Show the stack state along with resource URNs:

```
pulumi stack {{[-u|--show-urns]}}
```

- List stacks in the current project:

```
pulumi stack ls
```

- List stacks across all projects:

```
pulumi stack ls {{[-a|--all]}}
```

- Select an active stack:

```
pulumi stack select {{stack_name}}
```

- Delete a stack:

```
pulumi stack rm {{stack_name}}
```

- Show stack outputs, including secrets, in plaintext:

```
pulumi stack output --show-secrets
```

- Export the stack state to a JSON file:

```
pulumi stack export --file {{path/to/file.json}}
```

pulumi state

Edit the current stack's state.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_state/.

- Delete a resource from the current stack's state:

  ```
  pulumi state delete
  ```

- Move a resource from the current stack to another:

  ```
  pulumi state move {{resource_urn}} --dest {{stack_name}}
  ```

- Rename a resource in the current stack's state:

  ```
  pulumi state rename
  ```

- Repair an invalid state:

  ```
  pulumi state repair
  ```

- Edit a stack's state in the editor specified by the `EDITOR` environment variable:

  ```
  pulumi state edit --stack {{stack_name}}
  ```

- Display help:

  ```
  pulumi state {{[-h|--help]}}
  ```

pulumi up

Create or update the resources in a stack.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_up/.

- Preview and deploy changes to a program and/or infrastructure:

  ```
  pulumi up
  ```

- Automatically approve and perform the update after previewing it:

  ```
  pulumi up {{[-y|--yes]}}
  ```

- Preview and deploy changes in a specific stack:

  ```
  pulumi up {{[-s|--stack]}} {{stack}}
  ```

- Refresh the state of the stack's resources before updating:

  ```
  pulumi up {{[-r|--refresh]}}
  ```

- Don't display stack outputs:

  ```
  pulumi up --suppress-outputs
  ```

- Continue updating the resources, even if an error is encountered:

  ```
  pulumi up --continue-on-error
  ```

- Display help:

  ```
  pulumi up {{[-h|--help]}}
  ```

pulumi update

This command is an alias of **`pulumi up`**.

- View documentation for the original command:

```
tldr pulumi up
```

pulumi version

Display Pulumi's version number.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_version/.

- Display version:

```
pulumi version
```

- Display help:

```
pulumi version {{[-h|--help]}}
```

pulumi whoami

Print information about the currently logged in user.

More information: https://www.pulumi.com/docs/iac/cli/commands/pulumi_whoami/.

- Print the username of the currently logged in user:

```
pulumi whoami
```

- Print detailed information about the currently logged in user:

```
pulumi whoami {{[-v|--verbose]}}
```

- Print detailed information about the currently logged in user as JSON:

```
pulumi whoami {{[-j|--json]}}
```

- Display help:

```
pulumi whoami {{[-h|--help]}}
```

pulumi

Define infrastructure on any cloud using familiar programming languages.

Some subcommands such as **up** have their own usage documentation.

More information: https://www.pulumi.com/docs/iac/cli/.

- Create a new project using a template:

  `pulumi new`

- Create a new stack using an isolated deployment target:

  `pulumi stack init`

- Configure variables (e.g. keys, regions, etc.) interactively:

  `pulumi config`

- Preview and deploy changes to a program and/or infrastructure:

  `pulumi up`

- Preview deployment changes without performing them (dry-run):

  `pulumi preview`

- Destroy a program and its infrastructure:

  `pulumi destroy`

- Use Pulumi locally, independent of a Pulumi Cloud:

  `pulumi login {{[-l|--local]}}`

pup

HTML parsing tool.

More information: https://github.com/ericchiang/pup.

- Transform a raw HTML file into a cleaned, indented, and colored format:

```
cat {{index.html}} | pup --color
```

- Filter HTML by element tag name:

```
cat {{index.html}} | pup '{{tag}}'
```

- Filter HTML by ID:

```
cat {{index.html}} | pup '{{div#id}}'
```

- Filter HTML by attribute value:

```
cat {{index.html}} | pup '{{input[type="text"]}}'
```

- Print all text from the filtered HTML elements and their children:

```
cat {{index.html}} | pup '{{div}} text{}'
```

- Print HTML as JSON:

```
cat {{index.html}} | pup '{{div}} json{}'
```

puppet agent

Retrieves the client configuration from a Puppet server and applies it to the local host.

More information: https://github.com/puppetlabs/puppet/blob/main/references/man/agent.md.

- Register a node at a Puppet server and apply the received catalog:

```
puppet agent --test --server {{puppetserver_fqdn}} --serverport {{port}} --waitforcert {{poll_time}}
```

- Run the agent in the background (uses settings from `puppet.conf`):

```
puppet agent
```

- Run the agent once in the foreground, then exit:

```
puppet agent --test
```

- Run the agent in dry-mode:

```
puppet agent --test --noop
```

- Log every resource being evaluated (even if nothing is being changed):

```
puppet agent --test --evaltrace
```

- Disable the agent:

```
puppet agent --disable "{{message}}"
```

- Enable the agent:

```
puppet agent --enable
```

puppet apply

Apply Puppet manifests locally.

More information: https://github.com/puppetlabs/puppet/blob/main/references/man/apply.md.

- Apply a manifest:

```
puppet apply {{path/to/manifest}}
```

- Execute puppet code:

```
puppet apply --execute {{code}}
```

- Use a specific module and hiera configuration file:

```
puppet apply --modulepath {{path/to/directory}} --
hiera_config {{path/to/file}} {{path/to/manifest}}
```

puppet

Manage and automate the configuration of servers.

Puppet uses a declarative language to define system configurations and apply them automatically.

Some subcommands, such as **`agent`** and **`apply`**, have their own usage documentation.

More information: https://github.com/puppetlabs/puppet/blob/main/references/man/overview.md.

- Apply a Puppet manifest file to configure the system:

  ```
  puppet apply {{path/to/file.pp}}
  ```

- Apply a manifest in no operation (dry-run) mode to preview changes:

  ```
  puppet apply --noop {{path/to/file.pp}}
  ```

- Validate the syntax of a Puppet manifest:

  ```
  puppet parser validate {{path/to/file.pp}}
  ```

- Run the Puppet agent to fetch and apply configurations from the master:

  ```
  puppet agent {{[-t|--test]}}
  ```

- Display help for a specific subcommand:

  ```
  puppet help {{subcommand}}
  ```

- Display general help:

  ```
  puppet {{[-h|--help]}}
  ```

- Display version:

  ```
  puppet {{[-V|--version]}}
  ```

pushd

Place a directory on a stack so it can be accessed later.

See also: **popd** to switch back to original directory and **dirs** to display directory stack contents.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-pushd.

- Switch to directory and push it on the stack:

```
pushd {{path/to/directory}}
```

- Switch first and second directories on the stack:

```
pushd
```

- Rotate stack by making the 5th element the top of the stack:

```
pushd +4
```

- Rotate the stack 4 times to the left (the current directory stays at the top by replacing the 5th element):

```
pushd -n +4
```

pv

Monitor the progress of data through a pipe.

More information: https://manned.org/pv.

- Print the contents of the file and display a progress bar:

```
pv {{path/to/file}}
```

- Measure the speed and amount of data flow between pipes (`--size` is optional):

```
{{command1}} | pv {{[-s|--size]}}
{{expected_amount_of_data_for_eta}} | {{command2}}
```

- Filter a file, see both progress and amount of output data:

```
pv {{[-cN|--cursor --name]}} in {{path/to/file.txt}} | grep
{{pattern}} | pv {{[-cN|--cursor --name]}} out > {{path/to/
filtered_file.txt}}
```

- Attach to an already running process and see its file reading progress:

```
pv {{[-d|--watchfd]}} {{PID}}
```

- Read an erroneous file, skip errors as `dd conv=sync,noerror` would:

```
pv {{[-EE|--skip-errors --skip-errors]}} {{path/to/
faulty_media}} > {{path/to/image.img}}
```

- Stop reading after reading specified amount of data, rate limit to 1K/s:

```
pv {{[-L|--rate-limit]}} {{1K}} {{[-S|--stop-at-size]}}
{{maximum_file_size_to_be_read}}
```

- Calculate hash (MD5, SHA1, SHA256, etc) of a large file and show progress:

```
pv {{path/to/file}} | {{sha256sum}}
```

pwd

Print the name of current/working directory.

More information: https://www.gnu.org/software/coreutils/manual/html_node/pwd-invocation.html.

- Print the current directory:

```
pwd
```

- Print the current directory, and resolve all symlinks (i.e. show the "physical" path):

```
pwd {{[-P|--physical]}}
```

- Display help:

```
pwd --help
```

pwgen

Generate pronounceable passwords.

More information: https://manned.org/pwgen.

- Generate random password with symbols:

```
pwgen {{[-y|--symbols]}} {{length}}
```

- Generate secure, hard-to-memorize passwords:

```
pwgen {{[-s|--secure]}} {{length}}
```

- Generate password with at least one capital letter in them:

```
pwgen {{[-c|--capitalize]}} {{length}}
```

pwqgen

Generate a random passphrase.

See also: **`libpasswdqc`**.

More information: https://manned.org/pwqgen.

- Generate a passphrase:

```
pwqgen
```

- Generate a passphrase with a specific bit size between 24 and 136:

```
pwqgen random={{bitsize}}
```

- Use a config file to control password generation:

```
pwqgen config={{path/to/config_file}}
```

- Display help:

```
pwqgen {{[-h|--help]}}
```

- Display version:

```
pwqgen --version
```

pwsh

Shell and scripting language designed especially for system administration.

This command refers to PowerShell version 6 and above (also known as PowerShell Core and cross-platform PowerShell).

To use the original Windows version (5.1 and below, also known as the legacy Windows PowerShell), use **`powershell`** instead of **`pwsh`**.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/about/about_pwsh.

- Start an interactive shell session:

```
pwsh
```

- Start an interactive shell session without loading startup configs:

```
pwsh -NoProfile
```

- Execute specific commands:

```
pwsh -Command "{{echo 'powershell is executed'}}"
```

- Execute a specific script:

```
pwsh -File {{path/to/script.ps1}}
```

- Start a session with a specific version of PowerShell:

```
pwsh -Version {{version}}
```

- Prevent a shell from exit after running startup commands:

```
pwsh -NoExit
```

- Describe the format of data sent to PowerShell:

```
pwsh -InputFormat {{Text|XML}}
```

- Determine how an output from PowerShell is formatted:

```
pwsh -OutputFormat {{Text|XML}}
```

py-spy

A sampling profiler for Python programs.

More information: https://github.com/benfred/py-spy.

- Show a live view of the functions that take the most execution time of a running process:

```
py-spy top {{[-p|--pid]}} {{pid}}
```

- Start a program and show a live view of the functions that take the most execution time:

```
py-spy top -- python {{path/to/file.py}}
```

- Produce an SVG flame graph of the function call execution time:

```
py-spy record {{[-o|--output]}} {{path/to/profile.svg}} {{[-p|--pid]}} {{pid}}
```

- Dump the call stack of a running process:

```
py-spy dump {{[-p|--pid]}} {{pid}}
```

pyats shell

Start a pre-loaded pyATS interactive Python Shell to save time in prototyping.

More information: https://pubhub.devnetcloud.com/media/genie-docs/docs/cli/genie_shell.html.

- Open pyATS shell with a defined Testbed file:

  ```
  pyats shell --testbed-file {{path/to/testbed.yaml}}
  ```

- Open pyATS shell with a defined Pickle file:

  ```
  pyats shell --pickle-file {{path/to/pickle.file}}
  ```

- Open pyATS with IPython disabled:

  ```
  pyats shell --no-ipython
  ```

pyats version

View and upgrade the pyATS installation.

More information: https://developer.cisco.com/pyats/.

- Display version of all packages:

```
pyats version check
```

- Display outdated packages:

```
pyats version check --outdated
```

- Update packages to the most recent version:

```
pyats version update
```

- Update or downgrade packages to a specific version:

```
pyats version update {{version}}
```

pyats

A vendor agnostic test automation framework by Cisco Systems, used for network and systems testing.

More information: https://developer.cisco.com/pyats/.

- Run a `pyATS` subcommand:

  ```
  pyats {{subcommand}}
  ```

- Display help:

  ```
  pyats --help
  ```

- Display help about a specific subcommand:

  ```
  pyats {{subcommand}} --help
  ```

- Display version:

  ```
  pyats version check
  ```

pycodestyle

Check Python code against PEP 8 style conventions.

More information: https://pycodestyle.readthedocs.io.

- Check the style of a single file:

```
pycodestyle {{file.py}}
```

- Check the style of multiple files:

```
pycodestyle {{file1.py file2.py ...}}
```

- Show only the first occurrence of an error:

```
pycodestyle --first {{file.py}}
```

- Show the source code for each error:

```
pycodestyle --show-source {{file.py}}
```

- Show the specific PEP 8 text for each error:

```
pycodestyle --show-pep8 {{file.py}}
```

pydoc

Display offline Python documentation.

More information: https://docs.python.org/library/pydoc.html.

- Print documentation on a subject (Python keyword, topic, function, module, package, etc.):

```
pydoc {{subject}}
```

- Start an HTTP server on an arbitrary unused port and open a [b]rowser to see the documentation:

```
pydoc -b
```

- Display help:

```
pydoc
```

pydocstyle

Statically check Python scripts for compliance with Python docstring conventions.

More information: https://www.pydocstyle.org/en/latest/.

- Analyze a Python script or all the Python scripts in a specific directory:

```
pydocstyle {{file.py|path/to/directory}}
```

- Show an explanation of each error:

```
pydocstyle {{[-e|--explain]}} {{file.py|path/to/directory}}
```

- Show debug information:

```
pydocstyle {{[-d|--debug]}} {{file.py|path/to/directory}}
```

- Display the total number of errors:

```
pydocstyle --count {{file.py|path/to/directory}}
```

- Use a specific configuration file:

```
pydocstyle --config {{path/to/config_file}} {{file.py|path/
to/directory}}
```

- Ignore one or more errors:

```
pydocstyle --ignore {{D101,D2,D107,...}} {{file.py|path/to/
directory}}
```

- Check for errors from a specific convention:

```
pydocstyle --convention {{pep257|numpy|google}} {{file.py|
path/to/directory}}
```

pyenv virtualenv

Create virtual environments based on one's installed Python distributions.

More information: https://github.com/pyenv/pyenv-virtualenv.

- Create a new Python 3.6.6 virtual environment:

```
pyenv virtualenv {{3.6.6}} {{virtualenv_name}}
```

- List all existing virtual environments:

```
pyenv virtualenvs
```

- Activate a virtual environment:

```
pyenv activate {{virtualenv_name}}
```

- Deactivate the virtual environment:

```
pyenv deactivate
```

pyenv

Switch between multiple versions of Python easily.

See also: **`asdf`**.

More information: https://github.com/pyenv/pyenv.

- List all available commands:

```
pyenv commands
```

- List all Python versions under the `${PYENV_ROOT}/versions` directory:

```
pyenv versions
```

- List all Python versions that can be installed from upstream:

```
pyenv install --list
```

- Install a Python version under the `${PYENV_ROOT}/versions` directory:

```
pyenv install {{2.7.10}}
```

- Uninstall a Python version under the `${PYENV_ROOT}/versions` directory:

```
pyenv uninstall {{2.7.10}}
```

- Set Python version to be used globally in the current machine:

```
pyenv global {{2.7.10}}
```

- Set Python version to be used in the current directory and all directories below it:

```
pyenv local {{2.7.10}}
```

pyflakes

Checks Python source code files for errors.

More information: https://pypi.org/project/pyflakes.

- Check a single Python file:

```
pyflakes check {{path/to/file.py}}
```

- Check Python files in a specific directory:

```
pyflakes checkPath {{path/to/directory}}
```

- Check Python files in a directory recursively:

```
pyflakes checkRecursive {{path/to/directory}}
```

- Check all Python files found in multiple directories:

```
pyflakes iterSourceCode {{path/to/directory_1}} {{path/to/
directory_2}}
```

pygbag

Package Pygame projects as WebAssembly to run in web browsers.

More information: https://pygame-web.github.io.

- Package a Pygame project and start a local test server:

```
pygbag {{path/to/project_folder}}
```

- Package using Python module syntax:

```
python -m pygbag {{path/to/project_folder}}
```

- Package and build without starting the test server:

```
pygbag {{path/to/project_folder}} --build
```

- Package with a specific template:

```
pygbag {{path/to/project_folder}} --template
{{template_name.tmpl}}
```

- Package and create a ZIP archive for https://itch.io:

```
pygbag {{path/to/project_folder}} --archive
```

- Package with optimization turned off:

```
pygbag {{path/to/project_folder}} --no_opt
```

- Specify a custom port for the test server:

```
pygbag {{path/to/project_folder}} --port {{8080}}
```

- Display help:

```
pygbag {{[-h|--help]}}
```

pygmentize

Python-based syntax highlighter.

More information: https://pygments.org/docs/cmdline/.

- Highlight file syntax and print to `stdout` (language is inferred from the file extension):

```
pygmentize {{file.py}}
```

- Explicitly set the language for syntax highlighting:

```
pygmentize -l {{javascript}} {{input_file}}
```

- List available lexers (processors for input languages):

```
pygmentize -L lexers
```

- Save output to a file in HTML format:

```
pygmentize -f html -o {{output_file.html}} {{input_file.py}}
```

- List available output formats:

```
pygmentize -L formatters
```

- Output an HTML file, with additional formatter options (full page, with line numbers):

```
pygmentize -f html -O "full,linenos=True" -o
{{output_file.html}} {{input_file}}
```

pyinfra

Automates infrastructure at a large scale.

More information: https://docs.pyinfra.com/en/3.x/cli.html.

- Execute a command over SSH:

```
pyinfra {{target_ip_address}} exec --
{{command_name_and_arguments}}
```

- Execute contents of a deploy file on a list of targets:

```
pyinfra {{path/to/target_list.py}} {{path/to/deploy.py}}
```

- Execute commands on locally:

```
pyinfra @local {{path/to/deploy.py}}
```

- Execute commands over Docker:

```
pyinfra @docker/{{container}} {{path/to/deploy.py}}
```

pylint

A Python code linter.

More information: https://pylint.pycqa.org/en/latest/.

- Show lint errors in a file:

```
pylint {{path/to/file.py}}
```

- Lint a package or module (must be importable; no `.py` suffix):

```
pylint {{package_or_module}}
```

- Lint a package from a directory path (must contain an `__init__.py` file):

```
pylint {{path/to/directory}}
```

- Lint a file and use a configuration file (usually named `pylintrc`):

```
pylint --rcfile {{path/to/pylintrc}} {{path/to/file.py}}
```

- Lint a file and disable a specific error code:

```
pylint --disable {{C,W,no-error,design}} {{path/to/file}}
```

pypy

Fast and compliant alternative implementation of the Python language.

More information: https://doc.pypy.org.

- Start a REPL (interactive shell):

```
pypy
```

- Execute script in a given Python file:

```
pypy {{path/to/file.py}}
```

- Execute script as part of an interactive shell:

```
pypy -i {{path/to/file.py}}
```

- Execute a Python expression:

```
pypy -c "{{expression}}"
```

- Run library module as a script (terminates option list):

```
pypy -m {{module}} {{arguments}}
```

- Install a package using pip:

```
pypy -m pip install {{package}}
```

- Interactively debug a Python script:

```
pypy -m pdb {{path/to/file.py}}
```

pystun3

Classic STUN client written in Python.

More information: https://github.com/talkiq/pystun3.

- Make a STUN request:

  ```
  pystun3
  ```

- Make a STUN request and specify the stun server:

  ```
  pystun3 --stun-host {{stun.1und1.de}}
  ```

- Make a STUN request and specify the source port:

  ```
  pystun3 --source-port {{7932}}
  ```

pytest

Run Python tests.

More information: https://docs.pytest.org/en/latest/how-to/usage.html.

- Run tests from specific files:

```
pytest {{path/to/test_file1.py path/to/test_file2.py ...}}
```

- Run tests with names matching a specific [k]eyword expression:

```
pytest -k {{expression}}
```

- Exit as soon as a test fails or encounters an error:

```
pytest --exitfirst
```

- Run tests matching or excluding markers:

```
pytest -m {{marker_name1 and not marker_name2}}
```

- Run until a test failure, continuing from the last failing test:

```
pytest --stepwise
```

- Run tests without capturing output:

```
pytest --capture=no
```

python -m json.tool

Validate and pretty-print JSON data.

Part of Python's standard library.

More information: https://docs.python.org/library/json.html#module-json.tool.

- Pretty-print JSON from a file:

```
python -m json.tool {{path/to/file.json}}
```

- Validate and pretty-print JSON from `stdin`:

```
echo '{{{"key": "value"}}}' | python -m json.tool
```

python

Python language interpreter.

More information: https://docs.python.org/using/cmdline.html.

- Start a REPL (interactive shell):

```
python
```

- Execute a specific Python file:

```
python {{path/to/file.py}}
```

- Execute a specific Python file and start a REPL:

```
python -i {{path/to/file.py}}
```

- Execute a Python expression:

```
python -c "{{expression}}"
```

- Run the script of the specified library module:

```
python -m {{module}} {{arguments}}
```

- Install a package using `pip`:

```
python -m pip install {{package}}
```

- Interactively debug a Python script:

```
python -m pdb {{path/to/file.py}}
```

- Start the built-in HTTP server on port 8000 in the current directory:

```
python -m http.server
```

python3

This command is an alias of **`python`**.

- View documentation for the original command:

  ```
  tldr python
  ```

q

Execute SQL-like queries on CSV and TSV files.

More information: https://harelba.github.io/q.

- Query a CSV file by specifying the delimiter as ',':

```
q {{[-d|--delimiter]}} ',' "SELECT * from {{path/to/file}}"
```

- Query a TSV file:

```
q {{[-t|--tab-delimited]}} "SELECT * from {{path/to/file}}"
```

- Query file with header row:

```
q {{[-d|--delimiter]}} {{delimiter}} {{[-H|--skip-header]}}
"SELECT * from {{path/to/file}}"
```

- Read data from `stdin`; '-' in the query represents the data from `stdin`:

```
{{output}} | q "select * from -"
```

- Join two files (aliased as `f1` and `f2` in the example) on column `c1`, a common column:

```
q "SELECT * FROM {{path/to/file}} f1 JOIN {{path/to/
other_file}} f2 ON (f1.c1 = f2.c1)"
```

- Format output using an output delimiter with an output header line (Note: Command will output column names based on the input file header or the column aliases overridden in the query):

```
q {{[-D|--output-delimiter]}} {{delimiter}} {{[-O|--output-
header]}} "SELECT {{column}} as {{alias}} from {{path/to/
file}}"
```

qalc

Powerful and easy to use calculator.

See also: **bc**.

More information: https://qalculate.github.io/manual/qalc.html.

- Launch in interactive mode:

```
qalc {{[-i|--interactive]}}
```

- Launch in terse mode (print the results only):

```
qalc {{[-t|--terse]}}
```

- Update currency exchange rates:

```
qalc {{[-e|--exrates]}}
```

- Perform calculations non-interactively:

```
qalc {{66+99|2^4|6 feet to cm|1 bitcoin to USD|20 kmph to mph|...}}
```

- List all supported functions/prefixes/units/variables:

```
qalc --{{list-functions|list-prefixes|list-units|list-variables}}
```

- Execute commands from a file:

```
qalc {{[-f|--file]}} {{path/to/file}}
```

qc

Manage and execute command snippets stored in QOwnNotes notes.

See also: **`qownnotes`**.

More information: https://www.qownnotes.org/getting-started/command-line-snippet-manager.html.

- Configure the snippet manager, e.g. to set the security token from QOwnNotes:

  ```
  qc configure
  ```

- Search and print command snippets stored in your `Commands.md` note and all your notes tagged with `commands`:

  ```
  qc search
  ```

- Execute a snippet and show the command before executing:

  ```
  qc exec --command
  ```

- Execute the last snippet and show the command before executing:

  ```
  qc exec --command --last
  ```

- Switch between note folders in QOwnNotes:

  ```
  qc switch
  ```

qcp

Copy files using the default text editor to define the filenames.

More information: https://manned.org/qcp.

- Copy a single file (open an editor with the source filename on the left and the target filename on the right):

```
qcp {{source_file}}
```

- Copy multiple JPEG files:

```
qcp {{*.jpg}}
```

- Copy files, but swap the positions of the source and the target filenames in the editor:

```
qcp {{[-o|--option]}} swap {{*.jpg}}
```

qdbus

Inter-Process Communication (IPC) and Remote Procedure Calling (RPC) mechanism originally developed for Linux.

More information: https://doc.qt.io/qt-5/qtdbus-index.html.

- List available service names:

```
qdbus
```

- List object paths for a specific service:

```
qdbus {{service_name}}
```

- List methods, signals and properties available on a specific object:

```
qdbus {{service_name}} /{{path/to/object}}
```

- Execute a specific method passing arguments and display the returned value:

```
qdbus {{service_name}} /{{path/to/object}} {{method_name}}
{{argument1 argument2 ...}}
```

- Display the current brightness value in a KDE Plasma session:

```
qdbus {{org.kde.Solid.PowerManagement}} {{/org/kde/Solid/
PowerManagement/Actions/BrightnessControl}}
{{org.kde.Solid.PowerManagement.Actions.BrightnessControl.brightness}}
```

- Set a specific brightness to a KDE Plasma session:

```
qdbus {{org.kde.Solid.PowerManagement}} {{/org/kde/Solid/
PowerManagement/Actions/BrightnessControl}}
{{org.kde.Solid.PowerManagement.Actions.BrightnessControl.setBrightnes
{{5000}}
```

- Invoke volume up shortcut in a KDE Plasma session:

```
qdbus {{org.kde.kglobalaccel}} {{/component/kmix}}
{{invokeShortcut}} "{{increase_volume}}"
```

- Gracefully log out and then do nothing, reboot or shut down:

```
qdbus {{org.kde.Shutdown}} {{/Shutdown}} {{logout|
logoutAndReboot|logoutAndShutdown}}
```

qemu-img

Create and manipulate Quick Emulator Virtual HDD images.

More information: https://qemu.readthedocs.io/en/master/tools/qemu-img.html.

- Create disk image with a specific size (in gigabytes):

```
qemu-img create {{path/to/image_file.img}} {{gigabytes}}G
```

- Show information about a disk image:

```
qemu-img info {{path/to/image_file.img}}
```

- Increase or decrease image size:

```
qemu-img resize {{path/to/image_file.img}} {{gigabytes}}G
```

- Dump the allocation state of every sector of the specified disk image:

```
qemu-img map {{path/to/image_file.img}}
```

- Convert a VMware `.vmdk` disk image to a KVM `.qcow2` disk image and display [p]rogress:

```
qemu-img convert -f vmdk -O qcow2 -p {{path/to/
image_file.vmdk}} {{path/to/image_file.qcow2}}
```

- [c]reate an internal snapshot of a KVM `.qcow2` disk image:

```
qemu-img snapshot -c {{snapshot_tag_name}} {{path/to/
image_file.qcow2}}
```

- [a]pply an internal snapshot to a KVM `.qcow2` disk image:

```
qemu-img snapshot -a {{snapshot_tag_name}} {{path/to/
image_file.qcow2}}
```

qemu-system-i386

Emulate the **i386** architecture.

More information: https://www.qemu.org/docs/master/system/target-i386.html.

- Boot from an image emulating the i386 architecture:

```
qemu-system-i386 -hda {{image_name.img}} -m {{4096}}
```

- Boot a QEMU instance from a live ISO image:

```
qemu-system-i386 -hda {{image_name.img}} -m {{4096}} -cdrom
{{os_image.iso}} -boot d
```

- Boot from a physical device (e.g. from USB to test a bootable medium):

```
qemu-system-i386 -hda {{/dev/storage_device}} -m {{4096}}
```

- Do not launch a VNC server:

```
qemu-system-i386 -hda {{image_name.img}} -m {{4096}} -
nographic
```

- Exit non-graphical QEMU:

```
<Ctrl a><x>
```

- List the supported machine types:

```
qemu-system-i386 {{[-M|-machine]}} help
```

qemu-system-riscv64

Emulate **`riscv64`** architecture.

More information: https://www.qemu.org/docs/master/system/target-riscv.html.

- Boot a kernel emulating `riscv64` architecture:

```
qemu-system-riscv64 {{[-M|-machine]}} virt -bios none -kernel {{kernel.elf}} -nographic
```

- List supported machine types:

```
qemu-system-riscv64 {{[-M|-machine]}} help
```

- Exit non-graphical QEMU:

```
<Ctrl a><x>
```

qemu-system-x86_64

Emulate the **x86_64** architecture.

More information: https://www.qemu.org/docs/master/system/target-i386.html.

- Boot from an image emulating the x86_64 architecture:

```
qemu-system-x86_64 -hda {{image_name.img}} -m {{4096}}
```

- Boot a QEMU instance from a live ISO image:

```
qemu-system-x86_64 -hda {{image_name.img}} -m {{4096}} -cdrom {{os_image.iso}} -boot d
```

- Boot from a physical device (e.g. from USB to test a bootable medium):

```
qemu-system-x86_64 -hda {{/dev/storage_device}} -m {{4096}}
```

- Do not launch a VNC server:

```
qemu-system-x86_64 -hda {{image_name.img}} -m {{4096}} -nographic
```

- Exit non-graphical QEMU:

```
<Ctrl a><x>
```

- List the supported machine types:

```
qemu-system-x86_64 {{[-M|-machine]}} help
```

qmake

Generate Makefiles from Qt project files.

More information: https://doc.qt.io/qt-6/qmake-running.html.

- Generate a `Makefile` from a project file in the current directory:

```
qmake
```

- Specify `Makefile` and project file locations:

```
qmake -o {{path/to/Makefile}} {{path/to/project_file.pro}}
```

- Generate a default project file:

```
qmake -project
```

- Compile a project:

```
qmake && make
```

- Enable debug mode:

```
qmake -d
```

- Display help:

```
qmake -help
```

qmmp

An audio player with an interface similar to Winamp or XMMS.

See also: **`clementine`**, **`ncmpcpp`**, **`cmus`**.

More information: https://manned.org/qmmp.

- Launch the GUI:

```
qmmp
```

- Start or stop the currently playing audio:

```
qmmp {{[-t|--play-pause]}}
```

- Seek [f]or[w]ar[d]s or [b]ack[w]ar[d]s a specific amount of time in seconds:

```
qmmp --seek-{{fwd|bwd}} {{time_in_seconds}}
```

- Play the next audio file:

```
qmmp --next
```

- Play the previous audio file:

```
qmmp --previous
```

- Display the current volume:

```
qmmp --volume-status
```

- [inc]rease or [dec]rease the volume of the currently playing audio by 5%:

```
qmmp --volume-{{inc|dec}}
```

qmv

Move files and directories using the default text editor to define the filenames.

More information: https://manned.org/qmv.

- Move a single file (open an editor with the source filename on the left and the target filename on the right):

```
qmv {{source_file}}
```

- Move multiple JPEG files:

```
qmv {{*.jpg}}
```

- Move multiple directories:

```
qmv {{[-d|--directory]}} {{path/to/directory1 path/to/
directory2 path/to/directory3 ...}}
```

- Move all files and directories inside a directory:

```
qmv {{[-R|--recursive]}} {{path/to/directory}}
```

- Move files, but swap the positions of the source and the target filenames in the editor:

```
qmv {{[-o|--option]}} swap {{*.jpg}}
```

- Rename all files and folders in the current directory, but show only target filenames in the editor (you can think of it as a kind of simple mode):

```
qmv {{[-f|--format]}} do .
```

qoitopam

Convert a QOI image (Quite OK Image format) to Netpbm.

More information: https://netpbm.sourceforge.net/doc/qoitopam.html.

- Convert a QOI image to Netpbm:

```
qoitopam {{path/to/image.qoi}} > {{path/to/output.pnm}}
```

QOwnNotes

Markdown note-taking application.

Optionally integrates with the note-taking applications of Nextcloud and ownCloud.

See also: **qc** for managing command snippets.

More information: https://www.qownnotes.org/getting-started/cli-parameters.html.

- Run in portable mode:

  ```
  QOwnNotes --portable
  ```

- Dump settings and other information about the app and environment in GitHub Markdown:

  ```
  QOwnNotes --dump-settings
  ```

- Specify a different context for settings and internal files:

  ```
  QOwnNotes --session {{test}}
  ```

- Trigger a menu action after the application was started:

  ```
  QOwnNotes --action {{actionShow_Todo_List}}
  ```

qpdf

Versatile PDF transformation software.

More information: https://github.com/qpdf/qpdf.

- Extract pages 1-3, 5 and 6-10 from a PDF file and save them as another one:

```
qpdf --empty --pages {{path/to/input.pdf}} {{1-3,5,6-10}} --
{{path/to/output.pdf}}
```

- Merge (concatenate) all the pages of multiple PDF files and save the result as a new PDF:

```
qpdf --empty --pages {{path/to/file1.pdf file2.pdf ...}} --
{{path/to/output.pdf}}
```

- Merge (concatenate) given pages from a list of PDF files and save the result as a new PDF:

```
qpdf --empty --pages {{path/to/file1.pdf}} {{1,6-8}} {{path/
to/file2.pdf}} {{3,4,5}} -- {{path/to/output.pdf}}
```

- Write each group of `n` pages to a separate output file with a given filename pattern:

```
qpdf --split-pages={{n}} {{path/to/input.pdf}} {{path/to/
out_%d.pdf}}
```

- Rotate certain pages of a PDF with a given angle:

```
qpdf --rotate={{90:2,4,6}} --rotate={{180:7-8}} {{path/to/
input.pdf}} {{path/to/output.pdf}}
```

- Remove the password from a password-protected file:

```
qpdf --password={{password}} --decrypt {{path/to/input.pdf}}
{{path/to/output.pdf}}
```

qr

Generate QR codes in the terminal with ANSI VT-100 escape codes.

More information: https://manned.org/qr.

- Generate a QR code:

```
qr "{{data}}"
```

- Specify the error correction level (defaults to `M`):

```
qr --error-correction {{L|M|Q|H}} "{{data}}"
```

- Generate a QR code from the output of another command:

```
{{command}} | qr
```

- Save the QR code as a PNG image:

```
qr "{{data}}" > {{path/to/file.png}}
```

qrencode

QR Code generator. Supports PNG and EPS.

More information: https://manned.org/qrencode.

- Convert a string to a QR code and save to an output file:

```
qrencode {{[-o|--output]}} {{path/to/output_file.png}} {{string}}
```

- Convert an input file to a QR code and save to an output file:

```
qrencode {{[-o|--output]}} {{path/to/output_file.png}} {{[-r|--read-from]}} {{path/to/input_file}}
```

- Convert a string to a QR code and print it in terminal:

```
qrencode {{[-t|--type]}} ansiutf8 {{string}}
```

- Convert input from pipe to a QR code and print it in terminal:

```
echo {{string}} | qrencode {{[-t|--type]}} ansiutf8
```

qrttoppm

Convert a QRT ray tracer file to a PPM image.

More information: https://netpbm.sourceforge.net/doc/qrttoppm.html.

- Convert a QRT file to a PPM image:

```
qrttoppm {{path/to/file.qrt}} > {{path/to/image.ppm}}
```

qtcreator

Cross-platform IDE for Qt applications.

More information: https://doc.qt.io/qtcreator/creator-cli.html.

- Start Qt Creator:

```
qtcreator
```

- Start Qt Creator and restore the last session:

```
qtcreator -lastsession
```

- Start Qt Creator but don't load the specified plugin:

```
qtcreator -noload {{plugin}}
```

- Start Qt Creator but don't load any plugins:

```
qtcreator -noload {{all}}
```

- Start Qt Creator in presentation mode with pop-ups for keyboard shortcuts:

```
qtcreator -presentationMode
```

- Start Qt Creator and show the diff from a specific commit:

```
qtcreator -git-show {{commit}}
```

quarkus

Create Quarkus projects, manage extensions and perform essential build and development tasks.

More information: https://quarkus.io/guides/cli-tooling.

- Create a new application project in a new directory:

  ```
  quarkus create app {{project_name}}
  ```

- Run the current project in live coding mode:

  ```
  quarkus dev
  ```

- Run the application:

  ```
  quarkus run
  ```

- Run the current project in continuous testing mode:

  ```
  quarkus test
  ```

- Add one or more extensions to the current project:

  ```
  quarkus extension add {{extension_name1 extension_name2 ...}}
  ```

- Build a container image using Docker:

  ```
  quarkus image build docker
  ```

- Deploy the application to Kubernetes:

  ```
  quarkus deploy kubernetes
  ```

- Update project:

  ```
  quarkus update
  ```

quarto

An open-source scientific and technical publishing system built on Pandoc.

More information: https://quarto.org/.

- Create a new project:

```
quarto create-project {{path/to/destination_directory}} --
type {{book|default|website}}
```

- Create a new blog website:

```
quarto create-project {{path/to/destination_directory}} --
type {{website}} --template {{blog}}
```

- Render input file(s) to different formats:

```
quarto render {{path/to/file.[qmd|rmd|ipynb]}} --to {{html|
pdf|docx}}
```

- Render and preview a document or a website:

```
quarto preview {{path/to/destination_directory|path/to/file}}
```

- Publish a document or project to Quarto Pub, Github Pages, RStudio Connect or Netlify:

```
quarto publish {{quarto-pub|gh-pages|connect|netlify}}
```

quilt

Manage a series of patches.

More information: https://manned.org/quilt.

- Import an existing patch from a file:

```
quilt import {{path/to/file.patch}}
```

- Create a new patch:

```
quilt new {{filename.patch}}
```

- Add a file to the current patch:

```
quilt add {{path/to/file}}
```

- After editing the file, refresh the current patch with the changes:

```
quilt refresh
```

- Apply all the patches in the series file:

```
quilt push -a
```

- Remove all applied patches:

```
quilt pop -a
```

quota

Display users' disk space usage and allocated limits.

More information: https://manned.org/quota.

- Show disk quotas in human-readable units for the current user:

```
quota {{[-s|--human-readable]}}
```

- Verbose output (also display quotas on filesystems where no storage is allocated):

```
quota {{[-v|--verbose]}}
```

- Quiet output (only display quotas on filesystems where usage is over quota):

```
quota {{[-q|--quiet]}}
```

- Print quotas for the groups of which the current user is a member:

```
quota {{[-g|--group]}}
```

- Show disk quotas for another user:

```
sudo quota {{[-u|--user]}} {{username}}
```

qutebrowser

A keyboard-driven, vim-like browser based on PyQt5.

More information: https://qutebrowser.org/doc/qutebrowser.1.html.

- Open qutebrowser with a specified storage directory:

```
qutebrowser --basedir {{path/to/directory}}
```

- Open a qutebrowser instance with temporary settings:

```
qutebrowser --set {{content.geolocation}} {{true|false}}
```

- Restore a named session of a qutebrowser instance:

```
qutebrowser --restore {{session_name}}
```

- Launch qutebrowser, opening all URLs using the specified method:

```
qutebrowser --target {{auto|tab|tab-bg|tab-silent|tab-bg-silent|window|private-window}}
```

- Open qutebrowser with a temporary base directory and print logs to `stdout` as JSON:

```
qutebrowser --temp-basedir --json-logging
```

qwen

Launch an interactive prompt with Qwen3-Coder.

See also: **`gemini`**.

More information: https://github.com/QwenLM/qwen-code.

- Start a REPL session to chat interactively:

```
qwen
```

- Send the output of another command to Qwen and exit immediately:

```
{{echo "Summarize the history of Rome"}} | qwen {{[-p|--prompt]}}
```

- Override the default model (default: qwen3-coder-max):

```
qwen {{[-m|--model]}} {{model_name}}
```

- Run inside a sandbox container:

```
qwen {{[-s|--sandbox]}}
```

- Execute a prompt then stay in interactive mode:

```
qwen {{[-i|--prompt-interactive]}} "{{Give me an example of recursion in Python}}"
```

- Include all files in context:

```
qwen {{[-a|--all-files]}}
```

- Show memory usage in status bar:

```
qwen --show-memory-usage
```

R

R language interpreter.

More information: https://www.r-project.org.

- Start a REPL (interactive shell):

```
R
```

- Start R in vanilla mode (i.e. a blank session that doesn't save the workspace at the end):

```
R {{[-v|--vanilla]}}
```

- Execute a file:

```
R {{[-f|--file]}} {{path/to/file.R}}
```

- Execute an R expression and then exit:

```
R -e {{expr}}
```

- Run R with a debugger:

```
R {{[-d|--debugger]}} {{debugger}}
```

- Check R packages from package sources:

```
R CMD check {{path/to/package_source}}
```

- Display version:

```
R --version
```

r2

This command is an alias of **radare2**.

- View documentation for the original command:

```
tldr radare2
```

r2e

Forwards RSS feeds to an email address.

Requires a configured **`sendmail`** or smtp setup.

More information: https://manned.org/r2e.

- Create a new feed database that sends email to an email address:

```
r2e new {{email_address}}
```

- Subscribe to a feed:

```
r2e add {{feed_name}} {{feed_URI}}
```

- Send new stories to an email address:

```
r2e run
```

- List all feeds:

```
r2e list
```

- Delete a feed at a specified index:

```
r2e delete {{index}}
```

- Display help:

```
r2e {{[-h|--help]}}
```

rabbitmq-diagnostics

Diagnose, monitor, and run health checks on RabbitMQ nodes.

Many subcommands are delegated to **`rabbitmqctl`**.

More information: https://www.rabbitmq.com/docs/man/rabbitmq-diagnostics.8.html.

- List resource alarms:

  ```
  rabbitmq-diagnostics alarms
  ```

- List node certificates:

  ```
  rabbitmq-diagnostics certificates
  ```

- Check if RabbitMQ is running on the specified node:

  ```
  rabbitmq-diagnostics check_running --node {{node}}
  ```

- Run peer discovery:

  ```
  rabbitmq-diagnostics discover_peers
  ```

- List listeners (bound sockets):

  ```
  rabbitmq-diagnostics listeners
  ```

- Print the last n log lines on the specified node:

  ```
  rabbitmq-diagnostics log_tail --number {{n}} --node {{node}}
  ```

rabbitmq-plugins

Manage RabbitMQ plugins.

More information: https://www.rabbitmq.com/docs/man/rabbitmq-plugins.8.html.

- List plugins:

```
rabbitmq-plugins list
```

- Enable a plugin:

```
rabbitmq-plugins enable {{plugin}}
```

- Disable a plugin:

```
rabbitmq-plugins disable {{plugin}}
```

- Enable a plugin and disable the rest:

```
rabbitmq-plugins set {{plugin}}
```

rabbitmq-queues

Manage RabbitMQ queues.

More information: https://www.rabbitmq.com/docs/man/rabbitmq-queues.8.html.

- Grow quorum queue clusters by adding a member on the specified node:

```
rabbitmq-queues grow {{node}} {{all|even}}
```

- Rebalance leaders of replicated queues across nodes:

```
rabbitmq-queues rebalance {{all|quorum|stream}}
```

- Shrink quorum queue clusters by removing any members on the specified node:

```
rabbitmq-queues shrink {{node}}
```

- Add a quorum queue member on the specified node:

```
rabbitmq-queues add_member {{queue}} {{node}}
```

- Delete a quorum queue member on the specified node:

```
rabbitmq-queues delete_member {{queue}} {{node}}
```

- Display quorum status of a quorum queue:

```
rabbitmq-queues quorum_status {{queue}}
```

rabbitmq-streams

Manage RabbitMQ streams.

More information: https://www.rabbitmq.com/docs/man/rabbitmq-streams.8.html.

- Add a stream replica on the specified node:

```
rabbitmq-streams add_replica {{stream}} {{node}}
```

- Delete a stream replica on the specified node:

```
rabbitmq-streams delete_replica {{stream}} {{node}}
```

- Display the status of a stream:

```
rabbitmq-streams stream_status {{stream}}
```

- Restart a stream:

```
rabbitmq-streams restart_stream {{stream}}
```

- List stream connections:

```
rabbitmq-streams list_stream_connections
```

- List all stream consumers:

```
rabbitmq-streams list_stream_consumers
```

- List all stream publishers:

```
rabbitmq-streams list_stream_publishers
```

- List tracking information for a stream:

```
rabbitmq-streams list_stream_tracking {{stream}}
```

rabbitmq-upgrade

Manage the upgrade of a RabbitMQ node.

More information: https://www.rabbitmq.com/docs/man/rabbitmq-upgrade.8.html.

- Wait for all quorum queues to have an above minimum online quorum:

```
rabbitmq-upgrade await_online_quorum_plus_one
```

- Put the node in maintenance mode:

```
rabbitmq-upgrade drain
```

- Put the node out of maintenance and into regular operating mode:

```
rabbitmq-upgrade revive
```

- Run post-upgrade tasks:

```
rabbitmq-upgrade post_upgrade
```

- Display help:

```
rabbitmq-upgrade help
```

rabbitmqctl-cluster

Manage RabbitMQ nodes in a cluster.

More information: https://www.rabbitmq.com/rabbitmqctl.8.html.

- Display the status of the cluster:

```
rabbitmqctl cluster_status
```

- Display the status of the current node:

```
rabbitmqctl status
```

- Start the RabbitMQ application on a specific node:

```
rabbitmqctl {{[-n|--node]}} {{nodename}} start_app
```

- Stop the RabbitMQ application on a specific node:

```
rabbitmqctl {{[-n|--node]}} {{nodename}} stop_app
```

- Stop a specific RabbitMQ node:

```
rabbitmqctl {{[-n|--node]}} {{nodename}} stop
```

- Reset a specific RabbitMQ node to a clean state:

```
rabbitmqctl {{[-n|--node]}} {{nodename}} reset
```

- Make the current node join an existing cluster:

```
rabbitmqctl join_cluster {{nodename}}
```

rabbitmqctl-users

Manage RabbitMQ users, their permissions, and tags.

More information: https://www.rabbitmq.com/management.html.

- List all users:

```
rabbitmqctl list_users
```

- Add a new user with a password:

```
rabbitmqctl add_user {{username}} {{password}}
```

- Delete an existing user:

```
rabbitmqctl delete_user {{username}}
```

- Change the password for a user:

```
rabbitmqctl change_password {{username}} {{new_password}}
```

- Set permissions for a user on a specific virtual host:

```
rabbitmqctl set_permissions {{[-p|--vhost]}} {{vhost}}
{{username}} {{read}} {{write}} {{configure}}
```

- Clear all permissions for a user on a specific virtual host:

```
rabbitmqctl clear_permissions {{[-p|--vhost]}} {{vhost}}
{{username}}
```

- Assign one or more tags (e.g., administrator) to a user:

```
rabbitmqctl set_user_tags {{username}} {{tag1}} [{{tag2}}]
```

rabbitmqctl-vhosts

Manage virtual hosts (vhosts) in RabbitMQ.

Vhosts are used to separate multiple logical brokers on the same RabbitMQ server.

More information: https://www.rabbitmq.com/vhosts.html.

- List all virtual hosts:

```
rabbitmqctl list_vhosts
```

- Add a new virtual host:

```
rabbitmqctl add_vhost {{vhost_name}}
```

- Delete a virtual host:

```
rabbitmqctl delete_vhost {{vhost_name}}
```

- Set permissions for a user on a specific virtual host:

```
rabbitmqctl set_permissions {{[-p|--vhost]}} {{vhost_name}}
{{username}} {{read}} {{write}} {{configure}}
```

- Clear permissions for a user on a specific virtual host:

```
rabbitmqctl clear_permissions {{[-p|--vhost]}} {{vhost_name}}
{{username}}
```

rabbitmqctl

Manage, configure, and inspect RabbitMQ servers.

More information: https://www.rabbitmq.com/rabbitmqctl.8.html.

- Display the status of the current node:

```
rabbitmqctl status
```

- View documentation for RabbitMQ cluster commands:

```
tldr rabbitmqctl cluster
```

- View documentation for RabbitMQ user commands:

```
tldr rabbitmqctl users
```

- View documentation for RabbitMQ vhosts commands:

```
tldr rabbitmqctl vhosts
```

- Learn more about a specific command:

```
rabbitmqctl help {{command}}
```

rabin2

Get information about binary files (ELF, PE, Java CLASS, Mach-O) - symbols, sections, linked libraries, etc.

Comes bundled with **`radare2`**.

More information: https://manned.org/rabin2.

- Display general information about a binary (architecture, type, endianness):

```
rabin2 -I {{path/to/binary}}
```

- Display linked libraries:

```
rabin2 -l {{path/to/binary}}
```

- Display symbols imported from libraries:

```
rabin2 -i {{path/to/binary}}
```

- Display strings contained in the binary:

```
rabin2 -z {{path/to/binary}}
```

- Display the output in JSON:

```
rabin2 -j -I {{path/to/binary}}
```

racket

Racket language interpreter.

More information: https://docs.racket-lang.org/reference/running-sa.html#%28part._mz-cmdline%29.

- Start a REPL (interactive shell):

```
racket
```

- Execute a Racket script:

```
racket {{path/to/script.rkt}}
```

- Execute a Racket expression:

```
racket {{[-e|--eval]}} "{{expression}}"
```

- Run module as a script (terminates option list):

```
racket {{[-l|--lib]}} {{module_name}} {{[-m|--main]}} {{arguments}}
```

- Start a REPL (interactive shell) for the `typed/racket` hashlang:

```
racket -I typed/racket
```

raco

Racket tools.

More information: https://docs.racket-lang.org/raco/.

- Install a package, automatically installing dependencies:

```
raco pkg install --auto {{package_source}}
```

- Install the current directory as a package:

```
raco pkg install
```

- Build (or rebuild) bytecode, documentation, executables, and metadata indexes for collections:

```
raco setup {{collection1 collection2 ...}}
```

- Run tests in files:

```
raco test {{path/to/tests1.rkt path/to/tests2.rkt ...}}
```

- Search local documentation:

```
raco docs {{search_terms}}
```

- Display help:

```
raco help
```

radare2

A set of reverse engineering tools.

More information: https://book.rada.re/first_steps/commandline_flags.html.

- Open a file in write mode without parsing the file format headers:

```
radare2 -nw {{path/to/binary}}
```

- Debug a program:

```
radare2 -d {{path/to/binary}}
```

- Run a script before entering the interactive CLI:

```
radare2 -i {{path/to/script.r2}} {{path/to/binary}}
```

- Display help text for any command in the interactive CLI:

```
{{radare2_command}}?
```

- Run a shell command from the interactive CLI:

```
!{{shell_command}}
```

- Dump raw bytes of current block to a file:

```
> pr > {{path/to/file.bin}}
```

rage

A simple, secure and modern file encryption tool (and Rust library) with small explicit keys, no configuration options, and UNIX-style composability.

Rust implementation of **age**.

More information: https://github.com/str4d/rage.

- Encrypt a file for `user` and save it to `message.age`:

```
echo "{{Your secret message}}" | rage --encrypt --recipient {{user}} --output {{path/to/message.age}}
```

- Decrypt a file with `identity_file` and save it to `message`:

```
rage --decrypt --identity {{path/to/identity_file}} --output {{message}}
```

rails console

Interact with a Rails application.

More information: https://guides.rubyonrails.org/command_line.html#bin-rails-console.

- Start the Rails console:

```
rails console
```

- Start the Rails console and roll back all data modifications on exit:

```
rails console {{[-s|--sandbox]}}
```

- Start the Rails console on a specified environment:

```
rails console {{[-e|--environment]}} {{dev|test|
production|...}}
```

- Display help:

```
rails console {{[-h|--help]}}
```

rails db

Various database-related subcommands for Ruby on Rails.

More information: https://guides.rubyonrails.org/active_record_migrations.html.

- Create databases, load the schema, and initialize with seed data:

```
rails db:setup
```

- Access the database console:

```
rails db
```

- Create the databases defined in the current environment:

```
rails db:create
```

- Destroy the databases defined in the current environment:

```
rails db:drop
```

- Run pending migrations:

```
rails db:migrate
```

- View the status of each migration file:

```
rails db:migrate:status
```

- Rollback the last migration:

```
rails db:rollback
```

- Fill the current database with data defined in `db/seeds.rb`:

```
rails db:seed
```

rails destroy

Destroy Rails resources.

More information: https://guides.rubyonrails.org/command_line.html#bin-rails-destroy.

- List all available generators to destroy:

```
rails destroy
```

- Destroy a model named Post:

```
rails destroy model {{Post}}
```

- Destroy a controller named Posts:

```
rails destroy controller {{Posts}}
```

- Destroy a migration that creates Posts:

```
rails destroy migration {{CreatePosts}}
```

- Destroy a scaffold for a model named Post:

```
rails destroy scaffold {{Post}}
```

rails generate

Generate new Rails templates in an existing project.

More information: https://guides.rubyonrails.org/command_line.html#bin-rails-generate.

- List all available generators:

```
rails generate
```

- Generate a new model named Post with attributes title and body:

```
rails generate model {{Post}} {{title:string}} {{body:text}}
```

- Generate a new controller named Posts with actions index, show, new and create:

```
rails generate controller {{Posts}} {{index}} {{show}} {{new}} {{create}}
```

- Generate a new migration that adds a category attribute to an existing model called Post:

```
rails generate migration {{AddCategoryToPost}} {{category:string}}
```

- Generate a scaffold for a model named Post, predefining the attributes title and body:

```
rails generate scaffold {{Post}} {{title:string}} {{body:text}}
```

rails new

Create a new Rails application.

More information: https://guides.rubyonrails.org/command_line.html#rails-new.

- Create a Rails app named `blog` in the current directory:

```
rails new blog
```

- Create a Rails app with API-only configuration:

```
rails new {{app_name}} --api
```

- Create a Rails app with `postgresql` as the database:

```
rails new {{app_name}} {{[-d|--database]}} postgresql
```

- Create a Rails app without generating JavaScript files:

```
rails new {{app_name}} {{[-J|--skip-javascript]}}
```

- Display help:

```
rails new {{[-h|--help]}}
```

rails routes

List routes in a Rails application.

More information: https://guides.rubyonrails.org/routing.html.

- List all routes:

```
rails routes
```

- List all routes in an expanded format:

```
rails routes {{[-E|--expanded]}}
```

- List routes partially matching URL helper method name, HTTP verb, or URL path:

```
rails routes {{[-g|--grep]}} {{posts_path|GET|/posts}}
```

- List routes that map to a specified controller:

```
rails routes {{[-c|--controller]}} {{posts|Posts|
Blogs::PostsController}}
```

rails server

Serve the Rails app in the current directory using the Puma web server, which comes bundled with Rails.

More information: https://guides.rubyonrails.org/command_line.html#bin-rails-server.

- Run the web server:

```
rails server
```

- Run the web server on a specified port:

```
rails server {{[-p|--port]}} {{port_number}}
```

- Run the web server on a specified IP address:

```
rails server {{[-b|--binding]}} {{ip_address}}
```

- Run the web server on a specified environment:

```
rails server {{[-e|--environment]}} {{environment}}
```

- Display help:

```
rails server {{[-h|--help]}}
```

rails

A server-side MVC framework written in Ruby.

Some subcommands such as **`generate`** have their own usage documentation.

More information: https://guides.rubyonrails.org/command_line.html.

- Create a new rails project:

```
rails new "{{project_name}}"
```

- Generate a scaffold for a model named Post, predefining the attributes title and body:

```
rails generate scaffold Post title:string body:text
```

- Run migrations:

```
rails db:migrate
```

- List all routes:

```
rails routes
```

- Start local server for current project on port 3000:

```
rails server
```

- Start local server for current project on a specified port:

```
rails server {{[-p|--port]}} "{{port}}"
```

- Open console to interact with application from command-line:

```
rails console
```

- Display version:

```
rails {{[-v|--version]}}
```

railway

Connect code to a Railway project.

More information: https://docs.railway.com/reference/cli-api.

- Login to a Railway account:

  ```
  railway login
  ```

- Link to an existing Project under a Railway account or team:

  ```
  railway link {{projectId}}
  ```

- Create a new project:

  ```
  railway init
  ```

- Run a local command using variables from the active environment:

  ```
  railway run {{cmd}}
  ```

- Deploy the linked project directory (if running from a subdirectory, the project root is still deployed):

  ```
  railway up
  ```

- Open an interactive shell to a database:

  ```
  railway connect
  ```

rainbowstream

Terminal-based Twitter client supporting realtime tweetstream, trends, sending, search, favorites and user management.

Online help with **<h>**, **<ArrowUp>** and **<ArrowDown>** for history, **<Tab>** to auto-complete and 2-**<Tab>** for suggestion.

More information: https://github.com/orakaro/rainbowstream.

- Open RainbowStream:

  ```
  rainbowstream
  ```

- Show your timeline (optional number of tweets to display, default is 5):

  ```
  home [{{num_of_last_tweets}}]
  ```

- Show profile of a given user:

  ```
  whois @{{user}}
  ```

- Tweet the message as-is:

  ```
  t {{message}}
  ```

- Retweet the tweet with given ID (ID is beside the time):

  ```
  rt {{tweet_id}}
  ```

- Favorite the tweet with given ID:

  ```
  fav {{tweet_id}}
  ```

- Perform a search for a given word (with or without hashtag):

  ```
  s {{word}}
  ```

rake

A Make-like program for Ruby.

Tasks for **rake** are specified in a Rakefile.

More information: https://ruby.github.io/rake.

- Run the `default` Rakefile task:

  ```
  rake
  ```

- Run a specific task:

  ```
  rake {{task}}
  ```

- Execute `n` jobs at a time in parallel (number of CPU cores + 4 by default):

  ```
  rake --jobs {{n}}
  ```

- Use a specific Rakefile:

  ```
  rake --rakefile {{path/to/Rakefile}}
  ```

- Execute `rake` from another directory:

  ```
  rake --directory {{path/to/directory}}
  ```

ranger

Console file manager with VI key bindings.

See also: **`clifm`**, **`vifm`**, **`mc`**, **`dolphin`**.

More information: https://github.com/ranger/ranger.

- Launch ranger:

```
ranger
```

- Show only directories:

```
ranger --show-only-dirs
```

- Change the configuration directory:

```
ranger {{[-r|--confdir]}} {{path/to/directory}}
```

- Change the data directory:

```
ranger --datadir {{path/to/directory}}
```

- Print CPU usage statistics on exit:

```
ranger --profile
```

rapper

The Raptor RDF parsing utility.

Part of the Raptor RDF Syntax Library.

More information: https://librdf.org/raptor/rapper.html.

- Convert an RDF/XML document to Turtle:

```
rapper {{[-i|--input]}} rdfxml {{[-o|--output]}} turtle
{{path/to/file}}
```

- Count the number of triples in a Turtle file:

```
rapper {{[-i|--input]}} turtle {{[-c|--count]}} {{path/to/
file}}
```

rar

The RAR archiver. Supports multi-volume archives that can be optionally self-extracting.

More information: https://manned.org/rar.

- Archive 1 or more files:

```
rar a {{path/to/archive_name.rar}} {{path/to/file1 path/to/file2 path/to/file3 ...}}
```

- Archive a directory:

```
rar a {{path/to/archive_name.rar}} {{path/to/directory}}
```

- Split the archive into parts of equal size (50M):

```
rar a -v{{50M}} -R {{path/to/archive_name.rar}} {{path/to/file_or_directory}}
```

- Password protect the resulting archive:

```
rar a -p{{password}} {{path/to/archive_name.rar}} {{path/to/file_or_directory}}
```

- Encrypt file data and headers with password:

```
rar a -hp{{password}} {{path/to/archive_name.rar}} {{path/to/file_or_directory}}
```

- Use a specific compression level (0-5):

```
rar a -m{{compression_level}} {{path/to/archive_name.rar}} {{path/to/file_or_directory}}
```

rarcrack

Password cracker for RAR, Zip and 7z archives.

- Brute force the password for an archive (tries to guess the archive type):

```
rarcrack {{path/to/file.zip}}
```

- Specify the archive type:

```
rarcrack --type {{rar|zip|7z}} {{path/to/file.zip}}
```

- Use multiple threads:

```
rarcrack --threads {{6}} {{path/to/file.zip}}
```

rasttopnm

Convert a Sun rasterfile to a PNM file.

More information: https://netpbm.sourceforge.net/doc/rasttopnm.html.

- Convert a RAST image to a PNM file:

```
rasttopnm {{path/to/input.rast}} > {{path/to/output.pnm}}
```

- Use the color map indices in the raster if they are color values:

```
rasttopnm {{[-i|-index]}} {{path/to/input.rast}} > {{path/to/
output.pnm}}
```

rawtopgm

Convert a raw greyscale image to a PGM image.

More information: https://netpbm.sourceforge.net/doc/rawtopgm.html.

- Convert a raw greyscale image to a PGM image:

```
rawtopgm {{width}} {{height}} {{path/to/image.raw}} > {{path/to/output.pgm}}
```

- Convert a raw greyscale image to a PGM image, assume the image to be a square:

```
rawtopgm {{path/to/image.raw}} > {{path/to/output.pgm}}
```

- Convert a raw greyscale image in which the pixels come bottom-first instead of top-first to a PGM image:

```
rawtopgm {{width}} {{height}} {{[-bt|-bottomfirst]}} {{path/to/image.raw}} > {{path/to/output.pgm}}
```

- Ignore the first n bytes of the specified file:

```
rawtopgm {{width}} {{height}} {{[-h|-headerskip]}} {{n}} {{path/to/image.raw}} > {{path/to/output.pgm}}
```

- Ignore the last m bytes of each row in the specified file:

```
rawtopgm {{width}} {{height}} {{[-r|-rowskip]}} {{m}} {{path/to/image.raw}} > {{path/to/output.pgm}}
```

- Specify the maxval for the grey values in the input to be equal to n:

```
rawtopgm {{width}} {{height}} {{[-m|-maxval]}} {{n}} {{path/to/image.raw}} > {{path/to/output.pgm}}
```

- Specify the number of bytes that represent each sample in the input and that the byte-sequence is to be interpreted as little-endian:

```
rawtopgm {{width}} {{height}} -bpp {{1|2}} {{[-l|-littleendian]}} {{path/to/image.raw}} > {{path/to/output.pgm}}
```

rawtoppm

Convert a raw RGB stream to a PPM image.

More information: https://netpbm.sourceforge.net/doc/rawtoppm.html.

- Convert a raw RGB stream to a PPM image:

```
rawtoppm {{width}} {{height}} {{path/to/image.raw}} > {{path/to/output.ppm}}
```

- Convert a raw RGB stream in which the pixels come bottom-first instead of top-first to a PPM image:

```
rawtoppm {{width}} {{height}} {{path/to/image.raw}} | pamflip {{[-tb|-topbottom]}} > {{path/to/output.ppm}}
```

- Ignore the first n bytes of the specified file:

```
rawtoppm {{width}} {{height}} {{[-h|-headerskip]}} {{n}} {{path/to/image.raw}} > {{path/to/output.ppm}}
```

- Ignore the last m bytes of each row in the specified file:

```
rawtoppm {{width}} {{height}} {{[-ro|-rowskip]}} {{m}} {{path/to/image.raw}} > {{path/to/output.ppm}}
```

- Specify the order of color components for each pixel:

```
rawtoppm {{width}} {{height}} -{{rgb|rbg|grb|gbr|brg|bgr}} {{path/to/image.raw}} > {{path/to/output.ppm}}
```

rbac-lookup

Find roles and cluster roles attached to any user, service account or group name in your Kubernetes cluster.

More information: https://github.com/reactiveops/rbac-lookup.

- View all RBAC bindings:

```
rbac-lookup
```

- View RBAC bindings that match a given expression:

```
rbac-lookup {{search_term}}
```

- View all RBAC bindings along with the source role binding:

```
rbac-lookup {{[-o|--output]}} wide
```

- View all RBAC bindings filtered by subject:

```
rbac-lookup {{[-k|--kind]}} {{user|group|serviceaccount}}
```

- View all RBAC bindings along with IAM roles (if you are using GKE):

```
rbac-lookup --gke
```

rbash

Restricted Bash shell, equivalent to **`bash --restricted`**.

Does not permit changing the working directory, redirecting command output, or modifying environment variables, among other things.

See also: **`histexpand`** for history expansion.

More information: https://www.gnu.org/software/bash/manual/bash.html#The-Restricted-Shell.

- Start an interactive shell session:

```
rbash
```

- Execute a command and then exit:

```
rbash -c "{{command}}"
```

- Execute a script:

```
rbash {{path/to/script.sh}}
```

- Execute a script, printing each command before executing it:

```
rbash -x {{path/to/script.sh}}
```

- Execute commands from a script, stopping at the first error:

```
rbash -e {{path/to/script.sh}}
```

- Read and execute commands from `stdin`:

```
rbash -s
```

rbenv

Easily install Ruby versions and manage application environments.

See also: **`asdf`**.

More information: https://github.com/rbenv/rbenv.

- Install a Ruby version:

```
rbenv install {{version}}
```

- Display a list of the latest stable versions for each Ruby:

```
rbenv install --list
```

- Display a list of installed Ruby versions:

```
rbenv versions
```

- Use a specific Ruby version across the whole system:

```
rbenv global {{version}}
```

- Use a specific Ruby version for an application/project directory:

```
rbenv local {{version}}
```

- Display the currently selected Ruby version:

```
rbenv version
```

- Uninstall a Ruby version:

```
rbenv uninstall {{version}}
```

- Display all Ruby versions that contain the specified executable:

```
rbenv whence {{executable}}
```

rbt

RBTools, a set of tools for working with Review Board and RBCommons.

More information: https://www.reviewboard.org/docs/rbtools/dev/.

- Post changes to Review Board:

  ```
  rbt post {{change_number}}
  ```

- Display the diff that will be sent to Review Board:

  ```
  rbt diff
  ```

- Land a change in a local branch or on a review request:

  ```
  rbt land {{branch_name}}
  ```

- Patch your tree with a change on a review request:

  ```
  rbt patch {{review_request_id}}
  ```

- Set up RBTool to talk to a repository:

  ```
  rbt setup-repo
  ```

rbw

Unofficial Bitwarden-compatible password manager.

More information: https://github.com/doy/rbw#configuration.

- Log in to the vault:

```
rbw login
```

- Unlock the vault:

```
rbw unlock
```

- List all items in the vault:

```
rbw list
```

- Get a password for an entry:

```
rbw get "{{entry_name}}"
```

- Get a username for an entry:

```
rbw get {{[-f|--field]}} username "{{entry_name}}"
```

- Copy a password to the clipboard:

```
rbw get {{[-c|--clipboard]}} "{{entry_name}}"
```

- Generate a new password with the specified number of characters:

```
rbw generate {{password_length}}
```

- Lock the vault:

```
rbw lock
```

rc

A modern simplistic port listener & reverse shell.

Similar to **nc**.

More information: https://github.com/robiot/rustcat/wiki/Basic-Usage.

- Start listening on a specific port:

```
rc -lp {{port}}
```

- Start a reverse shell:

```
rc {{host}} {{port}} -r {{shell}}
```

rcat

This command is an alias of **rc**.

- View documentation for the original command:

```
tldr rc
```

rclone

Copy, synchronize or move files and directories to and from many cloud services.

More information: https://rclone.org.

- Launch an interactive menu to setup rclone:

```
rclone config
```

- List contents of a directory on an rclone remote:

```
rclone lsf {{remote_name}}:{{path/to/directory}}
```

- Copy a file or directory from the local machine to the remote destination:

```
rclone copy {{path/to/source_file_or_directory}}
{{remote_name}}:{{path/to/directory}}
```

- Copy files changed within the past 24 hours to a remote from the local machine, asking the user to confirm each file:

```
rclone copy {{[-i|--interactive]}} --max-age 24h
{{remote_name}}:{{path/to/directory}} {{path/to/
local_directory}}
```

- Mirror a specific file or directory (Note: Unlike copy, sync removes files from the remote if it does not exist locally):

```
rclone sync {{path/to/file_or_directory}} {{remote_name}}:
{{path/to/directory}}
```

- Delete a remote file or directory (Note: `--dry-run` means test, remove it from the command to actually delete):

```
rclone {{[-n|--dry-run]}} delete {{remote_name}}:{{path/to/
file_or_directory}}
```

- Mount rclone remote (experimental):

```
rclone mount {{remote_name}}:{{path/to/directory}} {{path/to/
mount_point}}
```

- Unmount rclone remote if `<Ctrl c>` fails (experimental):

```
fusermount {{[-u|--update]}} {{path/to/mount_point}}
```

rdfind

Find files with duplicate content and get rid of them.

More information: https://rdfind.pauldreik.se/rdfind.1.html.

- Identify all duplicates in a given directory and output a summary:

```
rdfind -dryrun true {{path/to/directory}}
```

- Replace all duplicates with hardlinks:

```
rdfind -makehardlinks true {{path/to/directory}}
```

- Replace all duplicates with symlinks/soft links:

```
rdfind -makesymlinks true {{path/to/directory}}
```

- Delete all duplicates and do not ignore empty files:

```
rdfind -deleteduplicates true -ignoreempty false {{path/to/
directory}}
```

rdiff-backup

Local/remote mirror and incremental backup tool.

More information: https://rdiff-backup.net/rdiff-backup.1.html.

- Back up `path/to/source` to `path/to/backup`:

```
rdiff-backup backup {{path/to/source}} {{path/to/backup}}
```

- List incremental backups in repository (location path, local or remote):

```
rdiff-backup list increments {{repository}}
```

- Restore from most recent backup:

```
rdiff-backup restore {{path/to/backup}} {{path/to/
destination}}
```

- Restore backed up files as they were 3 days ago:

```
rdiff-backup restore --at 3D {{path/to/backup}} {{path/to/
destination}}
```

react-native start

Start the React Native server.

More information: https://github.com/facebook/react-native/tree/main/packages/community-cli-plugin#start.

- Start the server that communicates with connected devices:

  ```
  react-native start
  ```

- Start the metro bundler with a clean cache:

  ```
  react-native start --reset-cache
  ```

- Start the server in a custom port (defaults to 8081):

  ```
  react-native start --port {{3000}}
  ```

- Start the server in verbose mode:

  ```
  react-native start --verbose
  ```

- Specify the maximum number of workers for transforming files (default is the number of CPU cores):

  ```
  react-native start --max-workers {{count}}
  ```

- Disable interactive mode:

  ```
  react-native start --no-interactive
  ```

react-native

A framework for building native apps with React.

More information: https://reactnative.dev.

- Initialize a new React Native project in a directory of the same name:

```
react-native init {{project_name}}
```

- Start the metro bundler:

```
react-native start
```

- Start the metro bundler with a clean cache:

```
react-native start --reset-cache
```

- Build the current application and start it on a connected Android device or emulator:

```
react-native run-android
```

- Build the current application and start it on an iOS simulator:

```
react-native run-ios
```

- Build the current application in `release` mode and start it on a connected Android device or emulator:

```
react-native run-android --variant={{release}}
```

- Start `logkitty` and print logs to `stdout`:

```
react-native log-android
```

- Start `tail system.log` for an iOS simulator and print logs to `stdout`:

```
react-native log-ios
```

read

Shell builtin for retrieving data from **`stdin`**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-read.

- Store data that you type from the keyboard:

```
read {{variable}}
```

- Store each of the next lines you enter as values of an array:

```
read -a {{array}}
```

- Specify the number of maximum characters to be read:

```
read -n {{character_count}} {{variable}}
```

- Assign multiple values to multiple variables:

```
read <<< "{{The surname is Bond}}" {{_ variable1 _ variable2}}
```

- Do not let backslash (\) act as an escape character:

```
read -r {{variable}}
```

- Display a prompt before the input:

```
read -p "{{Enter your input here: }}" {{variable}}
```

- Do not echo typed characters (silent mode):

```
read -s {{variable}}
```

- Read `stdin` and perform an action on every line:

```
while read line; do {{echo|ls|rm|...}} "$line"; done < {{/dev/stdin|path/to/file|...}}
```

readarray

Read lines from **`stdin`** into an array.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-readarray.

- Interactively input lines into an array:

```
readarray {{array_name}}
```

- Read lines from a file and insert them in an array:

```
readarray < {{path/to/file.txt}} {{array_name}}
```

- Remove trailing deliminators (newline by default):

```
readarray < {{path/to/file.txt}} -t {{array_name}}
```

- Copy at most `n` lines:

```
readarray < {{path/to/file.txt}} -n {{n}} {{array_name}}
```

- Display help:

```
help mapfile
```

readlink

Follow symlinks and get symlink information.

More information: https://www.gnu.org/software/coreutils/manual/html_node/readlink-invocation.html.

- Get the actual file to which the symlink points:

```
readlink {{path/to/file}}
```

- Get the absolute path to a file:

```
readlink {{[-f|--canonicalize]}} {{path/to/file}}
```

readonly

Set read-only shell variables.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-readonly.

- Set a read-only variable:

```
readonly {{variable_name}}={{value}}
```

- Mark a variable as read-only:

```
readonly {{existing_variable}}
```

- [p]rint the names and values of all read-only variables to `stdout`:

```
readonly -p
```

realpath

Display the resolved absolute path for a file or directory.

More information: https://www.gnu.org/software/coreutils/manual/html_node/realpath-invocation.html.

- Display the absolute path for a file or directory:

```
realpath {{path/to/file_or_directory}}
```

- Require all path components to exist:

```
realpath {{[-e|--canonicalize-existing]}} {{path/to/file_or_directory}}
```

- Resolve ".." components before symlinks:

```
realpath {{[-L|--logical]}} {{path/to/file_or_directory}}
```

- Disable symlink expansion:

```
realpath {{[-s|--no-symlinks]}} {{path/to/file_or_directory}}
```

- Suppress error messages:

```
realpath {{[-q|--quiet]}} {{path/to/file_or_directory}}
```

recon-ng

Automated reconnaissance and information gathering tool.

More information: https://github.com/lanmaster53/recon-ng/wiki.

- Start the tool:

```
recon-ng
```

- Create a workspace:

```
workspaces create {{workspace_name}}
```

- Search the marketplace for modules used to accomplish different reconnaissance tasks:

```
marketplace search
```

- Install all available modules (some may need API keys to function completely):

```
marketplace install all
```

- Load the profiler module. It is used to scan the web for profiles matching the target, scrape them, and store them:

```
modules load profiler
```

- Insert the target's username. After entering this command, enter the desired username of the search and leave the rest of the options blank:

```
db insert profiles
```

- Run the current module:

```
run
```

recsel

Print records from a recfile: a human-editable, plain text database.

More information: https://www.gnu.org/software/recutils/manual/recutils.html#Invoking-recsel.

- Extract name and version field:

```
recsel {{[-p|--print]}} name,version {{data.rec}}
```

- Use "~" to match a string with a given regex:

```
recsel {{[-e|--expression]}} "{{field_name}} ~
'{{regex}}' {{data.rec}}"
```

- Use a predicate to match a name and a version:

```
recsel {{[-e|--expression]}} "name ~ '{{regex}}' && version ~
'{{regex}}'" {{data.rec}}
```

rector

An automated tool for updating and refactoring PHP 5.3+ code.

More information: https://github.com/rectorphp/rector.

- Process a specific directory:

```
rector process {{path/to/directory}}
```

- Process a directory without applying changes (dry run):

```
rector process {{path/to/directory}} --dry-run
```

- Process a directory and apply coding standards:

```
rector process {{path/to/directory}} --with-style
```

- Display a list of available levels:

```
rector levels
```

- Process a directory with a specific level:

```
rector process {{path/to/directory}} --level {{level_name}}
```

redis-benchmark

Benchmark a Redis server.

More information: https://redis.io/docs/latest/operate/oss_and_stack/management/optimization/benchmarks/.

- Run full benchmark:

```
redis-benchmark
```

- Run benchmark on a specific Redis server:

```
redis-benchmark -h {{host}} -p {{port}} -a {{password}}
```

- Run a subset of tests with default 100000 requests:

```
redis-benchmark -h {{host}} -p {{port}} -t {{set,lpush}} -n {{100000}}
```

- Run with a specific script:

```
redis-benchmark -n {{100000}} script load "{{redis.call('set', 'foo', 'bar')}}"
```

- Run benchmark by using 100000 [r]andom keys:

```
redis-benchmark -t {{set}} -r {{100000}}
```

- Run benchmark by using a [P]ipelining of 16 commands:

```
redis-benchmark -n {{1000000}} -t {{set,get}} -P {{16}}
```

- Run benchmark [q]uietly and only show query per seconds result:

```
redis-benchmark -q
```

redis-cli

Open a connection to a Redis server.

More information: https://redis.io/topics/rediscli.

- Connect to the local server:

```
redis-cli
```

- Connect to a remote server on the default port (6379):

```
redis-cli -h {{host}}
```

- Connect to a remote server specifying a port number:

```
redis-cli -h {{host}} -p {{port}}
```

- Connect to a remote server specifying a URI:

```
redis-cli -u {{uri}}
```

- Specify a password:

```
redis-cli -a {{password}}
```

- Execute Redis command:

```
redis-cli {{redis_command}}
```

- Connect to the local cluster:

```
redis-cli -c
```

redis-server

Persistent key-value database.

More information: https://redis.io.

- Start Redis server, using the default port (6379), and write logs to `stdout`:

```
redis-server
```

- Start Redis server, using the default port, as a background process:

```
redis-server --daemonize yes
```

- Start Redis server, using the specified port, as a background process:

```
redis-server --port {{port}} --daemonize yes
```

- Start Redis server with a custom configuration file:

```
redis-server {{path/to/redis.conf}}
```

- Start Redis server with verbose logging:

```
redis-server --loglevel {{warning|notice|verbose|debug}}
```

redshift

Adjust the color temperature of a screen according to its surroundings.

Opmerking: Redshift ondersteunt Wayland niet.

More information: https://manned.org/redshift.

- Turn on Redshift with a specific [t]emperature during day (e.g., 5700K) and at night (e.g., 3600K):

```
redshift -t 5700:3600
```

- Turn on Redshift with a manually specified custom [l]ocation:

```
redshift -l {{latitude}}:{{longitude}}
```

- Turn on Redshift with a specific screen [b]rightness during the day (e.g, 70%) and at night (e.g., 40%):

```
redshift -b 0.7:0.4
```

- Turn on Redshift with custom [g]amma levels (between 0 and 1):

```
redshift -g {{red}}:{{green}}:{{blue}}
```

- [P]urge existing temperature changes and set a constant unchanging color temperature in [O]ne-shot mode:

```
redshift -PO {{temperature}}
```

reflac

Recompress FLAC files in-place while preserving metadata.

More information: https://github.com/chungy/reflac.

- Recompress a directory of FLAC files:

```
reflac {{path/to/directory}}
```

- Enable maximum compression (very slow):

```
reflac {{[-8|--best]}} {{path/to/directory}}
```

- Display filenames as they are processed:

```
reflac {{[-v|--verbose]}} {{path/to/directory}}
```

- Recurse into subdirectories:

```
reflac {{[-r|--recursive]}} {{path/to/directory}}
```

- Preserve file modification times:

```
reflac --preserve {{path/to/directory}}
```

reflex

Watch a directory and rerun a command when certain files change.

More information: https://github.com/cespare/reflex.

- Rebuild with `make` if any file changes:

```
reflex make
```

- Compile and run Go application if any `.go` file changes:

```
reflex --regex='{{\.go$}}' {{go run .}}
```

- Ignore a directory when watching for changes:

```
reflex --inverse-regex='{{^dir/}}' {{command}}
```

- Run command when reflex starts and restarts on file changes:

```
reflex --start-service=true {{command}}
```

- Substitute the filename that changed in:

```
reflex -- echo {}
```

reindexdb

Rebuild indexes in a PostgreSQL database.

More information: https://www.postgresql.org/docs/current/app-reindexdb.html.

- Reindex a specific database:

```
reindexdb {{database_name}}
```

- Reindex a specific database using connection options:

```
reindexdb {{database_name}} {{[-h|--host]}} {{hostname}} {{[-p|--port]}} {{port}} {{[-U|--username]}} {{username}}
```

- Reindex all databases:

```
reindexdb {{[-a|--all]}}
```

- Reindex a specific table in a database:

```
reindexdb {{database_name}} {{[-t|--table]}} {{table_name}}
```

- Reindex a specific index in a database:

```
reindexdb {{database_name}} {{[-i|--index]}} {{index_name}}
```

- Reindex a specific schema in a database:

```
reindexdb {{database_name}} {{[-S|--schema]}} {{schema_name}}
```

- Reindex with verbose output:

```
reindexdb {{database_name}} {{[-v|--verbose]}}
```

- Reindex a database using multiple parallel jobs:

```
reindexdb {{database_name}} {{[-j|--jobs]}} {{number_of_jobs}}
```

rekor-cli

Immutable tamper resistant ledger of metadata generated within a software projects supply chain.

More information: https://github.com/sigstore/rekor.

- Upload an artifact to Rekor:

```
rekor-cli upload --artifact {{path/to/file.ext}} --signature {{path/to/file.ext.sig}} --pki-format={{x509}} --public-key={{path/to/key.pub}}
```

- Get information regarding entries in the Transparency Log:

```
rekor-cli get --uuid={{0e81b4d9299e2609e45b5c453a4c0e7820ac74e02c4935a8b830d104632fd2c
```

- Search the Rekor index to find entries by Artifact:

```
rekor-cli search --artifact {{path/to/file.ext}}
```

- Search the Rekor index to find entries by a specific hash:

```
rekor-cli search --sha {{6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b}}
```

Remove-NodeVersion

Uninstall Node.js runtime versions for **`ps-nvm`**.

This command is part of **`ps-nvm`** and can only be run under PowerShell.

More information: https://github.com/aaronpowell/ps-nvm.

- Uninstall a given Node.js version:

```
Remove-NodeVersion {{node_version}}
```

- Uninstall multiple Node.js versions:

```
Remove-NodeVersion {{node_version1 , node_version2 , ...}}
```

- Uninstall all currently-installed versions of Node.js 20.x:

```
Get-NodeVersions -Filter ">=20.0.0 <21.0.0" | Remove-
NodeVersion
```

- Uninstall all currently-installed versions of Node.js:

```
Get-NodeVersions | Remove-NodeVersion
```

rename

Rename a file or group of files with a **regex**.

WARNING: This command will overwrite files without prompting unless the dry-run option is used.

Note: This page refers to the Perl version, also known as **file-rename**.

More information: https://manned.org/prename.

- Replace `from` with `to` in the filenames of the specified files:

```
rename 's/{{from}}/{{to}}/' {{*.txt}}
```

- Dry-run - display which changes would occur without performing them:

```
rename -n 's/{{from}}/{{to}}/' {{*.txt}}
```

- Change the extension:

```
rename 's/\.{{old}}$/\.{{new}}/' {{*.txt}}
```

- Change to lowercase (use `-f` in case-insensitive filesystems):

```
rename {{[-f|--force]}} 'y/A-Z/a-z/' {{*.txt}}
```

- Capitalize first letter of every word in the name:

```
rename {{[-f|--force]}} 's/\b(\w)/\U$1/g' {{*.txt}}
```

- Replace spaces with underscores:

```
rename 's/\s+/_/g' {{*.txt}}
```

renice

Alter the scheduling priority/niceness of running processes.

Niceness values range from -20 (most favorable to the process) to 19 (least favorable to the process).

See also: **`nice`**.

More information: https://manned.org/renice.1p.

- Increase/decrease the priority of a running [p]rocess:

```
renice -n {{3}} -p {{pid}}
```

- Increase/decrease the priority of all processes owned by a [u]ser:

```
renice -n {{-4}} -u {{uid|user}}
```

- Increase/decrease the priority of all processes that belong to a process [g]roup:

```
renice -n {{5}} -g {{process_group}}
```

repomix

Pack a Github repository into an AI-friendly file.

More information: https://github.com/yamadashy/repomix.

- Output custom format:

```
repomix {{[-o|--output]}} {{path/to/file}} --style {{xml|markdown|plain}}
```

- Send output to `stdout`:

```
repomix --stdout > {{path/to/file}}
```

- Send output to `stdout`, then pipe into another program:

```
repomix --stdout | {{less}}
```

- Output with compression:

```
repomix --compress
```

- Process specific files:

```
repomix --include "{{src/**/*.ts}}" --ignore "{{**/*.test.ts}}"
```

- Pack a repository from a branch:

```
repomix --remote {{https://github.com/user/repo/tree/main}}
```

- Pack a repository at a specific commit:

```
repomix --remote {{https://github.com/user/repo/commit/836abcd7335137228ad77feb28655d85712680f1}}
```

- Pack repository using shorthand:

```
repomix --remote {{user/repo}}
```

repren

Multi-pattern string replacement and file renaming tool.

More information: https://github.com/jlevy/repren.

- Do a dry-run renaming a directory of PNGs with a literal string replacement:

```
repren {{[-n|--dry-run]}} --rename --literal --from '{{find_string}}' --to '{{replacement_string}}' {{*.png}}
```

- Do a dry-run renaming a directory of JPEGs with a `regex`:

```
repren --rename {{[-n|--dry-run]}} --from '{{regex}}' --to '{{replacement_string}}' {{*.jpg}} {{*.jpeg}}
```

- Do a find-and-replace on the contents of a directory of CSV files:

```
repren --from '{{([0-9]+) example_string}}' --to '{{replacement_string \1}}' {{*.csv}}
```

- Do both a find-and-replace and a rename operation at the same time, using a pattern file:

```
repren {{[-p|--patterns]}} {{path/to/patfile.ext}} --full {{*.txt}}
```

- Do a case-insensitive rename:

```
repren --rename {{[-i|--insensitive]}} {{[-p|--patterns]}} {{path/to/patfile.ext}} *
```

resolvconf

Manage nameserver information.

Acts as an intermediary between programs that supply nameserver information and applications that use this information.

This page documents Debian's implementation of **`resolvconf`**.

More information: https://manned.org/resolvconf.8.

- Add or override the IFACE.PROG record and run the update scripts if updating is enabled:

```
resolvconf -a {{IFACE.PROG}}
```

- Delete the IFACE.PROG record and run the update scripts if updating is enabled:

```
resolvconf -d {{IFACR.PROG}}
```

- Just run the update scripts if updating is enabled:

```
resolvconf -u
```

- Set the flag indicating whether `resolvconf` should run update scripts when invoked with `-a`, `-d` or `-u`:

```
resolvconf --enable-updates
```

- Clear the flag indicating whether to run updates:

```
resolvconf --disable-updates
```

- Check whether updates are enabled:

```
resolvconf --updates-are-enabled
```

restic

A fast, secure and secure backup program.

More information: https://restic.net.

- Initialize a backup repository in the specified local directory:

```
restic init --repo {{path/to/repository}}
```

- Backup a directory to the repository:

```
restic --repo {{path/to/repository}} backup {{path/to/directory}}
```

- Show backup snapshots currently stored in the repository:

```
restic --repo {{path/to/repository}} snapshots
```

- Restore a specific backup snapshot to a target directory:

```
restic --repo {{path/to/repository}} restore {{latest|snapshot_id}} --target {{path/to/target}}
```

- Restore a specific path from a specific backup to a target directory:

```
restic --repo {{path/to/repository}} restore {{snapshot_id}} --target {{path/to/target}} --include {{path/to/restore}}
```

- Clean up the repository and keep only the most recent snapshot of each unique backup:

```
restic forget --keep-last 1 --prune
```

resticprofile schedule

Schedule backups and run them in the background.

See also: **`restic`**, **`resticprofile`**, **`resticprofile-unschedule`**.

More information: https://creativeprojects.github.io/resticprofile/schedules/configuration/index.html.

- Schedule default profile:

```
resticprofile schedule
```

- Schedule a profile (default profile is "default"):

```
resticprofile --name "{{group_name}}" schedule
```

- Schedule all profiles:

```
resticprofile schedule --all
```

- Don't start the job after installing:

```
resticprofile schedule --no-start
```

- Display status of scheduled jobs for a profile:

```
resticprofile status {{[-n|--name]}} {{profile_name}}
```

- Run a scheduled job manually (used by system scheduler):

```
resticprofile run-schedule "backup@{{profile_name}}"
```

resticprofile unschedule

The command removes jobs for schedules declared in the selected profile or group (or of all profiles and groups).

See also: **`restic`**, **`resticprofile`**, **`resticprofile-schedule`**.

More information: https://creativeprojects.github.io/resticprofile/schedules/configuration/index.html.

- Unschedule a backup for a specific profile:

```
resticprofile unschedule {{[-n|--name]}} "{{profile_name}}"
```

- Unschedule all backups:

```
resticprofile unschedule --all
```

resticprofile

Configuration profiles manager for restic backup.

See also: **`restic`**, **`resticprofile-schedule`**, **`resticprofile-unschedule`**.

More information: https://creativeprojects.github.io/resticprofile/configuration/getting_started/index.html#write-your-first-configuration-file.

- List all saved snapshots:

```
resticprofile
```

- Run a backup on the default profile:

```
resticprofile backup
```

- Run a backup with a specific profile (default profile is "default"):

```
resticprofile {{[-n|--name]}} "{{profile_name}}" backup
```

- Run in dry-run mode and show the underlying restic commands:

```
resticprofile --dry-run backup
```

- Display the names of all profiles from the configuration file:

```
resticprofile profiles
```

- Generate shell completions:

```
resticprofile generate {{--bash-completion|--zsh-completion}}
```

- Show all details of a profile:

```
resticprofile show {{[-n|--name]}} "{{profile_name}}"
```

resume

Easily setup a new resume.

More information: https://github.com/jsonresume/resume-cli.

- Create a new `resume.json` file in the current working directory:

  ```
  resume init
  ```

- Validate a `resume.json` against schema tests to ensure it complies with the standard:

  ```
  resume validate
  ```

- Export a resume locally in a stylized HTML or PDF format:

  ```
  resume export {{path/to/html_or_pdf}}
  ```

- Start a web server that serves a local `resume.json`:

  ```
  resume serve
  ```

retry

Repeat command until it succeeds or a criterion is met.

More information: https://github.com/minfrin/retry.

- Retry a command until it succeeds:

```
retry {{command}}
```

- Retry a command every n seconds until it succeeds:

```
retry --delay={{n}} {{command}}
```

- Give up after n attempts:

```
retry --times={{n}} {{command}}
```

return

Exit a function or a script if run with **`source`**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-return.

- Exit a function prematurely:

```
{{func_name}}() { {{echo "This is reached"}}; return; {{echo "This is not"}}; }
```

- Specify the function's return value:

```
{{func_name}}() { return {{exit_code}}; }
```

reuse

Tool for compliance with the REUSE recommendations.

More information: https://reuse.readthedocs.io/en/stable/man/index.html.

- Lint for REUSE compliance for the current project (version control aware):

```
reuse lint
```

- Lint for REUSE compliance from the specified directory:

```
reuse --root {{path/to/directory}} lint
```

- Add copyright statement to file:

```
reuse annotate {{[-c|--copyright]}} "{{your_name}}
<{{your_email}}>" --fallback-dot-license {{path/to/file}}
```

- Add license information to file:

```
reuse annotate {{[-l|--license]}} {{spdx_identifier}} --
fallback-dot-license {{path/to/file}}
```

- Download a license by its SPDX identifier and place it in the LICENSES directory:

```
reuse download {{spdx-identifier}}
```

- Download all missing licenses detected in the project:

```
reuse download --all
```

rfetch

A configurable fetch program to output system information.

More information: https://github.com/kamui-fin/rfetch.

- Display system information:

```
rfetch
```

- Display system [a]rchitecture:

```
rfetch -a
```

- Display system up[t]ime:

```
rfetch -t
```

- Display system [k]ernel:

```
rfetch -k
```

- Display system [c]PU:

```
rfetch -c
```

- Display Linux [D]istro:

```
rfetch -D
```

- View [d]esktop environment:

```
rfetch -d
```

rg

Ripgrep, a recursive line-oriented search tool.

Aims to be a faster alternative to **`grep`**.

More information: https://github.com/BurntSushi/ripgrep/blob/master/GUIDE.md.

- Recursively search current directory for a pattern (`regex`):

```
rg {{pattern}}
```

- Recursively search for a pattern in a file or directory:

```
rg {{pattern}} {{path/to/file_or_directory}}
```

- Include hidden files and entries listed in `.gitignore`:

```
rg {{[-.|--hidden]}} --no-ignore {{pattern}}
```

- Only search the files whose names match the glob pattern(s) (e.g. `README.*`):

```
rg {{pattern}} {{[-g|--glob]}} {{filename_glob_pattern}}
```

- Recursively list filenames in the current directory that match a pattern:

```
rg --files | rg {{pattern}}
```

- Only list matched files (useful when piping to other commands):

```
rg {{[-l|--files-with-matches]}} {{pattern}}
```

- Show lines that do not match the pattern:

```
rg {{[-v|--invert-match]}} {{pattern}}
```

- Search for a literal string pattern:

```
rg {{[-F|--fixed-strings]}} -- {{string}}
```

rga

Ripgrep wrapper with rich file type searching capabilities.

More information: https://github.com/phiresky/ripgrep-all.

- Search recursively for a pattern in all files in the current directory:

```
rga {{regex}}
```

- List available adapters:

```
rga --rga-list-adapters
```

- Change which adapters to use (e.g. `ffmpeg`, `pandoc`, `poppler` etc.):

```
rga --rga-adapters={{adapter1,adapter2}} {{regex}}
```

- Search for a pattern using the mime type instead of the file extension (slower):

```
rga --rga-accurate {{regex}}
```

- Display help:

```
rga --help
```

rgb3toppm

Combine three PGM images into one PPM image.

See also: **ppmtorgb3**.

More information: https://netpbm.sourceforge.net/doc/rgb3toppm.html.

- Combine three PGM images (representing red, green and blue color components) into one PPM image:

```
rgb3toppm {{path/to/red_file.ppm}} {{path/to/green_file.ppm}} {{path/to/blue_file.ppm}} > {{path/to/output.ppm}}
```

rgpt

An automated code review tool that uses GPT you can use straight from your terminal.

More information: https://github.com/vibovenkat123/review-gpt.

- Ask GPT to improve the code with no extra options:

```
rgpt --i "$(git diff {{path/to/file}})"
```

- Get a more detailed verbose output from `rgpt` while reviewing the code:

```
rgpt --v --i "$(git diff {{path/to/file}})"
```

- Ask GPT to improve the code and limit it to a certain amount of GPT3 tokens:

```
rgpt --max {{300}} --i "$(git diff {{path/to/file}})"
```

- Ask GPT for a more unique result using a float value between 0 and 2. (higher = more unique):

```
rgpt --pres {{1.2}} --i "$(git diff {{path/to/file}})"
```

- Ask GPT to review your code using a specific model:

```
rgpt --model {{davinci}} --i "$(git diff {{path/to/file}})"
```

- Make `rgpt` use a JSON output:

```
rgpt --json --i "$(git diff {{path/to/file}})"
```

rgrep

Recursively find patterns in files using **`regex`**.

Equivalent to **`grep -r`**.

More information: https://www.gnu.org/software/grep/manual/grep.html#Command_002dline-Options.

- Recursively search for a pattern in the current working directory:

```
rgrep "{{search_pattern}}"
```

- Recursively search for a case-insensitive pattern in the current working directory:

```
rgrep {{[-i|--ignore-case]}} "{{search_pattern}}"
```

- Recursively search for an extended `regex` pattern (supports `?`, `+`, `{}`, `()` and `|`) in the current working directory:

```
rgrep {{[-E|--extended-regexp]}} "{{search_pattern}}"
```

- Recursively search for an exact string (disables `regex`) in the current working directory:

```
rgrep {{[-F|--fixed-strings]}} "{{exact_string}}"
```

- Recursively search for a pattern in a specified directory (or file):

```
rgrep "{{search_pattern}}" {{path/to/file_or_directory}}
```

rhash

Calculate or check common message digests.

More information: https://rhash.sourceforge.net/manpage.php.

- Calculate default CRC32 digests of a file:

```
rhash {{path/to/file}}
```

- Recursively process a directory to generate an SFV file using SHA1:

```
rhash --sha1 --recursive {{path/to/folder}} > {{path/to/output.sfv}}
```

- Verify the integrity of files based on an SFV file:

```
rhash --check {{path/to/file.sfv}}
```

- Calculate the SHA3 digest of a text message:

```
rhash --sha3-256 --message '{{message}}'
```

- Calculate CRC32 digest of a file and output digest encoded in base64 using BSD format:

```
rhash --base64 --bsd {{path/to/file}}
```

- Use custom output template:

```
rhash --printf '{{%p\t%s\t%{mtime}\t%m\n}}' {{path/to/file}}
```

rich

A toolbox for fancy output in the terminal.

More information: https://github.com/Textualize/rich-cli.

- Display a file with syntax highlighting:

```
rich {{path/to/file.py}}
```

- Add line numbers, and indentation guides:

```
rich {{path/to/file.py}} --line-numbers --guides
```

- Apply a theme:

```
rich {{path/to/file.py}} --theme {{monokai}}
```

- Display a file in an interactive pager:

```
rich {{path/to/file.py}} --pager
```

- Display contents from a URL:

```
rich {{https://raw.githubusercontent.com/Textualize/rich-cli/main/README.md}} --markdown --pager
```

- Export a file as HTML:

```
rich {{path/to/file.md}} --export-html {{path/to/file.html}}
```

- Display text with formatting tags, custom alignment, and line width:

```
rich --print "{{Hello [green on black]Stylized[/green on black] [bold]World[/bold]}}" --{{left|center|right}} --width {{10}}
```

rip

Remove files or directories by sending them to the graveyard, allowing for them to be recovered.

More information: https://github.com/nivekuil/rip.

- Remove files or directories from specified locations and place them in the graveyard:

```
rip {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Interactively remove files or directories, with a prompt before every removal:

```
rip --inspect {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- List all files and directories in the graveyard that were originally within the current directory:

```
rip --seance
```

- Permanently delete every file and directory in the graveyard:

```
rip --decompose
```

- Put back the files and directories which were affected by the most recent removal:

```
rip --unbury
```

- Put back every file and directory that is listed by `rip --seance`:

```
rip --seance --unbury
```

ripgrep

`ripgrep` is the common name for the command **`rg`**.

- View documentation for the original command:

  ```
  tldr rg
  ```

rkdeveloptool

Flash, dump, and manage boot firmware for Rockchip-based computer devices.

You will need to turn on the device into Maskrom/Bootrom mode before connecting it through USB.

Some subcommands may require to run as root.

More information: https://github.com/rockchip-linux/rkdeveloptool.

- [l]ist all connected Rockchip-based flash [d]evices:

```
rkdeveloptool ld
```

- Initialize the device by forcing it to [d]ownload and install the [b]ootloader from the specified file:

```
rkdeveloptool db {{path/to/bootloader.bin}}
```

- [u]pdate the boot[l]oader software with a new one:

```
rkdeveloptool ul {{path/to/bootloader.bin}}
```

- Write an image to a GPT-formatted flash partition, specifying the initial storage sector (usually `0x0` alias `0`):

```
rkdeveloptool wl {{initial_sector}} {{path/to/image.img}}
```

- Write to the flash partition by its user-friendly name:

```
rkdeveloptool wlx {{partition_name}} {{path/to/image.img}}
```

- [r]eset/reboot the [d]evice, exit from the Maskrom/Bootrom mode to boot into the selected flash partition:

```
rkdeveloptool rd
```

rlatopam

Convert Alias/Wavefront RLA and RPF image files to Netpbm format.

More information: https://netpbm.sourceforge.net/doc/rlatopam.html.

- Convert the specified SPOT image to PGM format:

```
rlatopam {{path/to/file}} > {{path/to/output.pam}}
```

rletopnm

Convert a Utah Raster Tools RLE image file to a PNM file.

More information: https://netpbm.sourceforge.net/doc/rletopnm.html.

- Convert an RLE image to a PNM file:

```
rletopnm {{path/to/input.rle}} > {{path/to/output.pnm}}
```

- Create a PGM image containing the RLE file's alpha channel:

```
rletopnm {{[--a|--alphaout]}} {{path/to/alpha_file.pgm}}
{{path/to/input.rle}} > {{path/to/output.pnm}}
```

- Operate in verbose mode and print the contents of the RLE header to `stdout`:

```
rletopnm {{[--verb|--verbose]}} {{path/to/input.rle}} >
{{path/to/output.pnm}}
```

rlwrap

Add line editing, persistent history and prompt completion to a REPL command.

More information: https://manned.org/rlwrap.

- Run a REPL command with line editing, persistent history and prompt completion:

```
rlwrap {{command}}
```

- Use all words seen on input and output for prompt completion:

```
rlwrap {{[-r|--remember]}} {{command}}
```

- Better prompt completion if prompts contain ANSI colour codes:

```
rlwrap {{[-A|--ansi-colour-aware]}} {{command}}
```

- Enable filename completion (case sensitive):

```
rlwrap {{[-c|--complete-filenames]}} {{command}}
```

- Add coloured prompts, use colour name, or an ASCI-conformant colour specification. Use an uppercase colour name for bold styling:

```
rlwrap {{[-p|--prompt-colour=]}}{{black|red|green|yellow|
blue|cyan|purple|white|colour_spec}} {{command}}
```

rm

Remove files or directories.

See also: **`rmdir`**, **`trash`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/rm-invocation.html.

- Remove specific files:

```
rm {{path/to/file1 path/to/file2 ...}}
```

- Remove specific files ignoring nonexistent ones:

```
rm {{[-f|--force]}} {{path/to/file1 path/to/file2 ...}}
```

- Remove specific files interactively prompting before each removal:

```
rm {{[-i|--interactive]}} {{path/to/file1 path/to/file2 ...}}
```

- Remove specific files printing info about each removal:

```
rm {{[-v|--verbose]}} {{path/to/file1 path/to/file2 ...}}
```

- Remove specific files and directories recursively:

```
rm {{[-r|--recursive]}} {{path/to/file_or_directory1 path/to/file_or_directory2 ...}}
```

- Remove empty directories (this is considered the safe method):

```
rm {{[-d|--dir]}} {{path/to/directory}}
```

rmdir

Remove directories without files.

See also: **rm**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/rmdir-invocation.html.

- Remove specific directories:

```
rmdir {{path/to/directory1 path/to/directory2 ...}}
```

- Remove specific nested directories recursively:

```
rmdir {{[-p|--parents]}} {{path/to/directory1 path/to/directory2 ...}}
```

- Clean a directory of empty directories:

```
rmdir *
```

rmlint

Find space waste and other broken things on your filesystem.

More information: https://rmlint.readthedocs.io/en/latest/rmlint.1.html.

- Check directories for duplicated, empty and broken files:

```
rmlint {{path/to/directory1 path/to/directory2 ...}}
```

- Check for duplicates bigger than a specific size, preferably keeping files in tagged directories (after the double slash):

```
rmlint {{[-s|--size]}} {{1MB}} {{path/to/directory}} //
{{path/to/original_directory}}
```

- Check for space wasters, keeping everything in the untagged directories:

```
rmlint {{[-k|--keep-all-untagged]}} {{path/to/directory}} //
{{path/to/original_directory}}
```

- Delete duplicate files found by an execution of `rmlint`:

```
./rmlint.sh
```

- Find duplicate directory trees based on data, ignoring names:

```
rmlint {{[-D|--merge-directories]}} {{path/to/directory}}
```

- Mark files at lower path [d]epth as originals, on tie choose shorter [l]ength:

```
rmlint {{[-S|--rank-by]}} {{dl}} {{path/to/directory}}
```

- Find files with identical filename and contents, and link rather than delete the duplicates:

```
rmlint {{[-c|--config]}} sh:link {{[-b|--match-basename]}}
{{path/to/directory}}
```

- Use `data` as master directory. Find only duplicates in backup that are also in `data`. Do not delete any files in `data`:

```
rmlint {{path/to/backup}} // {{path/to/data}} {{[-k|--keep-
all-tagged]}} {{[-m|--must-match-tagged]}}
```

rnano

This command is an alias of **`nano --restricted`**.

More information: https://manned.org/rnano.

- View documentation for the original command:

```
tldr nano
```

roave-backward-compatibility-check

Verify backward compatibility breaks between two versions of a PHP library.

More information: https://github.com/Roave/BackwardCompatibilityCheck.

- Check for breaking changes since the last tag:

```
roave-backward-compatibility-check
```

- Check for breaking changes since a specific tag:

```
roave-backward-compatibility-check --from={{git_reference}}
```

- Check for breaking changes between the last tag and a specific reference:

```
roave-backward-compatibility-check --to={{git_reference}}
```

- Check for breaking changes and output to Markdown:

```
roave-backward-compatibility-check --format=markdown >
{{results.md}}
```

robo

PHP task runner.

More information: https://robo.li/getting-started.html.

- List available commands:

```
robo list
```

- Run a specific command:

```
robo {{command}}
```

- Simulate running a specific command:

```
robo --simulate {{command}}
```

roll

Rolls a user-defined dice sequence.

More information: https://manned.org/roll.

- Roll 3 6-sided dice and sums the results:

```
roll {{3d}}
```

- Roll 1 8-sided die, add 3 and sum the results:

```
roll {{d8 + 3}}
```

- Roll 4 6-sided dice, keep the 3 highest results and sum the results:

```
roll {{4d6h3}}
```

- Roll 2 12-sided dice 2 times and show every roll:

```
roll --verbose {{2{2d12}}}
```

- Roll 2 20-sided dice until the result is bigger than 10:

```
roll "{{2d20>10}}"
```

- Roll 2 5-sided dice 3 times and show the total sum:

```
roll --sum-series {{3{2d5}}}
```

ROPgadget

Find ROP gadgets in binary files.

More information: https://github.com/JonathanSalwan/ROPgadget.

- List gadgets in the binary file:

```
ROPgadget --binary {{path/to/binary}}
```

- Filter gadgets in the binary file by a `regex`:

```
ROPgadget --binary {{path/to/binary}} --re {{regex}}
```

- List gadgets in the binary file, excluding specified type:

```
ROPgadget --binary {{path/to/binary}} --{{norop|nojob|nosys}}
```

- Exclude bad byte gadgets in the binary file:

```
ROPgadget --binary {{path/to/binary}} --badbytes
{{byte_string}}
```

- List gadgets up to the specified number of bytes in the binary file:

```
ROPgadget --binary {{path/to/binary}} --depth {{nbyte}}
```

ropper

Find ROP gadgets in binary files.

More information: https://scoding.de/ropper/.

- List gadgets in the binary file:

```
ropper --file {{path/to/binary}}
```

- Filter gadgets in the binary file by a `regex`:

```
ropper --file {{path/to/binary}} --search {{regex}}
```

- List gadgets of specified type in the binary file:

```
ropper --file {{path/to/binary}} --type {{rop|job|sys|all}}
```

- Exclude bad byte gadgets in the binary file:

```
ropper --file {{path/to/binary}} --badbytes {{byte_string}}
```

- List gadgets up to the specified instruction count in the binary file:

```
ropper --file {{path/to/binary}} --inst-count {{count}}
```

route

Show and manipulate the route table.

More information: https://manned.org/route.

- Display the information of route table:

```
route -n
```

- Add route rule:

```
sudo route add -net {{ip_address}} netmask
{{netmask_address}} gw {{gw_address}}
```

- Delete route rule:

```
sudo route del -net {{ip_address}} netmask
{{netmask_address}} dev {{gw_address}}
```

rpcdump.py

Dump remote RPC endpoints information via the Endpoint Mapper.

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- Dump RPC endpoints using username and password:

```
rpcdump.py {{domain}}/{{username}}:{{password}}@{{target}}
```

- Dump RPC endpoints using NTLM hashes:

```
rpcdump.py -hashes {{LMHASH}}:{{NTHASH}} {{domain}}/
{{username}}:{{password}}@{{target}}
```

- Specify a target IP address explicitly (useful if the target name is a NetBIOS name):

```
rpcdump.py -target-ip {{target_ip}} {{domain}}/{{username}}:
{{password}}@{{target}}
```

- Connect to a specific port (default is 135 for RPC Endpoint Mapper):

```
rpcdump.py -port {{port_number}} {{domain}}/{{username}}:
{{password}}@{{target}}
```

- Enable debug output:

```
rpcdump.py -debug {{domain}}/{{username}}:{{password}}
@{{target}}
```

rpcmap.py

Lookup listening MSRPC interfaces using a string binding (e.g., **`ncacn_ip_tcp:host[port]`**).

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- Connect to an MSRPC interface using a string binding (e.g., `ncacn_ip_tcp:host[port]`):

```
rpcmap.py {{stringbinding}}
```

- Bruteforce UUIDs even if the MGMT interface is available:

```
rpcmap.py -brute-uuids {{stringbinding}}
```

- Bruteforce operation numbers (opnums) for discovered UUIDs:

```
rpcmap.py -brute-opnums {{stringbinding}}
```

- Bruteforce major versions of found UUIDs:

```
rpcmap.py -brute-versions {{stringbinding}}
```

- Specify a target IP address manually:

```
rpcmap.py -target-ip {{ip_address}} {{stringbinding}}
```

- Authenticate to the RPC interface with username and password:

```
rpcmap.py -auth-rpc {{domain}}/{{username}}:{{password}}
{{stringbinding}}
```

- Authenticate using NTLM hashes for RPC:

```
rpcmap.py -hashes-rpc {{LMHASH:NTHASH}} {{stringbinding}}
```

- Enable debug output for verbose information:

```
rpcmap.py -debug {{stringbinding}}
```

rpi-imager

Flash images onto storage devices.

More information: https://github.com/raspberrypi/rpi-imager.

- Write a specific image to a specific block device:

```
rpi-imager --cli {{path/to/image.zip}} {{/dev/sdX}}
```

- Write a specific image to a block device, disabling the checksum verification:

```
rpi-imager --cli --disable-verify {{path/to/image.zip}} {{/dev/sdX}}
```

- Write a specific image to a block device, which will expect a specific checksum when running the verification:

```
rpi-imager --cli --sha256 {{expected_hash}} {{path/to/image.zip}} {{/dev/sdX}}
```

rpk

Manage Redpanda topics, clusters, groups, security and more via a single binary.

More information: https://docs.redpanda.com/current/reference/rpk/.

- Create a new topic:

```
rpk topic create {{topic_name}}
```

- Produce a message to a topic:

```
rpk topic produce {{topic_name}}
```

- Consume messages from multiple topics:

```
rpk topic consume {{topic_name1 topic_name2 ...}}
```

- List all topics:

```
rpk topic list
```

- Display cluster information:

```
rpk cluster info
```

- List all consumer groups:

```
rpk group list
```

- Describe a consumer group with lag information:

```
rpk group describe {{group_name}}
```

- Display version:

```
rpk version
```

rr

Debugging tool designed to record and replay program execution.

More information: https://github.com/rr-debugger/rr/wiki/Usage.

- Record an application:

```
rr record {{path/to/binary --arg1 --arg2}}
```

- Replay latest recorded execution:

```
rr replay
```

RsaCtfTool.py

RSA attack tool for CTF challenges - recover private keys from weak public keys and/or decrypt data.

More information: https://github.com/RsaCtfTool/RsaCtfTool.

- Recover a private key from a public key file:

```
RsaCtfTool.py --publickey {{path/to/key.pub}} --private
```

- Decrypt a file using a public key:

```
RsaCtfTool.py --publickey {{path/to/key.pub}} --decryptfile
{{path/to/ciphered_file}}
```

- Decrypt a specific ciphertext string:

```
RsaCtfTool.py --publickey {{path/to/key.pub}} --decrypt
"{{ciphertext}}"
```

- Dump RSA key components (e.g., modulus, exponent) from a key file:

```
RsaCtfTool.py --dumpkey --key {{path/to/key.pub}}
```

- Run a specific attack (e.g., Fermat factorization) to recover the private key:

```
RsaCtfTool.py --publickey {{path/to/key.pub}} --private --
attack fermat
```

- Generate a public key from modulus (n) and exponent (e):

```
RsaCtfTool.py --createpub -n {{modulus}} -e {{exponent}}
```

- Attempt all available attacks to recover the private key:

```
RsaCtfTool.py --publickey {{path/to/key.pub}} --private --
attack all
```

Rscript

Run a script with the R programming language.

More information: https://manned.org/Rscript.

- Run a script:

```
Rscript {{path/to/file.R}}
```

- Run a script in vanilla mode (i.e. a blank session that doesn't save the workspace at the end):

```
Rscript --vanilla {{path/to/file.R}}
```

- Execute one or more R expressions:

```
Rscript -e {{expression1}} -e {{expression2}}
```

- Display R version:

```
Rscript --version
```

rspec

Behavior-driven development testing framework written in Ruby to test Ruby code.

More information: https://rspec.info.

- Initialize an .rspec configuration and a spec helper file:

```
rspec --init
```

- Run all tests:

```
rspec
```

- Run a specific directory of tests:

```
rspec {{path/to/directory}}
```

- Run one or more test files:

```
rspec {{path/to/file1 path/to/file2 ...}}
```

- Run a specific test in a file (e.g. the test starts on line 83):

```
rspec {{path/to/file}}:{{83}}
```

- Run specs with a specific seed:

```
rspec --seed {{seed_number}}
```

rsql

SQL client to interface with databases and other data sources inside the terminal.

More information: https://github.com/theseus-rs/rsql.

- Enter interactive mode:

```
rsql
```

- Connect to a database (e.g. PostgreSQL):

```
rsql --url "{{postgresql://user:pass@localhost/mydb}}"
```

- Connect to a PostgreSQL database with SSL:

```
rsql --url "{{postgresql://user:pass@localhost/db?
sslmode=require}}"
```

- Connect to a MySQL database with a specified charset:

```
rsql --url "{{mysql://user:pass@localhost/db?
charset=utf8mb4}}"
```

- Run a query and exit:

```
rsql --url "{{sqlite://database.db}}" -- "{{SELECT * FROM
users LIMIT 10}}"
```

- Set default format:

```
rsql --url "{{sqlite://db.sqlite}}" --format json
```

- Connect to file and use custom line separator:

```
rsql --url "{{delimited://data.txt?separator=|
&has_header=true}}"
```

rsstail

`tail` for RSS feeds.

More information: https://manned.org/rsstail.

- Show the feed of a given URL and wait for new entries appearing at the bottom:

```
rsstail -u {{url}}
```

- Show the feed in reverse chronological order (newer at the bottom):

```
rsstail -r -u {{url}}
```

- Include publication date and link:

```
rsstail -pl -u {{url}}
```

- Set update interval:

```
rsstail -u {{url}} -i {{interval_in_seconds}}
```

- Show feed and exit:

```
rsstail -1 -u {{url}}
```

rsync

Transfer files either to or from a remote host (but not between two remote hosts), by default using SSH.

To specify a remote path, use **`user@host:path/to/file_or_directory`**.

More information: https://download.samba.org/pub/rsync/rsync.1.

- Transfer a file:

```
rsync {{path/to/source}} {{path/to/destination}}
```

- Use archive mode (recursively copy directories, copy symlinks without resolving, and preserve permissions, ownership and modification times):

```
rsync {{[-a|--archive]}} {{path/to/source}} {{path/to/destination}}
```

- Compress the data as it is sent to the destination, display verbose and human-readable progress, and keep partially transferred files if interrupted:

```
rsync {{[-zvhP|--compress --verbose --human-readable --partial --progress]}} {{path/to/source}} {{path/to/destination}}
```

- Recursively copy directories:

```
rsync {{[-r|--recursive]}} {{path/to/source}} {{path/to/destination}}
```

- Transfer directory contents, but not the directory itself:

```
rsync {{[-r|--recursive]}} {{path/to/source}}/ {{path/to/destination}}
```

- Use archive mode, resolve symlinks, and skip files that are newer on the destination:

```
rsync {{[-auL|--archive --update --copy-links]}} {{path/to/source}} {{path/to/destination}}
```

- Transfer a directory from a remote host running `rsyncd` and delete files on the destination that do not exist on the source:

```
rsync {{[-r|--recursive]}} --delete rsync://{{host}}:{{path/
to/source}} {{path/to/destination}}
```

- Transfer a file over SSH using a different port than the default (22) and show global progress:

```
rsync {{[-e|--rsh]}} 'ssh -p {{port}}' --info=progress2
{{host}}:{{path/to/source}} {{path/to/destination}}
```

rtl_sdr

Raw data recorder for RTL-SDR receivers.

Data is encoded using I/Q sampling (aka quadrature sampling).

More information: https://osmocom.org/projects/rtl-sdr/wiki/Rtl-sdr.

- Save RAW data from a frequency (specified in Hz) to a file:

```
rtl_sdr -f {{100000000}} {{path/to/file}}
```

- Pipe data to another program:

```
rtl_sdr -f {{100000000}} - | {{aplay}}
```

- Read a specified number of samples:

```
rtl_sdr -f {{100000000}} -n {{20}} -
```

- Specify the sample rate in Hz (ranges 225001-300000 and 900001-3200000):

```
rtl_sdr -f {{100000000}} -s {{2400000}} -
```

- Specify the device by its index:

```
rtl_sdr -f {{100000000}} -d {{0}} -
```

- Specify the gain:

```
rtl_sdr -f {{100000000}} -g {{20}} -
```

- Specify the output block size:

```
rtl_sdr -f {{100000000}} -b {{9999999}} -
```

- Use synchronous output:

```
rtl_sdr -f {{100000000}} -S -
```

rtmpdump

Dump media content streamed over the RTMP protocol.

More information: https://rtmpdump.mplayerhq.hu/rtmpdump.1.html.

- Download a file:

```
rtmpdump {{[-r|--rtmp]}} {{rtmp://example.com/path/to/video}}
{{[-o|--flv]}} {{file.ext}}
```

- Download a file from a Flash player:

```
rtmpdump {{[-r|--rtmp]}} {{rtmp://example.com/path/to/video}}
{{[-W|--swfVfy]}} {{http://example.com/player}} {{[-f|--
flashVer]}} "{{LNX 10,0,32,18}}" {{[-o|--flv]}} {{file.ext}}
```

- Specify connection parameters if they are not detected correctly:

```
rtmpdump {{[-r|--rtmp]}} {{rtmp://example.com/path/to/video}}
{{[-a|--app]}} {{app_name}} {{[-y|--playpath]}} {{path/to/
video}} {{[-o|--flv]}} {{file.ext}}
```

- Download a file from a server that requires a referrer:

```
rtmpdump {{[-r|--rtmp]}} {{rtmp://example.com/path/to/video}}
{{[-p|--pageUrl]}} {{http://example.com/webpage}} {{[-o|--
flv]}} {{file.ext}}
```

rtv

Reddit Terminal Viewer.

Use **`<ArrowKeys>`** to navigate. **`<ArrowRight>`** and **`<ArrowLeft>`** to view and return from a submission, respectively.

More information: https://github.com/michael-lazar/rtv.

- Open the front page:

  `/front`

- Open a subreddit:

  `/r/{{subreddit_name}}`

- Expand/collapse comments:

  `<Space>`

- Open link:

  `<o>`

- Log in:

  `<u>`

- Display help:

  `<?>`

rubocop

Lint Ruby files.

More information: https://docs.rubocop.org/rubocop/usage/basic_usage.html.

- Check all files in the current directory (including subdirectories):

```
rubocop
```

- Check one or more specific files or directories:

```
rubocop {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Write output to file:

```
rubocop --out {{path/to/file}}
```

- View list of cops (linter rules):

```
rubocop --show-cops
```

- Exclude a cop:

```
rubocop --except {{cop1 cop2 ...}}
```

- Run only specified cops:

```
rubocop --only {{cop1 cop2 ...}}
```

- Auto-correct files (experimental):

```
rubocop --auto-correct
```

ruby

Ruby programming language interpreter.

See also: **`gem`**, **`bundler`**, **`rake`**, **`irb`**.

More information: https://manned.org/ruby.

- Execute a Ruby script:

```
ruby {{path/to/script.rb}}
```

- Execute a single Ruby command in the command-line:

```
ruby -e "{{command}}"
```

- Check for syntax errors on a given Ruby script:

```
ruby -c {{path/to/script.rb}}
```

- Start the built-in HTTP server on port 8080 in the current directory:

```
ruby -run -e httpd
```

- Locally execute a Ruby binary without installing the required library it depends on:

```
ruby -I {{path/to/library_folder}} -r
{{library_require_name}} {{path/to/bin_folder/bin_name}}
```

- Display version:

```
ruby {{[-v|--version]}}
```

ruff check

An extremely fast Python linter. **check** is the default command - it can be omitted everywhere.

If no files or directories are specified, the current working directory is used by default.

More information: https://docs.astral.sh/ruff/linter.

- Run the linter on the given files or directories:

```
ruff check {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Apply the suggested fixes, modifying the files in-place:

```
ruff check --fix
```

- Run the linter and re-lint on change:

```
ruff check --watch
```

- Only enable the specified rules (or all rules), ignoring the configuration file:

```
ruff check --select {{ALL|rule_code1,rule_code2,...}}
```

- Additionally enable the specified rules:

```
ruff check --extend-select {{rule_code1,rule_code2,...}}
```

- Disable the specified rules:

```
ruff check --ignore {{rule_code1,rule_code2,...}}
```

- Ignore all existing violations of a rule by adding `# noqa` directives to all lines that violate it:

```
ruff check --select {{rule_code}} --add-noqa
```

ruff format

An extremely fast Python code formatter.

If no files or directories are specified, the current working directory is used by default.

More information: https://docs.astral.sh/ruff/formatter.

- Format given files or directories in-place:

```
ruff format {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Print which files would have been modified and return a non-zero exit code if there are files to reformat, and zero otherwise:

```
ruff format --check
```

- Print what changes would be made without modifying the files:

```
ruff format --diff
```

ruff

An extremely fast Python linter and code formatter, written in Rust.

More information: https://docs.astral.sh/ruff/tutorial.

- View documentation for the Ruff linter:

```
tldr ruff check
```

- View documentation for the Ruff code formatter:

```
tldr ruff format
```

runit

3-stage init system.

More information: https://smarden.org/runit/runit.8.html.

- Start runit's 3-stage init scheme:

```
runit
```

- Shut down runit:

```
kill --CONT {{runit_pid}}
```

runsv

Start and manage a runit service.

More information: https://manned.org/runsv.8.

- Start a runit service as the current user:

```
runsv {{path/to/service}}
```

- Start a runit service as root:

```
sudo runsv {{path/to/service}}
```

runsvchdir

Change the directory **`runsvdir`** uses by default.

More information: https://manned.org/runsvchdir.8.

- Switch `runsvdir` directories:

```
sudo runsvchdir {{path/to/directory}}
```

runsvdir

Run an entire directory of services.

More information: https://manned.org/runsvdir.8.

- Start and manage all services in a directory as the current user:

```
runsvdir {{path/to/services}}
```

- Start and manage all services in a directory as root:

```
sudo runsvdir {{path/to/services}}
```

- Start services in separate sessions:

```
runsvdir -P {{path/to/services}}
```

rustc

The Rust compiler.

Rust projects usually use **`cargo`** instead of invoking **`rustc`** directly.

More information: https://doc.rust-lang.org/rustc.

- Compile a binary crate:

```
rustc {{path/to/main.rs}}
```

- Compile with optimizations (`s` means optimize for binary size; `z` is the same with even more optimizations):

```
rustc {{[-C|--codegen]}} lto {{[-C|--codegen]}} opt-level={{0|1|2|3|s|z}} {{path/to/main.rs}}
```

- Compile with debugging information:

```
rustc -g {{path/to/main.rs}}
```

- Explain an error message:

```
rustc --explain {{error_code}}
```

- Compile with architecture-specific optimizations for the current CPU:

```
rustc {{[-C|--codegen]}} target-cpu={{native}} {{path/to/main.rs}}
```

- Display the target list (Note: You have to add a target using `rustup` first to be able to compile for it):

```
rustc --print target-list
```

- Compile for a specific target:

```
rustc --target {{target_triple}} {{path/to/main.rs}}
```

rustdoc

Generate documentation for a Rust crate.

More information: https://doc.rust-lang.org/stable/rustdoc.

- Generate documentation from the crate's root:

```
rustdoc {{src/lib.rs}}
```

- Pass a name for the project:

```
rustdoc {{src/lib.rs}} --crate-name {{name}}
```

- Generate documentation from Markdown files:

```
rustdoc {{path/to/file.md}}
```

- Specify the output directory:

```
rustdoc {{src/lib.rs}} --out-dir {{path/to/output_directory}}
```

rustfmt

Format Rust source code.

More information: https://github.com/rust-lang/rustfmt.

- Format a file, overwriting the original file in-place:

```
rustfmt {{path/to/source.rs}}
```

- Check a file for formatting and display any changes on the console:

```
rustfmt --check {{path/to/source.rs}}
```

- Backup any modified files before formatting (the original file is renamed with a `.bk` extension):

```
rustfmt --backup {{path/to/source.rs}}
```

- Format code using a specific Rust style edition (formatting rules) verbosely:

```
rustfmt --style-edition {{2015|2018|2021|2024}} {{[-v|--verbose]}} {{path/to/source1.rs path/to/source2.rs ...}}
```

- Format code using a specific Rust edition (language features and parsing):

```
rustfmt --edition {{2015|2018|2021|2024}} {{path/to/source1.rs path/to/source2.rs ...}}
```

rustic

Create fast, encrypted, deduplicated backups powered by Rust.

More information: https://github.com/rustic-rs/rustic.

- Initialize a new repository:

```
rustic init --repository {{/srv/rustic-repo}}
```

- Create a new backup of a file/directory to a repository:

```
rustic backup --repository {{/srv/rustic-repo}} {{path/to/file_or_directory}}
```

rustscan

Modern Port Scanner written in Rust.

Note: **`nmap`** must be installed for some of the examples below to work.

More information: https://github.com/bee-san/RustScan/wiki.

- Scan all ports of one or more comma-delimited addresses using the default values:

```
rustscan {{[-a|--addresses]}} {{ip_or_hostname}}
```

- Scan the top 1000 ports with service and version detection:

```
rustscan --top {{[-a|--addresses]}} {{address_or_addresses}}
```

- Scan a specific list of ports:

```
rustscan {{[-p|--ports]}} {{port1,port2,...}} {{[-a|--
addresses]}} {{address_or_addresses}}
```

- Scan a specific range of ports:

```
rustscan {{[-r|--range]}} {{start}}-{{end}} {{[-a|--
addresses]}} {{address_or_addresses}}
```

- Invoke `nmap` functionalities (Nmap's OS detection and default scripts):

```
rustscan {{[-a|--addresses]}} {{address_or_addresses}} -- -O
{{[-sC|--script=default]}}
```

- Scan with custom batch size (default: 4500) and timeout (default: 1500ms):

```
rustscan {{[-b|--batch-size]}} {{batch_size}} {{[-t|--
timeout]}} {{timeout}} {{[-a|--addresses]}}
{{address_or_addresses}}
```

- Scan with specific port order:

```
rustscan --scan-order {{serial|random}} {{[-a|--addresses]}}
{{address_or_addresses}}
```

- Scan in greppable mode (only output of the ports, no `nmap`):

```
rustscan {{[-g|--greppable]}} {{[-a|--addresses]}}
{{address_or_addresses}}
```

rustup check

Check for updates to Rust toolchains and **rustup**.

More information: https://rust-lang.github.io/rustup.

- Check for all updates:

```
rustup check
```

rustup completions

Generate shell completions for **`rustup`** and **`cargo`**.

More information: https://rust-lang.github.io/rustup.

- Print the completion script to `stdout`:

```
rustup completions {{bash|elvish|fish|powershell|zsh}}
{{rustup|cargo}}
```

rustup component

Modify a toolchain's installed components.

Without the **`--toolchain`** option **`rustup`** will use the default toolchain. See **`rustup help toolchain`** for more information about toolchains.

More information: https://rust-lang.github.io/rustup.

- Add a component to a toolchain:

```
rustup component add --toolchain {{toolchain}} {{component}}
```

- Remove a component from a toolchain:

```
rustup component remove --toolchain {{toolchain}}
{{component}}
```

- List installed and available components for a toolchain:

```
rustup component list --toolchain {{toolchain}}
```

- List installed components for a toolchain:

```
rustup component list --toolchain {{toolchain}} --installed
```

rustup default

Set the default Rust toolchain.

More information: https://rust-lang.github.io/rustup.

- Switch the default Rust toolchain (see `rustup help toolchain` for more information):

```
rustup default {{toolchain}}
```

rustup doc

Open the offline Rust documentation for the current toolchain.

There are a lot more documentation pages not mentioned here. See **`rustup help doc`** for more information.

More information: https://rust-lang.github.io/rustup.

- Open the main page:

  ```
  rustup doc
  ```

- Open the documentation for a specific topic (a module in the standard library, a type, a keyword, etc.):

  ```
  rustup doc {{std::fs|usize|fn|...}}
  ```

- Open the Rust Programming Language book:

  ```
  rustup doc --book
  ```

- Open the Cargo book:

  ```
  rustup doc --cargo
  ```

- Open the Rust Reference:

  ```
  rustup doc --reference
  ```

rustup help

Display help on **`rustup`** and its subcommands.

More information: https://rust-lang.github.io/rustup.

- Display help:

```
rustup help
```

- Display help for a subcommand:

```
rustup help {{subcommand}}
```

rustup-init.sh

Script to install **`rustup`** and the Rust toolchain.

More information: https://forge.rust-lang.org/infra/other-installation-methods.html#rustup.

- Download and run `rustup-init` to install `rustup` and the default Rust toolchain:

```
curl https://sh.rustup.rs {{[-sSf|--silent --show-error --fail]}} | sh -s
```

- Download and run `rustup-init` and pass arguments to it:

```
curl https://sh.rustup.rs {{[-sSf|--silent --show-error --fail]}} | sh -s -- {{arguments}}
```

- Run `rustup-init` and specify additional components or targets to install:

```
rustup-init.sh --target {{target}} --component {{component}}
```

- Run `rustup-init` and specify the default toolchain to install:

```
rustup-init.sh --default-toolchain {{toolchain}}
```

- Run `rustup-init` and do not install any toolchain:

```
rustup-init.sh --default-toolchain {{none}}
```

- Run `rustup-init` and specify an installation profile:

```
rustup-init.sh --profile {{minimal|default|complete}}
```

- Run `rustup-init` without asking for confirmation:

```
rustup-init.sh -y
```

rustup install

This command is an alias of **`rustup toolchain install`**.

- View documentation for the original command:

  ```
  tldr rustup toolchain
  ```

rustup man

View the man page for a command managed by **`rustup`**.

More information: https://rust-lang.github.io/rustup.

- View the man page for a given command from the default toolchain:

```
rustup man {{command}}
```

- View the man page for a given command from the specified toolchain:

```
rustup man --toolchain {{command}}
```

rustup override

Modify directory toolchain overrides.

See **`rustup help toolchain`** for more information about toolchains.

More information: https://rust-lang.github.io/rustup.

- List directiory toolchain overrides:

  ```
  rustup override list
  ```

- Set the override toolchain for the current directory (i.e. tell `rustup` to run `cargo`, `rustc`, etc. from a specific toolchain when in that directory):

  ```
  rustup override set {{toolchain}}
  ```

- Remove the toolchain override for the current directory:

  ```
  rustup override unset
  ```

- Remove all toolchain overrides for directories that no longer exist:

  ```
  rustup override unset --nonexistent
  ```

rustup run

Run a command with an environment configured for a Rust toolchain.

Note: All commands managed by **`rustup`** have a shorthand for this: for example, **`cargo +nightly build`** is equivalent to **`rustup run nightly cargo build`**.

More information: https://rust-lang.github.io/rustup.

- Run a command using a given Rust toolchain (see `rustup help toolchain` for more information):

  ```
  rustup run {{toolchain}} {{command}}
  ```

rustup self

Modify the **`rustup`** installation.

More information: https://rust-lang.github.io/rustup.

- Update `rustup`:

```
rustup self update
```

- Uninstall `rustup`:

```
rustup self uninstall
```

rustup set

Alter **`rustup`** settings.

More information: https://rust-lang.github.io/rustup.

- Set the default host triple:

```
rustup set default-host {{host_triple}}
```

- Set the default profile (`minimal` includes only `rustc`, `rust-std` and `cargo`, whereas `default` adds `rust-docs`, `rustfmt` and `clippy`):

```
rustup set profile {{minimal|default}}
```

- Set whether `rustup` should update itself when running `rustup update`:

```
rustup set auto-self-update {{enable|disable|check-only}}
```

rustup show

Show installed toolchains, targets and the version of **`rustc`**.

More information: https://rust-lang.github.io/rustup.

- Show all information:

```
rustup show
```

- Show the active toolchain:

```
rustup show active-toolchain
```

- Show the rustup data directory:

```
rustup show home
```

rustup target

Modify a toolchain's supported targets.

Without the **`--toolchain`** option **`rustup`** will use the default toolchain. See **`rustup help toolchain`** for more information about toolchains.

More information: https://rust-lang.github.io/rustup.

- Add a target to a toolchain:

```
rustup target add --toolchain {{toolchain}} {{target}}
```

- Remove a target from a toolchain:

```
rustup target remove --toolchain {{toolchain}} {{target}}
```

- List available and installed targets for a toolchain:

```
rustup target list --toolchain {{toolchain}}
```

- List installed targets for a toolchain:

```
rustup target list --toolchain {{toolchain}} --installed
```

rustup toolchain

Manage Rust toolchains.

See **`rustup help toolchain`** for more information about toolchains.

More information: https://rust-lang.github.io/rustup.

- Install or update a given toolchain:

```
rustup toolchain install {{toolchain}}
```

- Uninstall a toolchain:

```
rustup toolchain uninstall {{toolchain}}
```

- List installed toolchains:

```
rustup toolchain list
```

- Create a custom toolchain by symlinking to a directory:

```
rustup toolchain link {{custom_toolchain_name}} {{path/to/
directory}}
```

rustup uninstall

This command is an alias of **`rustup toolchain uninstall`**.

- View documentation for the original command:

```
tldr rustup toolchain
```

rustup update

Update Rust toolchains and **`rustup`** itself (if not installed using a package manager).

More information: https://rust-lang.github.io/rustup.

- Update all installed toolchains and `rustup`:

  ```
  rustup update
  ```

- Install or update a specific toolchain (see `rustup help toolchain` for more information):

  ```
  rustup update {{toolchain}}
  ```

rustup which

Display which binary will be run for a command managed by **`rustup`**.

Like **`which`**, but searches a Rust toolchain instead of **`$PATH`**.

More information: https://rust-lang.github.io/rustup.

- Display the path to the binary in the default toolchain:

  ```
  rustup which {{command}}
  ```

- Display the path to the binary in the specified toolchain (see `rustup help toolchain` for more information):

  ```
  rustup which --toolchain {{toolchain}} {{command}}
  ```

rustup

Install, manage, and update Rust toolchains.

Some subcommands, such as **`toolchain`**, **`target`**, **`update`**, etc. have their own usage documentation.

More information: https://rust-lang.github.io/rustup.

- Install the nightly toolchain for your system:

  `rustup install nightly`

- Switch the default toolchain to nightly so that the `cargo` and `rustc` commands will use it:

  `rustup default nightly`

- Use the nightly toolchain when inside the current project but leave global settings unchanged:

  `rustup override set nightly`

- Update all toolchains:

  `rustup update`

- List installed toolchains:

  `rustup show`

- Run `cargo build` with a certain toolchain:

  `rustup run {{toolchain}} cargo build`

- Open the local Rust documentation in the default web browser:

  `rustup doc`

rvm

Easily installing, managing, and working with multiple ruby environments.

More information: https://rvm.io.

- Install one or more versions of Ruby:

```
rvm install {{version1 version2 ...}}
```

- Display a list of installed versions:

```
rvm list
```

- Use a specific version of Ruby:

```
rvm use {{version}}
```

- Set the default Ruby version:

```
rvm --default use {{version}}
```

- Upgrade a version of Ruby to a new version:

```
rvm upgrade {{current_version}} {{new_version}}
```

- Uninstall a version of Ruby and keep its sources:

```
rvm uninstall {{version}}
```

- Remove a version of Ruby and its sources:

```
rvm remove {{version}}
```

- Show specific dependencies for your OS:

```
rvm requirements
```

s

Web search from the terminal.

More information: https://github.com/zquestz/s.

- Search for a query on Google (default provider):

```
s {{query}}
```

- List all providers:

```
s {{[-l|--list-providers]}}
```

- Search for a query with a given provider:

```
s {{[-p|--provider]}} {{provider}} {{query}}
```

- Use a specified binary to perform the search query:

```
s {{[-b|--binary]}} "{{binary}} {{arguments}}" {{query}}
```

s3cmd

Command line tool and client for uploading, retrieving and managing data in S3 compatible object storage.

More information: https://s3tools.org/s3cmd.

- Invoke configuration/reconfiguration tool:

```
s3cmd --configure
```

- List Buckets/Folders/Objects:

```
s3cmd ls s3://{{bucket|path/to/file}}
```

- Create Bucket/Folder:

```
s3cmd mb s3://{{bucket}}
```

- Download a specific file from a bucket:

```
s3cmd get s3://{{bucket_name}}/{{path/to/file}} {{path/to/local_file}}
```

- Upload a file to a bucket:

```
s3cmd put {{local_file}} s3://{{bucket}}/{{file}}
```

- Move an object to a specific bucket location:

```
s3cmd mv s3://{{src_bucket}}/{{src_object}} s3://{{dst_bucket}}/{{dst_object}}
```

- Delete a specific object:

```
s3cmd rm s3://{{bucket}}/{{object}}
```

safe

Interact with HashiCorp Vault.

More information: https://github.com/starkandwayne/safe.

- Add a safe target:

  ```
  safe target {{vault_addr}} {{target_name}}
  ```

- Authenticate the CLI client against the Vault server, using an authentication token:

  ```
  safe auth {{authentication_token}}
  ```

- Print the environment variables describing the current target:

  ```
  safe env
  ```

- Display a tree hierarchy of all reachable keys for a given path:

  ```
  safe tree {{path}}
  ```

- Move a secret from one path to another:

  ```
  safe move {{path/to/old_secret}} {{path/to/new_secret}}
  ```

- Generate a new 2048-bit SSH key-pair and store it:

  ```
  safe ssh {{2048}} {{path/to/secret}}
  ```

- Set non-sensitive keys for a secret:

  ```
  safe set {{path/to/secret}} {{key}}={{value}}
  ```

- Set auto-generated password in a secret:

  ```
  safe gen {{path/to/secret}} {{key}}
  ```

sails

A realtime enterprise level MVC framework built on top of Node.js.

More information: https://sailsjs.com/documentation/reference/command-line-interface.

- Start Sails:

```
sails lift
```

- Create new Sails project:

```
sails new {{projectName}}
```

- Generate Sails API:

```
sails generate {{name}}
```

- Generate Sails Controller:

```
sails generate controller {{name}}
```

- Generate Sails Model:

```
sails generate model {{name}}
```

salt-call

Invoke salt locally on a salt minion.

More information: https://docs.saltproject.io/en/latest/ref/cli/salt-call.html.

- Perform a highstate on this minion:

```
salt-call state.highstate
```

- Perform a highstate dry-run, compute all changes but don't actually perform them:

```
salt-call state.highstate test=true
```

- Perform a highstate with verbose debugging output:

```
salt-call {{[-l|--log-level]}} debug state.highstate
```

- List this minion's grains:

```
salt-call grains.items
```

salt-key

Manage salt minion keys on the salt master.

Needs to be run on the salt master, likely as root or with sudo.

More information: https://docs.saltproject.io/en/latest/ref/cli/salt-key.html.

- List all accepted, unaccepted and rejected minion keys:

```
salt-key {{[-L|--list-all]}}
```

- Accept a minion key by name:

```
salt-key {{[-a|--accept-all]}} {{MINION_ID}}
```

- Reject a minion key by name:

```
salt-key {{[-r|--reject]}} {{MINION_ID}}
```

- Print fingerprints of all public keys:

```
salt-key {{[-F|--finger-all]}}
```

salt-run

Frontend for executing salt-runners on minions.

More information: https://docs.saltproject.io/en/latest/ref/cli/salt-run.html.

- Show status of all minions:

  ```
  salt-run manage.status
  ```

- Show all minions which are disconnected:

  ```
  salt-run manage.up
  ```

salt

Execute commands and assert state on remote salt minions.

More information: https://docs.saltproject.io/en/latest/ref/cli/index.html.

- List connected minions:

  ```
  salt '*' test.ping
  ```

- Execute a highstate on all connected minions:

  ```
  salt '*' state.highstate
  ```

- Upgrade packages using the OS package manager (apt, yum, brew) on a subset of minions:

  ```
  salt '*.example.com' pkg.upgrade
  ```

- Execute an arbitrary command on a particular minion:

  ```
  salt '{{minion_id}}' cmd.run "ls "
  ```

sam2p

Raster (bitmap) image converter with smart PDF and PostScript (EPS) output.

More information: https://github.com/pts/sam2p.

- Concatenate all PDF files into one:

```
sam2p *.pdf {{path/to/output.pdf}}
```

sambaPipe.py

Exploit CVE-2017-7494 (SambaCry) to upload and load a shared object (SO) file on a vulnerable Samba server for remote code execution.

More information: https://github.com/fortra/impacket.

- Upload and load a shared object file on a vulnerable Samba server:

```
sambaPipe.py -so {{path/to/file.so}} {{domain}}/{{username}}:
{{password}}@{{target}}
```

- Authenticate using NTLM hashes instead of a password:

```
sambaPipe.py -so {{path/to/file.so}} -hashes
{{LM_HASH:NT_HASH}} {{domain}}/{{username}}:{{password}}
@{{target}}
```

- Use Kerberos authentication for the target:

```
sambaPipe.py -so {{path/to/file.so}} -k -no-pass {{domain}}/
{{username}}:{{password}}@{{target}}
```

- Specify a domain controller IP for authentication:

```
sambaPipe.py -so {{path/to/file.so}} -dc-ip {{dc_ip}}
{{domain}}/{{username}}:{{password}}@{{target}}
```

- Use a custom port for the SMB connection:

```
sambaPipe.py -so {{path/to/file.so}} -port {{port}}
{{domain}}/{{username}}:{{password}}@{{target}}
```

samtools

Tools for handling high-throughput sequencing (genomics) data.

Used for reading/writing/editing/indexing/viewing of data in SAM/BAM/CRAM format.

More information: https://www.htslib.org/doc/samtools.html.

- Convert a SAM input file to BAM stream and save to file:

```
samtools view -S {{[-b|--bam]}} {{input.sam}} > {{output.bam}}
```

- Take input from `stdin` (-) and print the SAM header and any reads overlapping a specific region to `stdout`:

```
{{other_command}} | samtools view {{[-h|--with-header]}} - chromosome:start-end
```

- Sort file and save to BAM (the output format is automatically determined from the output file's extension):

```
samtools sort {{input}} {{[-o|--output]}} {{output.bam}}
```

- Index a sorted BAM file (creates `sorted_input.bam.bai`):

```
samtools index {{sorted_input.bam}}
```

- Print alignment statistics about a file:

```
samtools flagstat {{sorted_input}}
```

- Count alignments to each index (chromosome/contig):

```
samtools idxstats {{sorted_indexed_input}}
```

- Merge multiple files:

```
samtools merge {{output}} {{input1 input2 ...}}
```

- Split input file according to read groups:

```
samtools split {{merged_input}}
```

sass

Convert SCSS or Sass files to CSS.

More information: https://sass-lang.com/documentation/cli/dart-sass.

- Convert a SCSS or Sass file to CSS and print out the result:

```
sass {{inputfile.scss|inputfile.sass}}
```

- Convert a SCSS or Sass file to CSS and save the result to a file:

```
sass {{inputfile.scss|inputfile.sass}} {{outputfile.css}}
```

- Watch a SCSS or Sass file for changes and output or update the CSS file with same filename:

```
sass --watch {{inputfile.scss|inputfile.sass}}
```

- Watch a SCSS or Sass file for changes and output or update the CSS file with the given filename:

```
sass --watch {{inputfile.scss|inputfile.sass}}:
{{outputfile.css}}
```

satis

The utility for the Satis static Composer repository.

More information: https://github.com/composer/satis.

- Initialize a Satis configuration:

```
satis init {{satis.json}}
```

- Add a VCS repository to the Satis configuration:

```
satis add {{repository_url}}
```

- Build the static output from the configuration:

```
satis build {{satis.json}} {{path/to/output_directory}}
```

- Build the static output by updating only the specified repository:

```
satis build --repository-url {{repository_url}}
{{satis.json}} {{path/to/output_directory}}
```

- Remove useless archive files:

```
satis purge {{satis.json}} {{path/to/output_directory}}
```

sbcl

High performance Common Lisp compiler.

More information: https://www.sbcl.org/manual/index.html#Command-Line-Options.

- Start a REPL (interactive shell):

```
sbcl
```

- Execute a Lisp script:

```
sbcl --script {{path/to/script.lisp}}
```

sbigtopgm

Convert an SBIG CCDOPS file to PGM.

More information: https://netpbm.sourceforge.net/doc/sbigtopgm.html.

- Convert an SBIG CCDOPS image file to PGM:

```
sbigtopgm {{path/to/input_file.sbig}} > {{path/to/
output.pgm}}
```

sbt

Build tool for Scala and Java projects.

More information: https://www.scala-sbt.org/1.x/docs/.

- Start a REPL (interactive shell):

  ```
  sbt
  ```

- Create a new Scala project from an existing Giter8 template hosted on GitHub:

  ```
  sbt new {{scala/hello-world.g8}}
  ```

- Compile and run all tests:

  ```
  sbt test
  ```

- Delete all generated files in the `target` directory:

  ```
  sbt clean
  ```

- Compile the main sources in `src/main/scala` and `src/main/java` directories:

  ```
  sbt compile
  ```

- Use the specified version of sbt:

  ```
  sbt -sbt-version {{version}}
  ```

- Use a specific jar file as the sbt launcher:

  ```
  sbt -sbt-jar {{path}}
  ```

- List all sbt options:

  ```
  sbt -h
  ```

sbuild

Build a Debian binary package in a clean **`chroot`** environment.

More information: https://wiki.debian.org/sbuild.

- Build the package in the current directory:

```
sbuild
```

- Build the given package:

```
sbuild {{package}}
```

- Build for a certain distribution:

```
sbuild --dist {{distribution}}
```

- Build using custom dependencies (if a directory is passed, all files ending with `.deb` are used):

```
sbuild --extra-package {{path/to/file_or_directory}}
```

- Run a shell in case of build failure to further investigate:

```
sbuild --build-failed-commands=%SBUILD_SHELL
```

- Cross build for a certain architecture:

```
sbuild --host {{architecture}}
```

- Build for the given native architecture:

```
sbuild --arch {{architecture}}
```

sc_analysis_dump

Dump of traceroute data in an easily parsed format.

More information: https://www.caida.org/catalog/software/scamper/.

- Output the traceroute of `warts` files one after the other in an easy-to-parse format:

```
sc_analysis_dump {{path/to/file1.warts path/to/
file2.warts ...}}
```

sc_tracediff

Display traceroute paths where the path has changed.

More information: https://www.caida.org/catalog/software/scamper/.

- Show the difference between traceroutes in two `warts` files:

```
sc_tracediff {{path/to/file1.warts}} {{path/to/file2.warts}}
```

- Show the difference between the traceroutes in two `warts` files, including those that have not changed:

```
sc_tracediff -a {{path/to/file1.warts}} {{path/to/
file2.warts}}
```

- Show the difference between the traceroutes in two `warts` files and try to show DNS names and not IP addresses if possible:

```
sc_tracediff -n {{path/to/file1.warts}} {{path/to/
file2.warts}}
```

sc_ttlexp

Dump source addresses from ICMP TTL expired messages in **`warts`** files.

More information: https://www.caida.org/catalog/software/scamper/.

- Output the source address of ICMP TTL expire messages in `warts` files one after the other:

```
sc_ttlexp {{path/to/file1.warts path/to/file2.warts ...}}
```

sc_warts2csv

Dump traceroutes collected by **`scamper`** in CSV format.

More information: https://www.caida.org/catalog/software/scamper/.

- Convert traceroute data in `warts` files to CSV and output it:

```
sc_warts2csv {{path/to/file1.warts path/to/file2.warts ...}}
```

sc_warts2json

JSON dump of information contained in a **`warts`** file.

More information: https://www.caida.org/catalog/software/scamper/.

- Convert `warts` files to JSON and output the result:

```
sc_warts2json {{path/to/file1.warts path/to/file2.warts ...}}
```

sc_warts2pcap

Write packets included in **`warts`** object to a PCAP file.

This is only possible for tbit, sting and sniff.

More information: https://www.caida.org/catalog/software/scamper/.

- Convert the data from several `warts` files into one PCAP file:

```
sc_warts2pcap -o {{path/to/output.pcap}} {{path/to/
file1.warts path/to/file2.warts ...}}
```

- Convert the data from a `warts` file into a PCAP file and sort the packets by timestamp:

```
sc_warts2pcap -s -o {{path/to/output.pcap}} {{path/to/
file.warts}}
```

sc_warts2text

Simple dump of information contained in a **`warts`** file.

More information: https://www.caida.org/catalog/software/scamper/.

- Output the information in `warts` files as text:

```
sc_warts2text {{path/to/file1.warts path/to/file2.warts ...}}
```

sc_wartscat

Concatenate **`warts`** files.

More information: https://www.caida.org/catalog/software/scamper/.

- Concatenate `warts` files into one:

```
sc_wartscat -o {{path/to/output.warts}} {{path/to/file1.warts
path/to/file2.warts ...}}
```

sc_wartsdump

Verbose dump of information contained in a **`warts`** file.

More information: https://www.caida.org/catalog/software/scamper/.

- Output the content of `warts` files verbose:

```
sc_wartsdump {{path/to/file1.warts path/to/file2.warts ...}}
```

sc_wartsfilter

Select specific records from a **`warts`** file.

More information: https://www.caida.org/catalog/software/scamper/.

- Filter all data records that had specific destinations and write them to a separate file:

```
sc_wartsfilter -i {{path/to/input.warts}} -o {{path/to/
output.warts}} -a {{192.0.2.5}} -a {{192.0.2.6}}
```

- Filter all records that had certain destinations in a prefix and write them to a separate file:

```
sc_wartsfilter -i {{path/to/input.warts}} -o {{path/to/
output.warts}} -a {{2001:db8::/32}}
```

- Filter all records that using a specific action and output them as JSON:

```
sc_wartsfilter -i {{path/to/input.warts}} -t {{ping}} |
sc_warts2json
```

sc_wartsfix

Truncate damaged **warts** files.

More information: https://www.caida.org/catalog/software/scamper/.

- Save all records (in a separate file) up to the last intact one:

```
sc_wartsfix {{path/to/file1.warts path/to/file2.warts ...}}
```

scala-cli

Interact with the Scala programming language.

More information: https://scala-cli.virtuslab.org/docs/overview/.

- Start a REPL (interactive shell) using a specific Scala and JVM version:

```
scala-cli --scala {{3.1.0}} --jvm {{temurin:17}}
```

- Compile and run a Scala script:

```
scala-cli run {{path/to/script.scala}}
```

- Compile and test a Scala script:

```
scala-cli test {{path/to/script.scala}}
```

- Format a Scala script, updating the file in-place:

```
scala-cli fmt {{path/to/script.scala}}
```

- Generate files for IDE (VSCode and IntelliJ) support:

```
scala-cli setup-ide {{path/to/script.scala}}
```

scala

Scala application launcher and interactive interpreter.

More information: https://scala-lang.org.

- Start a REPL (interactive shell):

```
scala
```

- Start the interpreter with a dependency in the classpath:

```
scala -classpath {{filename.jar}} {{command}}
```

- Execute a Scala script:

```
scala {{script.scala}}
```

- Execute a `.jar` program:

```
scala {{filename.jar}}
```

- Execute a single Scala command in the command-line:

```
scala -e {{command}}
```

scalafmt

Code formatter for Scala.

Configurations are stored in the **`.scalafmt.conf`** file.

More information: https://scalameta.org/scalafmt.

- Reformat all `.scala` files in the current directory recursively:

  ```
  scalafmt
  ```

- Reformat specific files or directories with a custom formatting configuration:

  ```
  scalafmt --config {{path/to/.scalafmt.conf}} {{path/to/file_or_directory1 path/to/file_or_directory2 ...}}
  ```

- Check if files are correctly formatted, returning `0` if all files respect the formatting style:

  ```
  scalafmt --config {{path/to/.scalafmt.conf}} --test
  ```

- Exclude files or directories:

  ```
  scalafmt --exclude {{path/to/file_or_directory1 path/to/file_or_directory2 ...}}
  ```

- Format only files that were edited against the current Git branch:

  ```
  scalafmt --config {{path/to/.scalafmt.conf}} --mode diff
  ```

scamper

Actively probes the Internet in order to analyze topology and performance.

Includes some tools that start with **`sc_`**, for example **`sc_warts2text`** or **`sc_ttlexp`**.

More information: https://www.caida.org/catalog/software/scamper/.

- Execute the standard option (traceroute) to a destination:

```
scamper -i {{192.0.2.1}}
```

- Execute two actions (ping and traceroute) on two different targets:

```
scamper -I "{{ping}} {{192.0.2.1}}" -I "{{trace}}
{{192.0.2.2}}"
```

- Ping several hosts with UDP, use a specific port number for the first ping and increase it for each subsequent ping:

```
scamper -c "{{ping}} -P {{UDP-dport}} -d {{33434}}" -i
{{192.0.2.1}} -i {{192.0.2.2}}
```

- Use the Multipath Discovery Algorithm (MDA) to determine the presence of load-balanced paths to the destination and use ICMP echo packets to sample with a maximum of three attempts, write the result to a `warts` file:

```
scamper -O {{warts}} -o {{path/to/output.warts}} -I
"{{tracelb}} -P {{ICMP-echo}} -q {{3}} {{192.0.2.1}}"
```

- Execute a Paris traceroute with ICMP to a destination and save the result in a compressed `warts` file:

```
scamper -O {{warts.gz}} -o {{path/to/output.warts}} -I
"{{trace}} -P {{icmp-paris}} {{2001:db8:dead:beaf::4}}"
```

- Record all ICMP packets that arrive at a specific IP address and have a specific ICMP ID in a `warts` file:

```
scamper -O {{warts}} -o {{path/to/output.warts}} -I "sniff -S
{{2001:db8:dead:beef::6}} icmp[icmpid] == {{101}}"
```

scan-build

Run a static analyzer over a codebase as part of performing a regular build.

More information: https://clang-analyzer.llvm.org/scan-build.html.

- Build and analyze the project in the current directory:

```
scan-build {{make}}
```

- Run a command and pass all subsequent options to it:

```
scan-build {{command}} {{command_arguments}}
```

- Display help:

```
scan-build
```

scc

Count lines of code. Written in Go.

More information: https://github.com/boyter/scc.

- Print lines of code in the current directory:

```
scc
```

- Print lines of code in the target directory:

```
scc {{path/to/directory}}
```

- Display output for every file:

```
scc --by-file
```

- Display output using a specific output format (defaults to `tabular`):

```
scc {{[-f|--format]}} {{tabular|wide|json|csv|cloc-yaml|html|
html-table}}
```

- Only count files with specific file extensions:

```
scc {{[-i|--include-ext]}} {{go,java,js}}
```

- Exclude directories from being counted:

```
scc --exclude-dir {{.git,.hg}}
```

- Display output and sort by column (defaults to by files):

```
scc {{[-s|--sort]}} {{files|name|lines|blanks|code|comments|
complexity}}
```

- Display help:

```
scc {{[-h|--help]}}
```

sccache

A fast C/C++/Rust compiler cache.

Composed of a client and a server, both running on the machine.

More information: https://manned.org/sccache.

- Show compilation statistics:

```
sccache {{[-s|--show-stats]}}
```

- Run `gcc` (or any compiler command) through `sccache`:

```
sccache gcc {{path/to/file.c}}
```

- Start `sccache` server in the foreground and print logs:

```
sccache --stop-server; SCCACHE_LOG=trace
SCCACHE_START_SERVER=1 SCCACHE_NO_DAEMON=1 sccache
```

- Ask scheduler for distributed compilation status:

```
sccache --dist-status
```

sccmap

Extract strongly connected components of directed graphs.

Graphviz filters: **`acyclic`**, **`bcomps`**, **`comps`**, **`edgepaint`**, **`gvcolor`**, **`gvpack`**, **`mingle`**, **`nop`**, **`sccmap`**, **`tred`**, & **`unflatten`**.

More information: https://www.graphviz.org/pdf/sccmap.1.pdf.

- Extract strongly connected components of one or more directed graphs:

```
sccmap -S {{path/to/input1.gv path/to/input2.gv ...}} > {{path/to/output.gv}}
```

- Print statistics about a graph, producing no output graph:

```
sccmap -v -s {{path/to/input1.gv path/to/input2.gv ...}}
```

- Display help:

```
sccmap -?
```

scd

File manager focused on shell integration.

More information: https://github.com/cshuaimin/scd.

- Index paths recursively for the very first run:

```
scd -ar {{path/to/directory}}
```

- Change to a specific directory:

```
scd {{path/to/directory}}
```

- Change to a path matching specific patterns:

```
scd "{{pattern1 pattern2 ...}}"
```

- Show selection menu and ranking of 20 most likely directories:

```
scd -v
```

- Add a specific alias for the current directory:

```
scd --alias={{word}}
```

- Change to a directory using a specific alias:

```
scd {{word}}
```

scheme

MIT Scheme language interpreter and REPL (interactive shell).

More information: https://www.gnu.org/software/mit-scheme/documentation/stable/mit-scheme-user.html#Command_002dLine-Options.

- Start a REPL (interactive shell):

```
scheme
```

- Run a scheme program (with no REPL output):

```
scheme < {{script.scm}} --quiet
```

- Load a scheme program into the REPL:

```
scheme --load {{script.scm}}
```

- Load scheme expressions into the REPL:

```
scheme --eval "{{(define foo 'x)}}"
```

- Open the REPL in quiet mode:

```
scheme --quiet
```

scim

A curses based, vim-like spreadsheet calculator.

Use **<h><j><k><l>** or **<ArrowKeys>** to navigate.

More information: https://github.com/andmarti1424/sc-im.

- Start SC-IM:

  `scim {{path/to/file.csv}}`

- Enter a string into the current cell:

  `{{<<>|<>>}}`

- Enter a numeric constant into the current cell:

  `<=>`

- Edit string in the current cell:

  `<E>`

- Edit number in the current cell:

  `<e>`

- Center align the current cell:

  `<|>`

scp

Secure copy.

Copy files between hosts using Secure Copy Protocol over SSH.

More information: https://man.openbsd.org/scp.

- Copy a local file to a remote host:

```
scp {{path/to/local_file}} {{remote_host}}:{{path/to/remote_file}}
```

- Use a specific port when connecting to the remote host:

```
scp -P {{port}} {{path/to/local_file}} {{remote_host}}:{{path/to/remote_file}}
```

- Copy a file from a remote host to a local directory:

```
scp {{remote_host}}:{{path/to/remote_file}} {{path/to/local_directory}}
```

- Recursively copy the contents of a directory from a remote host to a local directory:

```
scp -r {{remote_host}}:{{path/to/remote_directory}} {{path/to/local_directory}}
```

- Copy a file between two remote hosts transferring through the local host:

```
scp -3 {{host1}}:{{path/to/remote_file}} {{host2}}:{{path/to/remote_directory}}
```

- Use a specific username when connecting to the remote host:

```
scp {{path/to/local_file}} {{remote_username}}@{{remote_host}}:{{path/to/remote_directory}}
```

- Use a specific SSH private key for authentication with the remote host:

```
scp -i {{~/.ssh/private_key}} {{path/to/local_file}} {{remote_host}}:{{path/to/remote_file}}
```

- Use a specific proxy when connecting to the remote host:

```
scp -J {{proxy_username}}@{{proxy_host}} {{path/to/local_file}} {{remote_host}}:{{path/to/remote_file}}
```

scrapy

Web-crawling framework.

More information: https://docs.scrapy.org/en/latest/topics/commands.html#using-the-scrapy-tool.

- Create a project:

```
scrapy startproject {{project_name}}
```

- Create a spider (in project directory):

```
scrapy genspider {{spider_name}} {{website_domain}}
```

- Edit spider (in project directory):

```
scrapy edit {{spider_name}}
```

- Run spider (in project directory):

```
scrapy crawl {{spider_name}}
```

- Fetch a webpage as Scrapy sees it and print the source to `stdout`:

```
scrapy fetch {{url}}
```

- Open a webpage in the default browser as Scrapy sees it (disable JavaScript for extra fidelity):

```
scrapy view {{url}}
```

- Open Scrapy shell for URL, which allows interaction with the page source in a Python shell (or IPython if available):

```
scrapy shell {{url}}
```

scrcpy

Display and control your Android device on a desktop.

More information: https://github.com/Genymobile/scrcpy.

- Display a mirror of a connected device:

```
scrcpy
```

- Turn the device screen off and prevent it from sleeping while mirroring:

```
scrcpy {{[-S|--turn-screen-off]}} {{[-w|--stay-awake]}}
```

- Display a mirror of a specific device based on its ID or IP address (find it under the `adb devices` command):

```
scrcpy {{[-s|--serial]}} {{0123456789abcdef|
192.168.0.1:5555}}
```

- Start display in fullscreen mode:

```
scrcpy {{[-f|--fullscreen]}}
```

- Show touches on physical device:

```
scrcpy {{[-t|--show-touches]}}
```

- Record display screen:

```
scrcpy {{[-r|--record]}} {{path/to/file.mp4}}
```

- Specify the target directory for pushing files to device by drag and drop (non-APK):

```
scrcpy --push-target {{path/to/directory}}
```

screen

Hold a session open on a remote server. Manage multiple windows with a single SSH connection.

See also: **`tmux`**, **`zellij`**.

More information: https://manned.org/screen.

- Start a new screen session:

```
screen
```

- Start a new named screen session:

```
screen -S {{session_name}}
```

- Start a new daemon and log the output to `screenlog.x`:

```
screen -dmLS {{session_name}} {{command}}
```

- Show open screen sessions:

```
screen -ls
```

- Reattach to an open screen:

```
screen -r {{session_name}}
```

- Detach from inside a screen:

```
<Ctrl a><d>
```

- Kill the current screen session:

```
<Ctrl a><k>
```

- Kill a detached screen:

```
screen -X -S {{session_name}} quit
```

screenfetch

Display system information.

More information: https://github.com/KittyKatt/screenFetch.

- Start `screenfetch`:

```
screenfetch
```

- Take a screenshot (requires `scrot`):

```
screenfetch -s
```

- Specify distribution logo:

```
screenfetch -A '{{distribution_name}}'
```

- Specify distribution logo and text:

```
screenfetch -D '{{distribution_name}}'
```

- Strip all color:

```
screenfetch -N
```

sd

Intuitive find and replace.

More information: https://github.com/chmln/sd.

- Trim some whitespace using a `regex` (output stream: `stdout`):

```
{{echo 'lorem ipsum 23 '}} | sd '\s+$' ''
```

- Replace words using capture groups (output stream: `stdout`):

```
{{echo 'cargo +nightly watch'}} | sd '(\w+)\s+\+(\w+)\s+
(\w+)' 'cmd: $1, channel: $2, subcmd: $3'
```

- Find and replace in a specific file (output stream: `stdout`):

```
sd {{[-p|--preview]}} '{{window.fetch}}' '{{fetch}}' {{path/
to/file.js}}
```

- Find and replace in all files in the current project (output stream: `stdout`):

```
sd '{{from "react"}}' '{{from "preact"}}' "$(find . -type f)"
```

sdcv

StarDict, a dictionary client.

Dictionaries are provided separately from the client.

More information: https://manned.org/sdcv.

- Start `sdcv` interactively:

```
sdcv
```

- List installed dictionaries:

```
sdcv --list-dicts
```

- Display a definition from a specific dictionary:

```
sdcv --use-dict {{dictionary_name}} {{search_term}}
```

- Look up a definition with a fuzzy search:

```
sdcv {{search_term}}
```

- Look up a definition with an exact search:

```
sdcv --exact-search {{search_term}}
```

- Look up a definition and format the output as JSON:

```
sdcv --json {{search_term}}
```

- Search for dictionaries in a specific directory:

```
sdcv --data-dir {{path/to/directory}} {{search_term}}
```

sdiff

Compare the differences between and optionally merge 2 files.

More information: https://manned.org/sdiff.

- Compare 2 files:

```
sdiff {{path/to/file1}} {{path/to/file2}}
```

- Compare 2 files, ignoring all tabs and whitespace:

```
sdiff {{[-W|--ignore-all-space]}} {{path/to/file1}} {{path/to/file2}}
```

- Compare 2 files, ignoring whitespace at the end of lines:

```
sdiff {{[-Z|--ignore-trailing-space]}} {{path/to/file1}} {{path/to/file2}}
```

- Compare 2 files in a case-insensitive manner:

```
sdiff {{[-i|--ignore-case]}} {{path/to/file1}} {{path/to/file2}}
```

- Compare and then merge, writing the output to a new file:

```
sdiff {{[-o|--output]}} {{path/to/merged_file}} {{path/to/file1}} {{path/to/file2}}
```

sdk

Manage parallel versions of multiple Software Development Kits.

Supports Java, Groovy, Scala, Kotlin, Gradle, Maven, Vert.x and many others.

More information: https://sdkman.io/usage.

- Install an SDK version:

```
sdk install {{sdk_name}} {{sdk_version}}
```

- Use a specific SDK version for the current terminal session:

```
sdk use {{sdk_name}} {{sdk_version}}
```

- Show the stable version of any available SDK:

```
sdk current {{sdk_name}}
```

- Show the stable versions of all installed SDKs:

```
sdk current
```

- List all available SDKs:

```
sdk list
```

- List all versions of an SDK:

```
sdk list {{sdk_name}}
```

- Upgrade an SDK to the latest stable version:

```
sdk upgrade {{sdk_name}}
```

- Uninstall a specific SDK version:

```
sdk rm {{sdk_name}} {{sdk_version}}
```

sdkmanager

Install packages for the Android SDK.

More information: https://developer.android.com/tools/sdkmanager.

- List available packages:

```
sdkmanager --list
```

- Install a package:

```
sdkmanager {{package}}
```

- Update every installed package:

```
sdkmanager --update
```

- Uninstall a package:

```
sdkmanager --uninstall {{package}}
```

searchsploit

Search Exploit Database for exploits, shellcodes and/or papers.

If known version numbers are used as search terms, exploits for both the exact version and others whose version range covers the one specified are shown.

More information: https://www.exploit-db.com/searchsploit.

- Search for an exploit, shellcode, or paper:

```
searchsploit {{search_terms}}
```

- Search for a known specific version, e.g. sudo version 1.8.27:

```
searchsploit sudo 1.8.27
```

- Show the exploit-db link to the found resources:

```
searchsploit {{[-w|--www]}} {{search_terms}}
```

- Copy the resource to the current directory (requires the number of the exploit):

```
searchsploit {{[-m|--mirror]}} {{exploit_number}}
```

- Examine the resource, using the pager defined in the `$PAGER` environment variable:

```
searchsploit {{[-x|--examine]}} {{exploit_number}}
```

- Update the local Exploit Database:

```
searchsploit {{[-u|--update]}}
```

- Search for the [c]ommon [v]ulnerabilities and [e]xposures (CVE) value:

```
searchsploit --cve {{2021-44228}}
```

- Check results in `nmap`'s XML output with service version (`nmap -sV -oX nmap-output.xml`) for known exploits:

```
searchsploit --nmap {{path/to/nmap-output.xml}}
```

secrethub

Keep secrets out of configuration files.

More information: https://github.com/secrethub/secrethub-cli.

- Print a secret to `stdout`:

```
secrethub read {{path/to/secret}}
```

- Generate a random value and store it as a new or updated secret:

```
secrethub generate {{path/to/secret}}
```

- Store a value from the clipboard as a new or updated secret:

```
secrethub write --clip {{path/to/secret}}
```

- Store a value supplied on `stdin` as a new or updated secret:

```
echo "{{secret_value}}" | secrethub write {{path/to/secret}}
```

- Audit a repository or secret:

```
secrethub audit {{path/to/repo_or_secret}}
```

secretsdump.py

Dump NTLM hashes, plaintext passwords, and domain credentials from remote Windows systems.

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- Dump credentials from a Windows machine using a username and password:

```
secretsdump.py {{domain}}/{{username}}:{{password}}
@{{target}}
```

- Dump hashes from a machine using pass-the-hash authentication:

```
secretsdump.py -hashes {{LM_Hash}}:{{NT_Hash}} {{domain}}/
{{username}}@{{target}}
```

- Dump credentials from Active Directory's NTDS.dit file:

```
secretsdump.py -just-dc {{domain}}/{{username}}:{{password}}
@{{target}}
```

- Extract credentials from a local SAM database using registry hives:

```
secretsdump.py -sam {{path/to/SAM}} -system {{path/to/
SYSTEM}}
```

- Dump hashes from a machine without providing a password (if a valid authentication session exists, e.g. via Kerberos or NTLM SSO):

```
secretsdump.py -no-pass {{domain}}/{{username}}@{{target}}
```

security-checker

Check if a PHP application uses dependencies with known security vulnerabilities.

More information: https://github.com/sensiolabs/security-checker.

- Look for security issues in the project dependencies (based on the `composer.lock` file in the current directory):

  ```
  security-checker security:check
  ```

- Use a specific `composer.lock` file:

  ```
  security-checker security:check {{path/to/composer.lock}}
  ```

- Return results as a JSON object:

  ```
  security-checker security:check --format=json
  ```

sed

Edit text in a scriptable manner.

See also: **awk**, **ed**.

More information: https://manned.org/sed.1posix.

- Replace all `apple` (basic `regex`) occurrences with `mango` (basic `regex`) in all input lines and print the result to `stdout`:

```
{{command}} | sed 's/apple/mango/g'
```

- Execute a specific script [f]ile and print the result to `stdout`:

```
{{command}} | sed -f {{path/to/script.sed}}
```

- Print just a first line to `stdout`:

```
{{command}} | sed -n '1p'
```

select

Bash builtin construct for creating menus.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-select.

- Create a menu out of individual words:

```
select {{word}} in {{apple orange pear banana}}; do echo ${{word}}; done
```

- Create a menu from the output of another command:

```
select {{line}} in $({{command}}); do echo ${{line}}; done
```

- Specify the prompt string for `select` and create a menu for picking a file or folder from the current directory:

```
PS3="{{Select a file: }}"; select {{file}} in *; do echo ${{file}}; done
```

- Create a menu from a Bash array:

```
{{fruits}}=({{apple orange pear banana}}); select {{word}} in ${{{fruits[@]}}}; do echo ${{word}}; done
```

semver

Semantic version string parser.

More information: https://github.com/npm/node-semver.

- Check if a version string respects the semantic versioning format (prints an empty string if it does not match):

```
semver {{1.2}}
```

- Convert a version string to the semantic versioning format:

```
semver --coerce {{1.2}}
```

- Test if `1.2.3` matches the `^1.0` range (prints an empty string if it does not match):

```
semver {{1.2.3}} --range "{{^1.0}}"
```

- Test with multiple ranges:

```
semver {{1.2.3}} --range "{{>=1.0}}" "{{<2.0}}"
```

- Test multiple version strings and return only the ones that match:

```
semver {{1.2.3}} {{2.0.0}} --range "{{^1.0}}"
```

sendmail

Send email.

More information: https://manned.org/sendmail.

- Send a message with the content of `message.txt` to the mail directory of local user `username`:

```
sendmail < {{message.txt}} {{username}}
```

- Send an email from `sender@example.com` (assuming the mail server is configured for this) to `receiver@example.com` containing the message in `message.txt`:

```
sendmail < message.txt -f sender@example.com receiver@example.com
```

- Send an email from `sender@example.com` (assuming the mail server is configured for this) to `receiver@example.com` containing the file `file.zip`:

```
sendmail < file.zip -f sender@example.com receiver@example.com
```

seq

Output a sequence of numbers to **`stdout`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/seq-invocation.html.

- Sequence from 1 to 10:

```
seq 10
```

- Every 3rd number from 5 to 20:

```
seq 5 3 20
```

- Separate the output with a space instead of a newline:

```
seq {{[-s|--separator]}} " " 5 3 20
```

- Format output width to a minimum of 4 digits padding with zeros as necessary:

```
seq {{[-f|--format]}} "%04g" 5 3 20
```

sequelize

Promise-based Node.js ORM for Postgres, MySQL, MariaDB, SQLite and Microsoft SQL Server.

More information: https://sequelize.org/docs/v7/cli/.

- Create a model with 3 fields and a migration file:

```
sequelize model:generate --name {{table_name}} --attributes {{field1:integer,field2:string,field3:boolean}}
```

- Run the migration file:

```
sequelize db:migrate
```

- Revert all migrations:

```
sequelize db:migrate:undo:all
```

- Create a seed file with the specified name to populate the database:

```
sequelize seed:generate --name {{seed_filename}}
```

- Populate database using all seed files:

```
sequelize db:seed:all
```

sequin

Make ANSI escape sequences human-readable.

Useful for debugging, learning, or inspecting terminal output.

More information: https://github.com/charmbracelet/sequin.

- Describe ANSI escape sequences from a string:

```
printf "{{\x1b[38;5;4mCiao, \x1b[1;7mBaby.\x1b[0m\n}}" | sequin
```

- Inspect colorized output from another command (e.g., `ls`):

```
ls -l --color=always | sequin
```

- Examine a file containing ANSI sequences (e.g., a TUI golden file):

```
cat {{path/to/file.golden}} | sequin
```

- Execute a command directly within a fake TTY to inspect its output:

```
sequin -- {{ls -l go.*}}
```

- Highlight raw ANSI sequences inline for easier reading:

```
git -c status.color=always status -sb | sequin -r
```

serialver

Returns the serialVersionUID of classes.

It does not set a security manager by default.

More information: https://docs.oracle.com/en/java/javase/20/docs/specs/man/serialver.html.

- Display the serialVersionUID of a class:

```
serialver {{classnames}}
```

- Display the serialVersionUID for a colon-separated list of classes and resources:

```
serialver -classpath {{path/to/directory}} {{classname1:classname2:...}}
```

- Use a specific option from reference page of Java application launcher to the Java Virtual Machine:

```
serialver -Joption {{classnames}}
```

serve

Static file serving and directory listing.

More information: https://github.com/vercel/serve.

- Start an HTTP server listening on the default port to serve the current directory:

```
serve
```

- Start an HTTP server on a specific [p]ort to serve a specific directory:

```
serve -p {{port}} {{path/to/directory}}
```

- Start an HTTP server with CORS enabled by including the `Access-Control-Allow-Origin: *` header in all responses:

```
serve {{[-C|--cors]}}
```

- Start an HTTP server on the default port rewriting all not-found requests to the `index.html` file:

```
serve {{[-s|--single]}}
```

- Start an HTTPS server on the default port using the specified certificate:

```
serve --ssl-cert {{path/to/cert.pem}} --ssl-key {{path/to/key.pem}}
```

- Start an HTTP server on the default port using a specific configuration file:

```
serve {{[-c|--config]}} {{path/to/serve.json}}
```

- Display help:

```
serve --help
```

serverless

Toolkit for deploying and operating serverless architectures on AWS, Google Cloud, Azure and IBM OpenWhisk.

Commands can be run either using the **`serverless`** command or its alias, **`sls`**.

More information: https://www.serverless.com/framework/docs/providers/aws/cli-reference.

- Create a serverless project:

  ```
  serverless create
  ```

- Create a serverless project from a template:

  ```
  serverless create --template {{template_name}}
  ```

- Deploy to a cloud provider:

  ```
  serverless deploy
  ```

- Display information about a serverless project:

  ```
  serverless info
  ```

- Invoke a deployed function:

  ```
  serverless invoke -f {{function_name}}
  ```

- Follow the logs for a project:

  ```
  serverless logs {{[-t|--tail]}}
  ```

Set-NodeInstallLocation

Set the default Node.js installation directory for **`ps-nvm`**.

This command is part of **`ps-nvm`** and can only be run under PowerShell.

More information: https://github.com/aaronpowell/ps-nvm.

- Change the Node.js install location to a specified directory (`ps-nvm` will create a new `.nvm` subdirectory to install them):

```
Set-NodeInstallLocation {{path/to/directory}}
```

Set-NodeVersion

Set the default Node.js version for **`ps-nvm`**.

Part of **`ps-nvm`** and can only be run under PowerShell.

More information: https://github.com/aaronpowell/ps-nvm.

- Use a specific version of Node.js in the current PowerShell session:

  ```
  Set-NodeVersion {{node_version}}
  ```

- Use the latest installed Node.js version 20.x:

  ```
  Set-NodeVersion ^20
  ```

- Set the default Node.js version for the current user (only applies to future PowerShell sessions):

  ```
  Set-NodeVersion {{node_version}} -Persist User
  ```

- Set the default Node.js version for all users (must be run as Administrator/root and only applies to future PowerShell sessions):

  ```
  Set-NodeVersion {{node_version}} -Persist Machine
  ```

set

Toggle shell options or set the values of positional parameters.

More information: https://www.gnu.org/software/bash/manual/bash.html#The-Set-Builtin.

- Display the names and values of shell variables:

```
set
```

- Export newly initialized variables to child processes:

```
set -a
```

- Write formatted messages to `stderr` when jobs finish:

```
set -b
```

- Write and edit text in the command-line with `vi`-like keybindings (e.g. `yy`):

```
set -o {{vi}}
```

- Return to default mode:

```
set -o {{emacs}}
```

- List all modes:

```
set -o
```

- Exit the shell when (some) command fails:

```
set -e
```

- Reset all shell parameters and assign new ones:

```
set -- {{argument1 argument2 ...}}
```

sf

A powerful command-line interface that simplifies development and build automation when working with your Salesforce org.

More information: https://developer.salesforce.com/tools/salesforcecli.

- Authorize a Salesforce Organization:

```
sf force:auth:web:login --setalias {{organization}} --instanceurl {{organization_url}}
```

- List all authorized organizations:

```
sf force:org:list
```

- Open a specific organization in the default web browser:

```
sf force:org:open --targetusername {{organization}}
```

- Display information about a specific organization:

```
sf force:org:display --targetusername {{organization}}
```

- Push source metadata to an Organization:

```
sf force:source:push --targetusername {{organization}}
```

- Pull source metadata from an Organization:

```
sf force:source:pull --targetusername {{organization}}
```

- Generate a password for the organization's logged-in user:

```
sf force:user:password:generate --targetusername {{organization}}
```

- Assign a permission set for the organization's logged-in user:

```
sf force:user:permset:assign --permsetname {{permission_set_name}} --targetusername {{organization}}
```

sfdk apply

Apply patches from RPM SPEC file.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/20-building-mb2/doc/command.apply.adoc.

- Apply all patches:

```
sfdk apply
```

- Reverse apply all patches:

```
sfdk apply -R
```

sfdk build-init

Initialize build directory.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/20-building-mb2/doc/command.build-init.adoc.

- Initialize the current directory as the build directory:

```
sfdk build-init
```

- Initialize the specified directory as the build directory:

```
sfdk build-init {{directory}}
```

sfdk build-requires

Update build time dependencies.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/20-building-mb2/doc/command.build-requires.adoc.

- Run a subcommand refreshing the cache:

```
sfdk build-requires --refresh {{subcommand}}
```

- Run a subcommand without refreshing the cache:

```
sfdk build-requires --no-refresh {{subcommand}}
```

- Install or update the build-time dependencies:

```
sfdk build-requires pull
```

- Install or update the build-time dependencies, omitting all extra ones:

```
sfdk build-requires reset
```

- Show the difference between current and clean build environments:

```
sfdk build-requires diff
```

sfdk build-shell

Execute custom steps in build engine.

See also: **`sfdk config`** for configuring the build target and **`sfdk build-init`** for initializing build tree.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/20-building-mb2/doc/command.build-shell.adoc.

- Launch interactive shell in the build engine:

  ```
  sfdk build-shell
  ```

- Run a specified command in the build shell:

  ```
  sfdk build-shell {{command}}
  ```

- Launch interactive shell in the build engine in maintenance mode, when inspecting or modifying the build environment:

  ```
  sfdk build-shell --maintain
  ```

sfdk check

Perform quality checks.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/20-building-mb2/doc/command.cmake.adoc.

- Display test suites:

```
sfdk check --list-suites
```

- Run all or essential test suites:

```
sfdk check
```

- Add testing level to the check:

```
sfdk check {{[-l|--levels]}} +{{level}}
```

- Remove testing level from the check:

```
sfdk check {{[-l|--levels]}} -{{level}}
```

- Add testing suite to the check:

```
sfdk check {{[-s|--suites]}} +{{suite}}
```

- Remove testing suite from the check:

```
sfdk check {{[-s|--suites]}} -{{suite}}
```

sfdk cmake

Execute cmake build step.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/20-building-mb2/doc/command.cmake.adoc.

- Run cmake:

```
sfdk cmake
```

- Run cmake in specified project directory:

```
sfdk cmake {{project}}
```

- Run cmake with extra arguments:

```
sfdk cmake -- {{arguments}}
```

- Run cmake build in current directory:

```
sfdk cmake --build .
```

- Run cmake build in current directory with extra cmake arguments:

```
sfdk cmake --build . {{cmake-arguments}}
```

- Run cmake build in current directory with extra build tool arguments:

```
sfdk cmake --build . -- {{build-tool-arguments}}
```

sfdk compiledb

Generate compilation database.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/70-ide-compiledb/doc/command.compiledb.adoc.

- Generate compilation database:

```
sfdk compiledb
```

- Generate compilation database with extra `make` arguments:

```
sfdk compiledb {{arguments}}
```

sfdk config

Configure sfdk.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/10-general/doc/command.config.adoc.

- Show configuration in all scopes:

```
sfdk config --show
```

- Set a configuration value:

```
sfdk config {{name}}={{value}}
```

- Mask an option as empty:

```
sfdk config {{name}}=
```

- Mask an option as empty without pushing it at the inner scope:

```
sfdk config {{name}}
```

- Clear option value:

```
sfdk --drop {{name}}
```

- Run subcommand in specified scope (`global`, `session` or `command`):

```
sfdk config --{{scope}} {{subcommand}}
```

sfdk deploy

Deploy build results to a device.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/50-testing-mb2/doc/command.deploy.adoc.

- Deploy using a specified method (`pkcon`, `rsync`, `sdk`, `zypper`, `zypper-dup` or `manual`):

```
sfdk deploy --{{method}}
```

- Preview deploy without applying the changes:

```
sfdk deploy --{{method}} {{[-n|--dry-run]}}
```

- Deploy files in glob pattern `package*`:

```
sfdk deploy --{{method}} "+package*"
```

- Deploy all files excluding `ignore*`:

```
sfdk deploy --{{method}} "-ignore*"
```

- Undeploy using a specified method (`pkcon`, `rpm`, `rsync`, `sdk` or `zypper`):

```
sfdk undeploy --{{method}}
```

sfdk device

Maintain and control registered devices.

More information: https://docs.sailfishos.org/Develop/Apps/Tutorials/Building_packages_-_advanced_techniques/#running-the-sample-application-in-the-emulator.

- Display the registered devices:

```
sfdk device list
```

- Execute a command on a device by name or index:

```
sfdk device exec {{device-name-or-idx}} {{command}}
```

- Run an interactive shell on a device by name or index:

```
sfdk device exec {{device-name-or-idx}}
```

sfdk emulator device-model-list

This command has been moved to **`sfdk emulator device-model-show`**.

- View documentation for `sfdk emulator device-model-show`:

  ```
  tldr sfdk emulator device-model-show
  ```

sfdk emulator device-model-show

Show emulated device model's properties.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/40-testing-maintain/doc/command.emulator.adoc.

- Show a model properties:

```
sfdk emulator device-model-show {{model}}
```

- Display available device models:

```
sfdk emulator device-model-list
```

sfdk emulator install

Install emulators.

More information: https://docs.sailfishos.org/Tools/Sailfish_SDK/Early_Access/#early-access-build-targets-and-emulator.

- Install an emulator:

```
sfdk emulator install {{name}}
```

- Remove an emulator:

```
sfdk emulator remove {{name}}
```

- Display the available emulators:

```
sfdk emulator list {{[-a|--available]}}
```

- Display the installed emulators:

```
sfdk emulator list
```

sfdk emulator remove

This command has been moved to **`sfdk emulator install`**.

- View documentation for `sfdk emulator install`:

  ```
  tldr sfdk emulator install
  ```

sfdk emulator

Maintain and control emulators.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/40-testing-maintain/doc/command.emulator.adoc.

- Display the installed emulators:

```
sfdk emulator list
```

- Start an emulator:

```
sfdk emulator start {{name}}
```

- Stop an emulator:

```
sfdk emulator stop {{name}}
```

- Display emulator status:

```
sfdk emulator status {{name}}
```

- Run an interactive shell on an emulator:

```
sfdk emulator exec {{emulator}}
```

- Execute a command on an emulator:

```
sfdk emulator exec {{emulator}} {{command}}
```

- Set a property:

```
sfdk emulator set {{name}} {{property}}={{value}}
```

- Show emulator properties:

```
sfdk emulator show {{name}}
```

sfdk engine

Maintain and control the build engine.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/30-building-maintain/doc/command.engine.adoc.

- Start the build engine:

```
sfdk engine start
```

- Stop the build engine:

```
sfdk engine stop
```

- Show the build engine status:

```
sfdk engine status
```

- Set an engine property:

```
sfdk engine set {{property}}={{value}}
```

- Show the engine properties:

```
sfdk engine show
```

- Execute a command on the build engine:

```
sfdk engine exec {{command}}
```

- Start an interactive shell on the build engine:

```
sfdk engine exec
```

sfdk init

Initialize a new project.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/15-building-init/doc/command.init.adoc.

- Initialize a new project of the given type:

```
sfdk init {{[-t|--type]}} {{type}}
```

- Initialize a new project with a specified builder:

```
sfdk init {{[-t|--type]}} {{type}} {{[-b|--builder]}} {{builder}}
```

- Initialize a new project ignoring non-empty directory:

```
sfdk init {{[-t|--type]}} {{type}} --force
```

- List available project types:

```
sfdk init {{[-l|--list-types]}}
```

sfdk maintain

Launch the interactive SDK Maintenance tool.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/10-general/doc/command.maintain.adoc.

- Launch SDK Maintenance tool:

```
sfdk maintain
```

sfdk make-install

Execute make-install build step.

More information: https://docs.sailfishos.org/Develop/Apps/Tutorials/Building_packages_-_advanced_techniques/#building-the-sample-application.

- Run make-install section of RPM SPEC file:

```
sfdk make-install
```

sfdk make

Execute make build step.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/20-building-mb2/doc/command.make.adoc.

- Run make build:

```
sfdk make
```

- Run make with extra arguments:

```
sfdk make {{arguments}}
```

sfdk package

Execute package creation build step.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/20-building-mb2/doc/command.package.adoc.

- Create a package:

  ```
  sfdk package
  ```

- Create a package without executing the check section in RPM SPEC file:

  ```
  sfdk package --no-check
  ```

- Create and sign a package:

  ```
  sfdk package --sign
  ```

sfdk prepare

Execute preparation build step.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/20-building-mb2/doc/command.prepare.adoc.

- Prepare sources using the recipe from RPM SPEC file:

```
sfdk prepare
```

sfdk qmake

Execute qmake build step.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/20-building-mb2/doc/command.qmake.adoc.

- Run qmake build:

```
sfdk qmake
```

- Run shadow qmake build in a specified project:

```
sfdk qmake {{project}}
```

- Run build with extra qmake arguments:

```
sfdk qmake -- {{arguments}}
```

- Run shadow qmake build with extra qmake arguments:

```
sfdk qmake {{project}} {{arguments}}
```

sfdk qmltypes

Generate qmltypes files.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/80-ide-qmltypes/doc/command.qmltypes.adoc.

- Generate qmltypes files:

```
sfdk qmltypes
```

- Generate qmltypes files without deploying them (usually to emulator):

```
sfdk qmltypes --no-deploy
```

- Generate qmltypes files without restoring the emulator after deployment:

```
sfdk qmltypes --no-restore-emulator
```

- Generate qmltypes files and restore the emulator after deployment even on failure:

```
sfdk qmltypes --restore-emulator
```

- Generate qmltypes files without reverting changes which only include removal of statements with `sdk-make-qmltypes:keep` in comments:

```
sfdk qmltypes --no-keep
```

sfdk scrape

Convert source code modifications to patches.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/65-maintaining-mb2/doc/command.scrape.adoc.

- Save source modifications as patches:

```
sfdk scrape
```

- Preview the list of commits to be scrapped:

```
sfdk scrape {{[-n|--dry-run]}}
```

- Scrape while preserving the original patches file names:

```
sfdk scrape --stable
```

- Scrape while saving patches to a specified output directory:

```
sfdk scrape {{[-o|--output-dir]}} {{directory}}
```

- Scrape without removing commits from submodules after creating patches:

```
sfdk scrape --keep
```

sfdk undeploy

Undeploy build results from a device.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/50-testing-mb2/doc/command.undeploy.adoc.

- Undeploy using a specified method (`pkcon`, `rpm`, `rsync`, `sdk` or `zypper`):

```
sfdk undeploy --{{method}}
```

- Preview undeploy without applying the changes:

```
sfdk undeploy --{{method}} {{[-n|--dry-run]}}
```

- Undeploy files in glob pattern `package*`:

```
sfdk undeploy --{{method}} "+package*"
```

- Undeploy all files excluding `ignore*`:

```
sfdk undeploy --{{method}} "-ignore*"
```

sfdk

The command line frontend of the Sailfish SDK.

More information: https://github.com/sailfishos/sailfish-qtcreator/blob/master/share/qtcreator/sfdk/modules/10-general/doc/module.adoc.

- Execute a subcommand:

```
sfdk {{subcommand}}
```

- Execute a subcommand on a custom working directory:

```
git -C {{path/to/directory}} {{subcommand}}
```

- Execute a subcommand with a given configuration set:

```
git -c '{{name}}={{value}}' {{subcommand}}
```

- Display help:

```
sfdk {{[-h|--help]}}
```

- Display help for specific topic (`building`, `testing`, `maintaining`, `ide`, `all`):

```
sfdk --help-{{topic}}
```

- Display version:

```
sfdk --version
```

sfdp

Render an image of a **scaled force-directed** network graph from a **graphviz** file.

Layouts: **dot**, **neato**, **twopi**, **circo**, **fdp**, **sfdp**, **osage** & **patchwork**.

More information: https://graphviz.org/doc/info/command.html.

- Render a PNG image with a filename based on the input filename and output format (uppercase -O):

```
sfdp -T {{png}} -O {{path/to/input.gv}}
```

- Render a SVG image with the specified output filename (lowercase -o):

```
sfdp -T {{svg}} -o {{path/to/image.svg}} {{path/to/input.gv}}
```

- Render the output in PS, PDF, SVG, Fig, PNG, GIF, JPEG, JSON, or DOT format:

```
sfdp -T {{format}} -O {{path/to/input.gv}}
```

- Render a GIF image using `stdin` and `stdout`:

```
echo "{{digraph {this -> that} }}" | sfdp -T {{gif}} >
{{path/to/image.gif}}
```

- Display help:

```
sfdp -?
```

sftp

Secure File Transfer Program.

Interactive program to copy files between hosts over SSH.

For non-interactive file transfers, see **`scp`** or **`rsync`**.

More information: https://manned.org/sftp.

- Connect to a remote server and enter an interactive command mode:

```
sftp {{remote_user}}@{{remote_host}}
```

- Connect using an alternate port:

```
sftp -P {{remote_port}} {{remote_user}}@{{remote_host}}
```

- Connect using a predefined host (in `~/.ssh/config`):

```
sftp {{host}}
```

- Transfer remote file to the local system:

```
get {{path/to/remote_file}}
```

- Transfer local file to the remote system:

```
put {{path/to/local_file}}
```

- Transfer remote directory to the local system recursively (works with `put` too):

```
get -R {{path/to/remote_directory}}
```

- Get list of files on local machine:

```
lls
```

- Get list of files on remote machine:

```
ls
```

sg

Ast-grep is a tool for code structural search, lint, and rewriting.

More information: https://ast-grep.github.io/guide/introduction.html.

- Scan for possible queries using interactive mode:

```
sg scan --interactive
```

- Rewrite code in the current directory using patterns:

```
sg run --pattern '{{foo}}' --rewrite '{{bar}}' --lang
{{python}}
```

- Visualize possible changes without applying them:

```
sg run --pattern '{{useState<number>($A)}}' --rewrite
'{{useState($A)}}' --lang {{typescript}}
```

- Output results as JSON, extract information using `jq` and interactively view it using `jless`:

```
sg run --pattern '{{Some($A)}}' --rewrite '{{None}}' --json |
jq '{{.[].replacement}}' | jless
```

sg_raw

Send arbitrary SCSI command to a connected device.

More information: https://manned.org/sg_raw.

- Send a command to an optical SCSI device assigned to `sr0` to load the media in its tray:

```
sg_raw /dev/sr0 EA 00 00 00 00 01
```

- Read data from `IFILE` instead of `stdin`:

```
sg_raw {{[-i|--infile]}} {{path/to/IFILE}} {{/dev/sgX}} {{SCSI_command}}
```

- Skip the first `LEN` bytes of input data:

```
sg_raw {{[-k|--skip]}} {{LEN}} {{/dev/sgX}} {{SCSI_command}}
```

- Read `SLEN` bytes of data and send to the device:

```
sg_raw {{[-s|--send]}} {{SLEN}} {{/dev/sgX}} {{SCSI_command}}
```

- Wait up to `SEC` seconds for `sg_raw` to finish processing:

```
sg_raw {{[-t|--timeout]}} {{SEC}} {{/dev/sgX}} {{SCSI_command}}
```

- Increase verbosity level by 1:

```
sg_raw {{[-v|--verbose]}} {{/dev/sgX}} {{SCSI_command}}
```

- Dump returned data in binary form:

```
sg_raw {{[-b|--binary]}} {{/dev/sgX}} {{SCSI_command}}
```

- Write data received from the specified device to an `OFILE`:

```
sg_raw {{[-o|--outfile]}} {{path/to/OFILE}} {{/dev/sgX}} {{SCSI_command}}
```

sgitopnm

Convert an SGI file to a PNM file.

More information: https://netpbm.sourceforge.net/doc/sgitopnm.html.

- Convert an SGI image to a PNM file:

```
sgitopnm {{path/to/input.sgi}} > {{path/to/output.pnm}}
```

- Display information about the SGI file:

```
sgitopnm {{[-verb|-verbose]}} {{path/to/input.sgi}} > {{path/to/output.pnm}}
```

- Extract channel n of the SGI file:

```
sgitopnm {{[-c|-channel]}} {{n}} {{path/to/input.sgi}} > {{path/to/output.pnm}}
```

sgpt

Productivity tool powered by OpenAI's GPT models.

More information: https://github.com/TheR1D/shell_gpt#readme.

- Use it as a search engine, asking for the mass of the sun:

```
sgpt "{{mass of the sun}}"
```

- Execute Shell commands, and apply `chmod 444` to all files in the current directory:

```
sgpt --shell "{{make all files in current directory read only}}"
```

- Generate code, solving classic fizz buzz problem:

```
sgpt --code "{{solve fizz buzz problem using Python}}"
```

- Start a chat session with a unique session name:

```
sgpt --chat {{session_name}} "{{please remember my favorite number: 4}}"
```

- Start a REPL (Read-eval-print loop) session:

```
sgpt --repl {{command}}
```

- Display help:

```
sgpt --help
```

sh

Bourne shell, the standard command language interpreter.

See also: **`histexpand`** for history expansion.

More information: https://manned.org/sh.

- Start an interactive shell session:

```
sh
```

- Execute a command and then exit:

```
sh -c "{{command}}"
```

- Execute a script:

```
sh {{path/to/script.sh}}
```

- Read and execute commands from `stdin`:

```
sh -s
```

sha1sum

Calculate SHA1 cryptographic checksums.

More information: https://www.gnu.org/software/coreutils/manual/html_node/sha1sum-invocation.html.

- Calculate the SHA1 checksum for one or more files:

```
sha1sum {{path/to/file1 path/to/file2 ...}}
```

- Calculate and save the list of SHA1 checksums to a file:

```
sha1sum {{path/to/file1 path/to/file2 ...}} > {{path/to/file.sha1}}
```

- Calculate a SHA1 checksum from `stdin`:

```
{{command}} | sha1sum
```

- Read a file of SHA1 checksums and filenames and verify all files have matching checksums:

```
sha1sum {{[-c|--check]}} {{path/to/file.sha1}}
```

- Only show a message for missing files or when verification fails:

```
sha1sum {{[-c|--check]}} --quiet {{path/to/file.sha1}}
```

- Only show a message when verification fails, ignoring missing files:

```
sha1sum --ignore-missing {{[-c|--check]}} --quiet {{path/to/file.sha1}}
```

- Check a known SHA1 checksum of a file:

```
echo {{known_sha1_checksum_of_the_file}} {{path/to/file}} | sha1sum {{[-c|--check]}}
```

sha224sum

Calculate SHA224 cryptographic checksums.

More information: https://www.gnu.org/software/coreutils/manual/html_node/sha2-utilities.html.

- Calculate the SHA224 checksum for one or more files:

```
sha224sum {{path/to/file1 path/to/file2 ...}}
```

- Calculate and save the list of SHA224 checksums to a file:

```
sha224sum {{path/to/file1 path/to/file2 ...}} > {{path/to/file.sha224}}
```

- Calculate a SHA224 checksum from `stdin`:

```
{{command}} | sha224sum
```

- Read a file of SHA224 checksums and filenames and verify all files have matching checksums:

```
sha224sum {{[-c|--check]}} {{path/to/file.sha224}}
```

- Only show a message for missing files or when verification fails:

```
sha224sum {{[-c|--check]}} --quiet {{path/to/file.sha224}}
```

- Only show a message when verification fails, ignoring missing files:

```
sha224sum --ignore-missing {{[-c|--check]}} --quiet {{path/to/file.sha224}}
```

- Check a known SHA224 checksum of a file:

```
echo {{known_sha224_checksum_of_the_file}} {{path/to/file}} | sha224sum {{[-c|--check]}}
```

sha256sum

Calculate SHA256 cryptographic checksums.

More information: https://www.gnu.org/software/coreutils/manual/html_node/sha2-utilities.html.

- Calculate the SHA256 checksum for one or more files:

```
sha256sum {{path/to/file1 path/to/file2 ...}}
```

- Calculate and save the list of SHA256 checksums to a file:

```
sha256sum {{path/to/file1 path/to/file2 ...}} > {{path/to/file.sha256}}
```

- Calculate a SHA256 checksum from `stdin`:

```
{{command}} | sha256sum
```

- Read a file of SHA256 checksums and filenames and verify all files have matching checksums:

```
sha256sum {{[-c|--check]}} {{path/to/file.sha256}}
```

- Only show a message for missing files or when verification fails:

```
sha256sum {{[-c|--check]}} --quiet {{path/to/file.sha256}}
```

- Only show a message when verification fails, ignoring missing files:

```
sha256sum --ignore-missing {{[-c|--check]}} --quiet {{path/to/file.sha256}}
```

- Check a known SHA256 checksum of a file:

```
echo {{known_sha256_checksum_of_the_file}} {{path/to/file}} | sha256sum {{[-c|--check]}}
```

sha384sum

Calculate SHA384 cryptographic checksums.

More information: https://www.gnu.org/software/coreutils/manual/html_node/sha2-utilities.html.

- Calculate the SHA384 checksum for one or more files:

```
sha384sum {{path/to/file1 path/to/file2 ...}}
```

- Calculate and save the list of SHA384 checksums to a file:

```
sha384sum {{path/to/file1 path/to/file2 ...}} > {{path/to/file.sha384}}
```

- Calculate a SHA384 checksum from `stdin`:

```
{{command}} | sha384sum
```

- Read a file of SHA384 checksums and filenames and verify all files have matching checksums:

```
sha384sum {{[-c|--check]}} {{path/to/file.sha384}}
```

- Only show a message for missing files or when verification fails:

```
sha384sum {{[-c|--check]}} --quiet {{path/to/file.sha384}}
```

- Only show a message when verification fails, ignoring missing files:

```
sha384sum --ignore-missing {{[-c|--check]}} --quiet {{path/to/file.sha384}}
```

- Check a known SHA384 checksum of a file:

```
echo {{known_sha384_checksum_of_the_file}} {{path/to/file}} | sha384sum {{[-c|--check]}}
```

sha512sum

Calculate SHA512 cryptographic checksums.

More information: https://www.gnu.org/software/coreutils/manual/html_node/sha2-utilities.html.

- Calculate the SHA512 checksum for one or more files:

```
sha512sum {{path/to/file1 path/to/file2 ...}}
```

- Calculate and save the list of SHA512 checksums to a file:

```
sha512sum {{path/to/file1 path/to/file2 ...}} > {{path/to/file.sha512}}
```

- Calculate a SHA512 checksum from `stdin`:

```
{{command}} | sha512sum
```

- Read a file of SHA512 checksums and filenames and verify all files have matching checksums:

```
sha512sum {{[-c|--check]}} {{path/to/file.sha512}}
```

- Only show a message for missing files or when verification fails:

```
sha512sum {{[-c|--check]}} --quiet {{path/to/file.sha512}}
```

- Only show a message when verification fails, ignoring missing files:

```
sha512sum --ignore-missing {{[-c|--check]}} --quiet {{path/to/file.sha512}}
```

- Check a known SHA512 checksum of a file:

```
echo {{known_sha512_checksum_of_the_file}} {{path/to/file}} | sha512sum {{[-c|--check]}}
```

shar

Create a shell archive.

More information: https://manned.org/shar.

- Create a shell script that when executed extracts the given files from itself:

```
shar {{path/to/file1 path/to/file2 ...}} > {{path/to/
archive.sh}}
```

shards

Dependency management tool for the Crystal language.

More information: https://crystal-lang.org/reference/the_shards_command.

- Create a skeleton `shard.yml` file:

```
shards init
```

- Install dependencies from a `shard.yml` file:

```
shards install
```

- Update all dependencies:

```
shards update
```

- List all installed dependencies:

```
shards list
```

- Display version of dependency:

```
shards version {{path/to/dependency_directory}}
```

shasum

Calculate SHA cryptographic checksums.

More information: https://manned.org/shasum.

- Calculate the SHA1 checksum for one or more files:

```
shasum {{path/to/file1 path/to/file2 ...}}
```

- Calculate the SHA checksum for one or more files with the specified algorithm:

```
shasum --algorithm {{1|224|256|384|512|512224|512256}} {{path/to/file1 path/to/file2 ...}}
```

- Calculate a SHA1 checksum from `stdin`:

```
{{command}} | shasum
```

- Calculate and save the list of SHA256 checksums to a file:

```
shasum --algorithm 256 {{path/to/file1 path/to/file2 ...}} > {{path/to/file.sha256}}
```

- Read a file of SHA checksums and filenames and verify all files have matching checksums (the algorithm will be automatically detected):

```
shasum {{[-c|--check]}} {{path/to/file}}
```

- Only show a message for missing files or when verification fails:

```
shasum {{[-c|--check]}} --quiet {{path/to/file}}
```

- Only show a message when verification fails, ignoring missing files:

```
shasum --ignore-missing {{[-c|--check]}} --quiet {{path/to/file}}
```

- Check a known SHA checksum of a file:

```
echo {{known_sha_checksum_of_the_file}} {{path/to/file}} | shasum {{[-c|--check]}}
```

shc

Generic shell script compiler.

More information: https://manned.org/shc.

- Compile a shell script:

```
shc -f {{script}}
```

- Compile a shell script and specify an output binary file:

```
shc -f {{script}} -o {{binary}}
```

- Compile a shell script and set an expiration date for the executable:

```
shc -f {{script}} -e {{dd/mm/yyyy}}
```

- Compile a shell script and set a message to display upon expiration:

```
shc -f {{script}} -e {{dd/mm/yyyy}} -m "{{Please contact your provider}}"
```

shellcheck

Statically check shell scripts for errors, usage of deprecated/insecure features, and bad practices.

More information: https://github.com/koalaman/shellcheck/wiki.

- Check a shell script:

```
shellcheck {{path/to/script.sh}}
```

- Check a shell script interpreting it as the specified shell dialect (overrides the shebang at the top of the script):

```
shellcheck {{[-s|--shell]}} {{sh|bash|dash|ksh}} {{path/to/script.sh}}
```

- Ignore one or more error types:

```
shellcheck {{[-e|--exclude]}} {{SC1009,SC1073,...}} {{path/to/script.sh}}
```

- Also check any sourced shell scripts:

```
shellcheck {{[-a|--check-sourced]}} {{path/to/script.sh}}
```

- Display output in the specified format (defaults to `tty`):

```
shellcheck {{[-f|--format]}} {{tty|checkstyle|diff|gcc|json|json1|quiet}} {{path/to/script.sh}}
```

- Enable one or more [o]ptional checks:

```
shellcheck {{[-o|--enable]}} {{add-default-case,avoid-nullary-conditions,...}} {{path/to/script.sh}}
```

- List all available optional checks that are disabled by default:

```
shellcheck --list-optional
```

- Adjust the level of severity to consider (defaults to `style`):

```
shellcheck {{[-S|--severity]}} {{error|warning|info|style}} {{path/to/script.sh}}
```

shfmt

Shell parser, formatter and interpreter.

More information: https://pkg.go.dev/mvdan.cc/sh.

- Print a formatted version of a shell script:

```
shfmt {{path/to/file}}
```

- List unformatted files:

```
shfmt --list {{path/to/directory}}
```

- Write the result to the file instead of printing it to the terminal:

```
shfmt --write {{path/to/file}}
```

- Simplify the code, removing redundant pieces of syntax (i.e. removing "$" from vars in expressions):

```
shfmt --simplify {{path/to/file}}
```

shift

Move positional parameters.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-shift.

- Remove the first positional parameter:

```
shift
```

- Remove the first `n` positional parameters:

```
shift {{n}}
```

shiori

Simple bookmark manager built with Go.

More information: https://github.com/go-shiori/shiori/blob/master/docs/Usage.md.

- Import bookmarks from HTML Netscape bookmark format file:

  ```
  shiori import {{path/to/bookmarks.html}}
  ```

- Save the specified URL as bookmark:

  ```
  shiori add {{url}}
  ```

- List the saved bookmarks:

  ```
  shiori print
  ```

- Open the saved bookmark in a browser:

  ```
  shiori open {{bookmark_id}}
  ```

- Start the web interface for managing bookmarks at port 8181:

  ```
  shiori serve --port {{8181}}
  ```

shopt

Manage Bash shell options: variables (stored in **`$BASHOPTS`**) that control behavior specific to the Bash shell.

Generic POSIX shell variables (stored in **`$SHELLOPTS`**) are managed with the **`set`** command instead.

More information: https://www.gnu.org/software/bash/manual/bash.html#The-Shopt-Builtin.

- List of all settable options and whether they are set:

```
shopt
```

- Set an option:

```
shopt -s {{option_name}}
```

- Unset an option:

```
shopt -u {{option_name}}
```

- Print a list of all options and their status formatted as runnable `shopt` commands:

```
shopt -p
```

- Display help:

```
help shopt
```

shotcut

A program for video editing.

More information: https://shotcut.org/notes/command-line-options/.

- Start Shotcut:

```
shotcut
```

- Open audio/video files:

```
shotcut {{path/to/file1 path/to/file2 ...}}
```

- Start with a specific audio driver:

```
shotcut --SDL_AUDIODRIVER "{{pulseaudio}}"
```

- Start in fullscreen:

```
shotcut --fullscreen
```

- Start with GPU processing:

```
shotcut --gpu
```

showfigfonts

Display available figlet fonts.

See also: **`figlet`**.

More information: https://manned.org/showfigfonts.

- Display available fonts:

```
showfigfonts
```

- Display available fonts using a specific text:

```
showfigfonts {{input_text}}
```

shred

Overwrite files to securely delete data.

More information: https://www.gnu.org/software/coreutils/manual/html_node/shred-invocation.html.

- Overwrite a file:

```
shred {{path/to/file}}
```

- Overwrite a file and show progress on the screen:

```
shred {{[-v|--verbose]}} {{path/to/file}}
```

- Overwrite a file, leaving zeros instead of random data:

```
shred {{[-z|--zero]}} {{path/to/file}}
```

- Overwrite a file a specific number of times:

```
shred {{[-n|--iterations]}} {{25}} {{path/to/file}}
```

- Overwrite a file and remove it:

```
shred {{[-u|--remove]}} {{path/to/file}}
```

- Overwrite a file 100 times, add a final overwrite with zeros, remove the file after overwriting it and show verbose progress on the screen:

```
shred {{[-vzun|--verbose --zero --remove --iterations]}} 100 {{path/to/file}}
```

shuf

Generate random permutations.

More information: https://www.gnu.org/software/coreutils/manual/html_node/shuf-invocation.html.

- Randomize the order of lines in a file and output the result:

```
shuf {{path/to/file}}
```

- Only output the first 5 entries of the result:

```
shuf {{[-n|--head-count]}} 5 {{path/to/file}}
```

- Write the output to another file:

```
shuf {{path/to/input_file}} {{[-o|--output]}} {{path/to/output_file}}
```

- Generate 3 random numbers in the range 1-10 (inclusive):

```
shuf {{[-n|--head-count]}} 3 {{[-i|--input-range]}} 1-10 {{[-r|--repeat]}}
```

siege

HTTP loadtesting and benchmarking tool.

More information: https://www.joedog.org/siege-manual/.

- Test a URL with default settings:

```
siege {{https://example.com}}
```

- Test a list of URLs:

```
siege {{[-f|--file]}} {{path/to/url_list.txt}}
```

- Test list of URLs in a random order (Simulates internet traffic):

```
siege {{[-i|--internet]}} {{[-f|--file]}} {{path/to/
url_list.txt}}
```

- Benchmark a list of URLs (without waiting between requests):

```
siege {{[-b|--benchmark]}} {{[-f|--file]}} {{path/to/
url_list.txt}}
```

- Set the amount of concurrent connections:

```
siege {{[-c|--concurrent]}} {{50}} {{[-f|--file]}} {{path/to/
url_list.txt}}
```

- Set how long for the siege to run for:

```
siege {{[-t|--time]}} {{30s}} {{[-f|--file]}} {{path/to/
url_list.txt}}
```

silicon

Create an image of source code.

See also: **`freeze`**.

More information: https://github.com/Aloxaf/silicon#examples.

- Generate an image from a specific source file:

```
silicon {{path/to/source_file}} {{[-o|--output]}} {{path/to/output_image}}
```

- Generate an image from a source file with a specific programming language syntax highlighting (e.g. `rust`, `py`, `js`, etc.):

```
silicon {{path/to/source_file}} {{[-o|--output]}} {{path/to/output_image}} {{[-l|--language]}} {{language|extension}}
```

- Generate an image from `stdin`:

```
{{command}} | silicon {{[-o|--output]}} {{path/to/output_image}}
```

simplehttpserver

A simple HTTP/S server that supports file upload, basic authentication, and YAML rules for custom responses.

A Go alternative to Python's **`http.server`**.

More information: https://github.com/projectdiscovery/simplehttpserver.

- Start the HTTP server serving the current directory with verbose output (listen on all interfaces and port 8000 by default):

```
simplehttpserver -verbose
```

- Start the HTTP server with basic authentication serving a specific path over port 80 on all interfaces:

```
sudo simplehttpserver -basic-auth {{username}}:{{password}} -path {{/var/www/html}} -listen 0.0.0.0:80
```

- Start the HTTP server, enabling HTTPS using a self-signed certificate with custom SAN on all interfaces:

```
sudo simplehttpserver -https -domain {{*.selfsigned.com}} -listen 0.0.0.0:443
```

- Start the HTTP server with custom response headers and upload capability:

```
simplehttpserver -upload -header '{{X-Powered-By: Go}}' -header '{{Server: SimpleHTTPServer}}'
```

- Start the HTTP server with customizable rules in YAML (see documentation for DSL):

```
simplehttpserver -rules {{rules.yaml}}
```

sindresorhus

Sindre Sorhus's personal CLI.

More information: https://github.com/sindresorhus/sindresorhus-cli.

- Start Sindre's interactive CLI:

```
sindresorhus
```

singularity

Manage Singularity containers and images.

More information: https://singularity-docs.readthedocs.io/en/latest/#commands.

- Download a remote image from Sylabs Cloud:

```
singularity pull --name {{image.sif}} {{library://godlovedc/funny/lolcow:latest}}
```

- Rebuild a remote image using the latest Singularity image format:

```
singularity build {{image.sif}} {{docker://godlovedc/lolcow}}
```

- Start a container from an image and get a shell inside it:

```
singularity shell {{image.sif}}
```

- Start a container from an image and run a command:

```
singularity exec {{image.sif}} {{command}}
```

- Start a container from an image and execute the internal runscript:

```
singularity run {{image.sif}}
```

- Build a singularity image from a recipe file:

```
sudo singularity build {{image.sif}} {{recipe}}
```

sirtopnm

Convert a Solitaire Image Recorder file to a PNM file.

More information: https://netpbm.sourceforge.net/doc/sirtopnm.html.

- Convert a SIR image to a PNM file:

```
sirtopnm {{path/to/input.sir}} > {{path/to/output.pnm}}
```

sk

Fuzzy finder written in Rust.

Similar to **`fzf`**.

More information: https://github.com/lotabout/skim.

- Start `skim` on all files in the specified directory:

```
find {{path/to/directory}} -type f | sk
```

- Start `skim` for running processes:

```
ps aux | sk
```

- Start `skim` with a specified query:

```
sk --query "{{query}}"
```

- Select multiple files with `<Shift Tab>` and write to a file:

```
find {{path/to/directory}} -type f | sk --multi > {{path/to/file}}
```

skaffold

Facilitate continuous development for Kubernetes applications.

More information: https://skaffold.dev.

- Build the artifacts:

  ```
  skaffold build {{[-f|--filename]}} {{skaffold.yaml}}
  ```

- Build and deploy your app every time your code changes:

  ```
  skaffold dev {{[-f|--filename]}} {{skaffold.yaml}}
  ```

- Run a pipeline file:

  ```
  skaffold run {{[-f|--filename]}} {{skaffold.yaml}}
  ```

- Run a diagnostic on Skaffold:

  ```
  skaffold diagnose {{[-f|--filename]}} {{skaffold.yaml}}
  ```

- Deploy the artifacts:

  ```
  skaffold deploy {{[-f|--filename]}} {{skaffold.yaml}}
  ```

skate

Simple and powerful key-value store.

More information: https://github.com/charmbracelet/skate.

- Store a key and a value on the default database:

```
skate set "{{key}}" "{{value}}"
```

- Show your keys saved on the default database:

```
skate list
```

- Delete key and value from the default database:

```
skate delete "{{key}}"
```

- Create a new key and value in a new database:

```
skate set "{{key}}"@"{{database_name}}" "{{value}}"
```

- Show your keys saved in a non default database:

```
skate list @"{{database_name}}"
```

- Delete key and value from a specific database:

```
skate delete "{{key}}"@"{{database_name}}"
```

- Show the databases available:

```
skate list-dbs
```

- Delete local db and pull down fresh copy from Charm Cloud:

```
skate reset @"{{database_name}}"
```

skicka

Manage your Google Drive.

More information: https://github.com/google/skicka#usage.

- Upload a file/folder to Google Drive:

```
skicka upload {{path/to/local}} {{path/to/remote}}
```

- Download a file/folder from Google Drive:

```
skicka download {{path/to/remote}} {{path/to/local}}
```

- List files:

```
skicka ls {{path/to/folder}}
```

- Show amount of space used by children folders:

```
skicka du {{path/to/parent_folder}}
```

- Create a folder:

```
skicka mkdir {{path/to/folder}}
```

- Delete a file:

```
skicka rm {{path/to/file}}
```

skopeo

Container image management toolbox.

Provides various utility commands to manage remote container images.

More information: https://github.com/containers/skopeo.

- Inspect a remote image from a registry:

```
skopeo inspect docker://{{registry_hostname}}/{{image:tag}}
```

- List available tags for a remote image:

```
skopeo list-tags docker://{{registry_hostname}}/{{image}}
```

- Download an image from a registry:

```
skopeo copy docker://{{registry_hostname}}/{{image:tag}} dir:
{{path/to/directory}}
```

- Copy an image from one registry to another:

```
skopeo copy docker://{{source_registry}}/{{image:tag}}
docker://{{destination_registry}}/{{image:tag}}
```

- Delete an image from a registry:

```
skopeo delete docker://{{registry_hostname}}/{{image:tag}}
```

- Log in to a registry:

```
skopeo login --username {{username}} {{registry_hostname}}
```

sl

Steam locomotive running through your terminal.

More information: https://github.com/mtoyoda/sl.

- Let a steam locomotive run through your terminal:

```
sl
```

- The train burns, people scream:

```
sl -a
```

- Let the train fly:

```
sl -F
```

- Make the train little:

```
sl -l
```

- Let the user exit (<Ctrl c>):

```
sl -e
```

slackcat

Utility for passing files and command output to Slack.

More information: https://github.com/bcicen/slackcat#usage.

- Post a file to Slack:

```
slackcat {{[-c|--channel]}} {{channel_name}} {{path/to/file}}
```

- Post a file to Slack with a custom filename:

```
slackcat {{[-c|--channel]}} {{channel_name}} {{[-n|--
filename]}} {{filename}} {{path/to/file}}
```

- Pipe command output to Slack as a text snippet:

```
{{command}} | slackcat {{[-c|--channel]}} {{channel_name}}
{{[-n|--filename]}} {{snippet_name}}
```

- Stream command output to Slack continuously:

```
{{command}} | slackcat {{[-c|--channel]}} {{channel_name}}
{{[-s|--stream]}}
```

sldtoppm

Convert an AutoCAD slide file to a PPM image.

More information: https://netpbm.sourceforge.net/doc/sldtoppm.html.

- Convert an SLD file to a PPM image:

```
sldtoppm {{path/to/input.sld}} > {{path/to/output.ppm}}
```

- Compensate for non-square pixels by scaling the width of the image:

```
sldtoppm {{[-a|-adjust]}} {{path/to/input.sld}} > {{path/to/output.ppm}}
```

sleep

Delay for a specified amount of time.

More information: https://manned.org/sleep.

- Delay in seconds:

```
sleep {{seconds}}
```

- Execute a specific command after 20 seconds delay:

```
sleep 20 && {{command}}
```

slimrb

Convert Slim files to HTML.

More information: https://rubydoc.info/gems/slim/frames#slim-command-slimrb.

- Convert a Slim file to HTML:

```
slimrb {{input.slim}} {{output.html}}
```

- Convert a Slim file and output to prettified HTML:

```
slimrb --pretty {{input.slim}} {{output.html}}
```

- Convert a Slim file to ERB:

```
slimrb --erb {{input.slim}} {{output.erb}}
```

slocate

Secure variant of GNU Locate.

See also: **`locate`**.

More information: https://manned.org/slocate.

- Enable quiet mode to suppress error messages:

```
slocate -q
```

- Limit the number of results shown:

```
slocate -n {{number}}
```

- Build an `slocate` database starting at path `/`:

```
slocate -u
```

- Build an `slocate` database starting at a given directory:

```
slocate -U {{path/to/directory}}
```

- Update an `slocate` database using the default `/etc/updatedb.conf` configuration:

```
slocate -c
```

- Set the security level of `slocate`, with `0` being disabled, and `1` being secure:

```
slocate -l {{0|1}}
```

- Specify the database that `slocate` should search in:

```
slocate {{[-d|--database]}} {{path/to/directory}}
```

- Search the `slocate` database using a specific `regex` string:

```
slocate {{[-r|--regexp]}} {{regex}}
```

smalltalkci

Framework for testing Smalltalk projects with GitHub Actions, Travis CI, AppVeyor, GitLab CI, and others.

More information: https://github.com/hpi-swa/smalltalkCI.

- Run tests for a configuration file:

  ```
  smalltalkci {{path/to/.smalltalk.ston}}
  ```

- Run tests for the `.smalltalk.ston` configuration in the current directory:

  ```
  smalltalkci
  ```

- Debug tests in headful mode (show VM window):

  ```
  smalltalkci --headful
  ```

- Download and prepare a well-known smalltalk image for the tests:

  ```
  smalltalkci --smalltalk {{Squeak64-Trunk}}
  ```

- Specify a custom Smalltalk image and VM:

  ```
  smalltalkci --image {{path/to/Smalltalk.image}} --vm {{path/to/vm}}
  ```

- Clean up caches and delete builds:

  ```
  smalltalkci --clean
  ```

smartctl

Monitor disk health including SMART data.

More information: https://manned.org/smartctl.

- Display SMART health summary:

```
sudo smartctl {{[-H|--health]}} {{/dev/sdX}}
```

- Display device information:

```
sudo smartctl {{[-i|--info]}} {{/dev/sdX}}
```

- Start a short/long self-test in the background:

```
sudo smartctl {{[-t|--test]}} {{short|long}} {{/dev/sdX}}
```

- Display the self-test log:

```
sudo smartctl {{[-l|--log]}} selftest
```

- Display current/last self-test status and other SMART capabilities:

```
sudo smartctl {{[-c|--capabilities]}} {{/dev/sdX}}
```

- Display exhaustive SMART data:

```
sudo smartctl {{[-a|--all]}} {{/dev/sdX}}
```

smbclient.py

A Python-based SMB client for interacting with SMB servers.

More information: https://github.com/fortra/impacket.

- Connect to an SMB server with username and password:

```
smbclient.py {{domain}}/{{username}}:{{password}}@{{target}}
```

- Connect using NTLM hashes for authentication:

```
smbclient.py -hashes {{LM_HASH}}:{{NT_HASH}} {{domain}}/
{{username}}@{{target}}
```

- Connect using Kerberos authentication:

```
smbclient.py -k {{domain}}/{{username}}@{{target}}
```

- Connect specifying a domain controller IP:

```
smbclient.py -dc-ip {{domain_controller_ip}} {{domain}}/
{{username}}:{{password}}@{{target}}
```

- Connect to a specific target IP instead of NetBIOS name:

```
smbclient.py -target-ip {{target_ip}} {{domain}}/
{{username}}:{{password}}@{{target}}
```

- Connect to a non-standard SMB port:

```
smbclient.py -port {{port}} {{domain}}/{{username}}:
{{password}}@{{target}}
```

- Execute commands from an input file in the SMB shell:

```
smbclient.py -inputfile {{path/to/input_file}} {{domain}}/
{{username}}:{{password}}@{{target}}
```

- Log SMB client commands to an output file:

```
smbclient.py -outputfile {{path/to/output_file}} {{domain}}/
{{username}}:{{password}}@{{target}}
```

smbmap

Enumerate samba share drives across an entire domain.

More information: https://github.com/ShawnDEvans/smbmap.

- Enumerate hosts with NULL sessions enabled and open shares:

```
smbmap --host-file {{path/to/file}}
```

- Display SMB shares and permissions on a [H]ost, prompting for user's password or NTLM hash:

```
smbmap {{[-u|--username]}} {{username}} --prompt -H {{ip}}
```

- Execute a shell command on a remote system:

```
smbmap {{[-u|--username]}} {{username}} --prompt -H {{ip}} -x
{{command}}
```

- Enumerate hosts and check SMB file permissions:

```
smbmap --host-file {{path/to/file}} {{[-u|--username]}}
{{username}} {{[-p|--password]}} {{password}} -q
```

- Connect to an ip or hostname through smb using a username and password:

```
smbmap {{[-u|--username]}} {{username}} {{[-p|--password]}}
{{password}} -d {{domain}} -H {{ip_or_hostname}}
```

- Locate and download files [R]ecursively up to n levels depth, searching for filename pattern (regex), and excluding certain shares:

```
smbmap --host-file {{path/to/file}} {{[-u|--username]}}
{{username}} {{[-p|--password]}} {{password}} -q -R --depth
{{n}} --exclude {{sharename}} -A {{filepattern}}
```

- Upload file through smb using username and password:

```
smbmap {{[-u|--username]}} {{username}} {{[-p|--password]}}
{{password}} -d {{domain}} -H {{ip_or_hostname}} --upload
{{path/to/file}} '{{/share_name/remote_filename}}'
```

- Display SMB shares and recursively list directories and files, searching for file content matching a regex:

```
smbmap {{[-u|--username]}} {{username}} --prompt -H {{ip}} -R
-F {{pattern}}
```

smbserver.py

A Python-based SMB server for hosting shares (requires root for port 445).

More information: https://github.com/fortra/impacket.

- Set up a basic SMB share:

```
smbserver.py {{sharename}} {{path/to/share}}
```

- Set up a share with a custom comment:

```
smbserver.py -comment {{my_share}} {{sharename}} {{path/to/share}}
```

- Set up a share with username and password authentication:

```
smbserver.py -username {{username}} -password {{password}} {{sharename}} {{path/to/share}}
```

- Set up a share with NTLM hash authentication:

```
smbserver.py -hashes {{LMHASH}}:{{NTHASH}} {{sharename}} {{path/to/share}}
```

- Set up a share on a specific interface:

```
smbserver.py {{[-ip|--interface-address]}} {{interface_ip_address}} {{sharename}} {{path/to/share}}
```

- Set up a share on a non-standard SMB port:

```
smbserver.py -port {{port}} {{sharename}} {{path/to/share}}
```

- Set up a share with SMB2 support:

```
smbserver.py -smb2support {{sharename}} {{path/to/share}}
```

- Set up a share and log commands to an output file:

```
smbserver.py -outputfile {{path/to/output_file}} {{sharename}} {{path/to/share}}
```

sn

Mono StrongName utility for signing and verifying IL assemblies.

More information: https://manned.org/sn.

- Generate a new StrongNaming key:

```
sn -k {{path/to/key.snk}}
```

- Re-sign an assembly with the specified private key:

```
sn -R {{path/to/assembly.dll}} {{path/to/key_pair.snk}}
```

- Show the public key of the private key that was used to sign an assembly:

```
sn -T {{path/to/assembly.exe}}
```

- Extract the public key to a file:

```
sn -e {{path/to/assembly.dll}} {{path/to/output.pub}}
```

snakefmt

Format Snakemake files.

More information: https://github.com/snakemake/snakefmt.

- Format a specific Snakefile:

```
snakefmt {{path/to/snakefile}}
```

- Format all Snakefiles recursively in a specific directory:

```
snakefmt {{path/to/directory}}
```

- Format a file using a specific configuration file:

```
snakefmt --config {{path/to/config.toml}} {{path/to/
snakefile}}
```

- Format a file using a specific maximum line length:

```
snakefmt --line-length {{100}} {{path/to/snakefile}}
```

- Display the changes that would be performed without performing them (dry-run):

```
snakefmt --diff {{path/to/snakefile}}
```

sngrep

Display SIP calls message flows from terminal.

More information: https://manned.org/sngrep.

- Visualize SIP packets from a PCAP file:

```
sngrep -I {{path/to/file.pcap}}
```

- Visualize only dialogs starting with INVITE packets with RTP packets from a PCAP file:

```
sngrep -crI {{path/to/file.pcap}}
```

- Real-time interface with only dialogs starting with INVITE packets with RTP packets:

```
sngrep -cr
```

- Only capture packets without interface to a file:

```
sngrep -NO {{path/to/file.pcap}}
```

sniff.py

Capture and display network packets using the **`pcapy`** library.

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- List available network interfaces and select one to start capturing packets (requires `sudo`):

```
sudo sniff.py
```

- Capture packets and save output to a file while displaying it on the terminal:

```
sudo sniff.py | sudo tee {{path/to/output_file}}
```

sniffer.py

Capture and display network packets for specified protocols using raw sockets.

Part of the Impacket suite.

More information: https://github.com/fortra/impacket.

- Capture packets for default protocols (ICMP, TCP, UDP):

```
sniffer.py
```

- Capture packets for specific protocols (e.g., ICMP, TCP):

```
sniffer.py {{protocol1 protocol2 ...}}
```

- Capture packets for specific protocols (e.g., TCP):

```
sniffer.py tcp
```

snmpbulkget

Query the next value in the MIB tree and all of its adjacent values.

More information: https://manned.org/snmpbulkget.

- Request the next value from the SNMP agent:

```
snmpbulkget -v {{version}} -c {{community}} {{ip}} {{oid}}
```

- Display the full Object Identifier (OID) path:

```
snmpbulkget -v {{version}} -c {{community}} -O f {{ip}}
{{oid}}
```

- Display help:

```
snmpbulkget {{[-h|--help]}}
```

snmpdf

Fetch disk space usage information.

More information: https://manned.org/snmpdf.

- Fetch the disk space usage:

```
snmpdf -v {{version}} -c {{community}} {{ip}}
```

- Display help:

```
snmpdf {{[-h|--help]}}
```

snmpget

Query using the SNMP protocol.

More information: https://manned.org/snmpget.

- Request a single value from the SNMP agent:

```
snmpget -v {{version}} -c {{community}} {{ip}} {{oid}}
```

- Display the full Object Identifier (OID) path:

```
snmpget -v {{version}} -c {{community}} -O f {{ip}} {{oid}}
```

- Display help:

```
snmpget {{[-h|--help]}}
```

snmpgetnext

Query the next value in the MIB tree.

More information: https://manned.org/snmpgetnext.

- Request the next value from the SNMP agent:

```
snmpgetnext -v {{version}} -c {{community}} {{ip}} {{oid}}
```

- Display the full Object Identifier (OID) path:

```
snmpgetnext -v {{version}} -c {{community}} -O f {{ip}}
{{oid}}
```

- Display help:

```
snmpgetnext {{[-h|--help]}}
```

snmpnetstat

Fetch networking status using SNMP.

More information: https://manned.org/snmpnetstat.

- Fetch the networking status:

```
snmpnetstat -v {{version}} -c {{community}} {{ip}}
```

- Display help:

```
snmpnetstat {{[-h|--help]}}
```

snmpset

Set a value using the SNMP protocol.

More information: https://manned.org/snmpset.

- Set a value:

```
snmpset -v {{version}} -c {{community}} {{ip}} {{oid}}
{{value_type}} {{value}}
```

- Display help:

```
snmpset {{[-h|--help]}}
```

snmpstatus

Fetch fixed MIB status values.

More information: https://manned.org/snmpstatus.

- Fetch the values:

```
snmpstatus -v {{version}} -c {{community}} {{ip}}
```

- Display help:

```
snmpstatus {{[-h|--help]}}
```

snmptable

Fetch data in tabular format.

More information: https://manned.org/snmptable.

- Fetch data:

```
snmptable -v {{version}} -c {{community}} {{ip}} {{oid}}
```

- Display help:

```
snmptable {{[-h|--help]}}
```

snmptest

Start an interactive SNMP prompt.

More information: https://manned.org/snmptest.

- Start the prompt:

```
snmptest -v {{version}} -c {{community}} {{ip}}
```

- Display help:

```
snmptest {{[-h|--help]}}
```

snmpwalk

SNMP query tool.

More information: https://manned.org/snmpwalk.

- Query the system information of a remote host using SNMPv1 and a community string:

```
snmpwalk -v 1 -c {{community}} {{ip}}
```

- Query system information on a remote host by OID using SNMPv2 on a specified port:

```
snmpwalk -v 2c -c {{community}} {{ip}}:{{port}} {{oid}}
```

- Query system information on a remote host by OID using SNMPv3 and authentication without encryption:

```
snmpwalk -v 3 -l {{authNoPriv}} -u {{username}} -a {{MD5|SHA}} -A {{passphrase}} {{ip}} {{oid}}
```

- Query system information on a remote host by OID using SNMPv3, authentication, and encryption:

```
snmpwalk -v 3 -l {{authPriv}} -u {{username}} -a {{MD5|SHA}} -A {{auth_passphrase}} -x {{DES|AES}} -X {{enc_passphrase}} {{ip}} {{oid}}
```

- Query system information on a remote host by OID using SNMPv3 without authentication or encryption:

```
snmpwalk -v 3 -l {{noAuthNoPriv}} -u {{username}} {{ip}} {{oid}}
```

- Display help:

```
snmpwalk {{[-h|--help]}}
```

snort

Open-source network intrusion detection system.

More information: https://www.snort.org/#documents.

- Capture packets with verbose output:

```
sudo snort -v -i {{interface}}
```

- Capture packets and dump application layer data with verbose output:

```
sudo snort -vd -i {{interface}}
```

- Capture packets and display link layer packet headers with verbose output:

```
sudo snort -ve -i {{interface}}
```

- Capture packets and save them in the specified directory:

```
sudo snort -i {{interface}} -l {{path/to/directory}}
```

- Capture packets according to rules and save offending packets along with alerts:

```
sudo snort -i {{interface}} -c {{path/to/rules.conf}} -l
{{path/to/directory}}
```

snowsql

SnowSQL client for Snowflake's Data Cloud.

More information: https://docs.snowflake.com/en/user-guide/snowsql.html.

- Connect to a specific instance at https://account.snowflakecomputing.com (password can be provided in prompt or configuration file):

```
snowsql --accountname {{account}} --username {{username}} --dbname {{database}} --schemaname {{schema}}
```

- Connect to an instance specified by a specific configuration file (defaults to `~/.snowsql/config`):

```
snowsql --config {{path/to/configuration_file}}
```

- Connect to the default instance using a token for multi-factor authentication:

```
snowsql --mfa-passcode {{token}}
```

- Execute a single SQL query or SnowSQL command on the default connection (useful in shell scripts):

```
snowsql --query '{{query}}'
```

- Execute commands from a specific file on the default connection:

```
snowsql --filename {{path/to/file.sql}}
```

snyk

Find vulnerabilities in your code and remediate risks.

More information: https://docs.snyk.io/snyk-cli/commands.

- Log in to your Snyk account:

```
snyk auth
```

- Test your code for any known vulnerabilities:

```
snyk test
```

- Test a local Docker image for any known vulnerabilities:

```
snyk test --docker {{docker_image}}
```

- Record the state of dependencies and any vulnerabilities on snyk.io:

```
snyk monitor
```

- Auto patch and ignore vulnerabilities:

```
snyk wizard
```

socat

Multipurpose relay (SOcket CAT).

More information: http://www.dest-unreach.org/socat/.

- Listen to a port, wait for an incoming connection and transfer data to STDIO:

```
sudo socat - TCP-LISTEN:8080,fork
```

- Listen on a port using SSL and print to `stdout`:

```
sudo socat OPENSSL-LISTEN:4433,reuseaddr,cert=./
cert.pem,cafile=./ca.cert.pem,key=./key.pem,verify=0 STDOUT
```

- Create a connection to a host and port, transfer data in STDIO to connected host:

```
sudo socat - TCP4:www.example.com:80
```

- Forward incoming data of a local port to another host and port:

```
sudo socat TCP-LISTEN:80,fork TCP4:www.example.com:80
```

- Send data with multicast routing scheme:

```
{{echo "Hello Multicast"}} | socat - UDP4-DATAGRAM:
{{224.0.0.1}}:{{5000}}
```

- Receive data from a multicast:

```
socat - UDP4-RECVFROM:{{5000}}
```

soffice

CLI for the powerful and free LibreOffice suite.

More information: https://help.libreoffice.org/latest/en-US/text/shared/guide/pdf_params.html.

- Open one or more files in read-only mode:

```
soffice --view {{path/to/file1 path/to/file2 ...}}
```

- Display the content of one or more files:

```
soffice --cat {{path/to/file1 path/to/file2 ...}}
```

- Print files using a specific printer:

```
soffice --pt {{printer_name}} {{path/to/file1 path/to/file2 ...}}
```

- Convert all `.doc` files in the current directory to PDF:

```
soffice --convert-to pdf *.doc
```

solcjs

A set of JavaScript bindings for the Solidity compiler.

More information: https://github.com/ethereum/solc-js.

- Compile a specific contract to hex:

```
solcjs --bin {{path/to/file.sol}}
```

- Compile the ABI of a specific contract:

```
solcjs --abi {{path/to/file.sol}}
```

- Specify a base path to resolve imports from:

```
solcjs --bin --base-path {{path/to/directory}} {{path/to/
file.sol}}
```

- Specify one or more paths to include containing external code:

```
solcjs --bin --include-path {{path/to/directory}} {{path/to/
file.sol}}
```

- Optimise the generated bytecode:

```
solcjs --bin --optimize {{path/to/file.sol}}
```

solo

Interact with Solo hardware security keys.

More information: https://github.com/solokeys/solo-python.

- List connected Solos:

  ```
  solo ls
  ```

- Update the currently connected Solo's firmware to the latest version:

  ```
  solo key update
  ```

- Blink the LED of a specific Solo:

  ```
  solo key wink --serial {{serial_number}}
  ```

- Generate random bytes using the currently connected Solo's secure random number generator:

  ```
  solo key rng raw
  ```

- Monitor the serial output of a Solo:

  ```
  solo monitor {{path/to/serial_port}}
  ```

sonar-scanner

A generic scanner for SonarQube projects that do not use build tools such as Maven, Gradle, or Ant.

More information: https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/.

- Scan a project with configuration file in your project's root directory named `sonar-project.properties`:

  ```
  sonar-scanner
  ```

- Scan a project using configuration file other than `sonar-project.properties`:

  ```
  sonar-scanner {{[-D|--define]}} {{project.settings=myproject.properties}}
  ```

- Print debugging information:

  ```
  sonar-scanner {{[-X|--debug]}}
  ```

- Display help:

  ```
  sonar-scanner {{[-h|--help]}}
  ```

sops

SOPS (Secrets OPerationS): a simple and flexible tool for managing secrets.

More information: https://github.com/getsops/sops.

- Encrypt a file:

```
sops -e {{path/to/file.json}} > {{path/to/file.enc.json}}
```

- Decrypt a file to `stdout`:

```
sops -d {{path/to/file.enc.json}}
```

- Update the declared keys in a `sops` file:

```
sops updatekeys {{path/to/file.enc.yaml}}
```

- Rotate data keys for a `sops` file:

```
sops -r {{path/to/file.enc.yaml}}
```

- Change the extension of the file once encrypted:

```
sops -d --input-type json {{path/to/file.enc.json}}
```

- Extract keys by naming them, and array elements by numbering them:

```
sops -d --extract '["an_array"][1]' {{path/to/file.enc.json}}
```

- Show the difference between two `sops` files:

```
diff <(sops -d {{path/to/secret1.enc.yaml}}) <(sops -d
{{path/to/secret2.enc.yaml}})
```

sort

Sort lines of text files.

More information: https://www.gnu.org/software/coreutils/manual/html_node/sort-invocation.html.

- Sort a file in ascending order:

```
sort {{path/to/file}}
```

- Sort a file in descending order:

```
sort {{[-r|--reverse]}} {{path/to/file}}
```

- Sort a file in case-insensitive way:

```
sort {{[-f|--ignore-case]}} {{path/to/file}}
```

- Sort a file using numeric rather than alphabetic order:

```
sort {{[-n|--numeric-sort]}} {{path/to/file}}
```

- Sort `/etc/passwd` by the 3rd field of each line numerically, using ":" as a field separator:

```
sort {{[-t|--field-separator]}} {{:}} {{[-k|--key]}} {{3n}} {{/etc/passwd}}
```

- As above, but when items in the 3rd field are equal, sort by the 4th field by numbers with exponents:

```
sort {{[-t|--field-separator]}} {{:}} {{[-k|--key]}} {{3,3n}} {{[-k|--key]}} {{4,4g}} {{/etc/passwd}}
```

- Sort a file preserving only unique lines:

```
sort {{[-u|--unique]}} {{path/to/file}}
```

- Sort a file, printing the output to the specified output file (can be used to sort a file in-place):

```
sort {{[-o|--output]}} {{path/to/file}} {{path/to/file}}
```

soupault

A static website generator based on HTML element tree rewriting.

It can also be used as an HTML post-processor or metadata extractor.

More information: https://soupault.app.

- Initialize a minimal website project in the current working directory:

  ```
  soupault --init
  ```

- Build a website:

  ```
  soupault
  ```

- Override default configuration file and directory locations:

  ```
  soupault --config {{config_path}} --site-dir
  {{input_directory}} --build-dir {{output_directory}}
  ```

- Extract metadata into a JSON file without generating pages:

  ```
  soupault --index-only --dump-index-json {{path/to/file.json}}
  ```

- Show the effective configuration (values from `soupault.toml` plus defaults):

  ```
  soupault --show-effective-config
  ```

source

Execute commands from a file in the current shell.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-source.

- Evaluate contents of a given file:

```
source {{path/to/file}}
```

- Evaluate contents of a given file (alternatively replacing `source` with `.`):

```
. {{path/to/file}}
```

sox

Sound eXchange: play, record and convert audio files.

Audio formats are identified by the extension.

More information: https://manned.org/sox.

- Merge two audio files into one:

```
sox {{[-m|--combine mix]}} {{path/to/input_audio1}} {{path/to/input_audio2}} {{path/to/output_audio}}
```

- Trim an audio file to the specified times:

```
sox {{path/to/input_audio}} {{path/to/output_audio}} trim {{start}} {{duration}}
```

- Normalize an audio file (adjust volume to the maximum peak level, without clipping):

```
sox --norm {{path/to/input_audio}} {{path/to/output_audio}}
```

- Reverse and save an audio file:

```
sox {{path/to/input_audio}} {{path/to/output_audio}} reverse
```

- Print statistical data of an audio file:

```
sox {{path/to/input_audio}} {{[-n|--null]}} stat
```

- Increase the volume of an audio file by 2x:

```
sox {{[-v|--volume]}} 2.0 {{path/to/input_audio}} {{path/to/output_audio}}
```

soxi

SoXI - Sound eXchange Information, display sound file metadata.

More information: https://manned.org/soxi.1.

- Display the sound file metadata:

```
soxi {{path/to/file.wav}}
```

spark

The Laravel Spark tool.

More information: https://spark.laravel.com.

- Register your API token:

  ```
  spark register {{token}}
  ```

- Display the currently registered API token:

  ```
  spark token
  ```

- Create a new Spark project:

  ```
  spark new {{project_name}}
  ```

- Create a new Spark project with Braintree stubs:

  ```
  spark new {{project_name}} --braintree
  ```

- Create a new Spark project with team-based billing stubs:

  ```
  spark new {{project_name}} --team-billing
  ```

spctoppm

Convert an Atari compressed Spectrum image to a PPM image.

More information: https://netpbm.sourceforge.net/doc/spctoppm.html.

- Convert an SPC file to a PPM image:

```
spctoppm {{path/to/input.spc}} > {{path/to/output.ppm}}
```

spectacle

KDE's screenshot utility.

More information: https://manned.org/spectacle.

- Capture a screenshot of the entire desktop:

```
spectacle
```

- Capture a screenshot of the active window:

```
spectacle {{[-a|--activewindow]}}
```

- Capture a screenshot of a specific region:

```
spectacle {{[-r|--region]}}
```

- Start another Spectacle instance so that you can screenshot Spectacle itself:

```
spectacle {{[-i|--new-instance]}}
```

speed-test

Test your internet connection speed and ping using https://speedtest.net.

More information: https://github.com/sindresorhus/speed-test.

- Test your internet connection and ping speed:

```
speed-test
```

- Print the results as JSON:

```
speed-test --json
```

- Print the results in megabytes per second (MBps):

```
speed-test --bytes
```

- Print more detailed information:

```
speed-test --verbose
```

speedcrunch

A high-precision scientific calculator.

More information: https://www.speedcrunch.org.

- Start SpeedCrunch:

  `speedcrunch`

- Copy the result of the most recent calculation:

  `<Ctrl r>`

- Open the formula book:

  `<Ctrl 1>`

- Clear the calculator of recent calculations:

  `<Ctrl n>`

- Wrap highlighted in parentheses (defaults to wrapping all if nothing selected):

  `<Ctrl p>`

- Load a speedcrunch session:

  `<Ctrl l>`

- Save a speedcrunch session:

  `<Ctrl s>`

- Toggle keypad:

  `<Ctrl k>`

speedtest-cli

Test internet bandwidth using https://speedtest.net.

See also: **`speedtest`** for the official CLI.

More information: https://github.com/sivel/speedtest-cli.

- Run a speed test:

```
speedtest-cli
```

- Run a speed test and display values in bytes, instead of bits:

```
speedtest-cli --bytes
```

- Run a speed test using `HTTPS`, instead of `HTTP`:

```
speedtest-cli --secure
```

- Run a speed test without performing download tests:

```
speedtest-cli --no-download
```

- Run a speed test and generate an image of the results:

```
speedtest-cli --share
```

- List all `speedtest.net` servers, sorted by distance:

```
speedtest-cli --list
```

- Run a speed test to a specific speedtest.net server:

```
speedtest-cli --server {{server_id}}
```

- Run a speed test and display the results as JSON (suppresses progress information):

```
speedtest-cli --json
```

speedtest-rs

An unofficial Rust-based tool for testing network speeds using speedtest.net, limited to HTTP Legacy Fallback.

More information: https://github.com/nelsonjchen/speedtest-rs.

- Run a full speed test (download and upload):

  `speedtest-rs`

- Display a list of `speedtest.net` servers sorted by distance:

  `speedtest-rs --list`

- Run a download test only:

  `speedtest-rs --no-upload`

- Run an upload test only:

  `speedtest-rs --no-download`

- Generate a shareable link to the test results image:

  `speedtest-rs --share`

- Display basic output information only:

  `speedtest-rs --simple`

speedtest

Official command-line interface for testing internet bandwidth using https://speedtest.net.

Note: Some platforms link **`speedtest`** to **`speedtest-cli`** or other tools like **`librespeed`**, which can also be installed as **`speedtest`** on certain Linux distributions.

These command examples apply only to the official client.

More information: https://www.speedtest.net/apps/cli.

- Run a speed test:

```
speedtest
```

- Run a speed test and specify the unit of the output:

```
speedtest {{[-u|--unit]}} {{auto-decimal-bits|auto-decimal-bytes|auto-binary-bits|auto-binary-bytes}}
```

- Run a speed test and specify the output format:

```
speedtest {{[-f|--format]}} {{human-readable|csv|tsv|json|jsonl|json-pretty}}
```

- Run a speed test and specify the number of decimal points to use (0 to 8, defaults to 2):

```
speedtest {{[-P|--precision]}} {{precision}}
```

- Run a speed test and print its progress (only available for output format `human-readable` and `json`):

```
speedtest {{[-p|--progress]}} {{yes|no}}
```

- List all `speedtest.net` servers, sorted by distance:

```
speedtest {{[-L|--servers]}}
```

- Run a speed test to a specific `speedtest.net` server:

```
speedtest {{[-s|--server-id]}} {{server_id}}
```

spf

The superfile – Modern terminal file manager.

More information: https://github.com/yorukot/superfile.

- Launch `spf` with a specific path:

```
spf {{path/to/directory}}
```

- Launch `spf` with multiple paths:

```
spf {{path/to/directory1 path/to/directory2 ...}}
```

- Fix hotkey settings by appending missing keys:

```
spf {{[--fh|--fix-hotkeys]}}
```

- Fix the configuration file by appending missing entries:

```
spf {{[--fch|--fix-config-file]}}
```

- Use specific configuration and hotkey files:

```
spf {{[-c|--config-file]}} {{path/to/config.toml}} {{[--hf|--hotkey-file]}} {{path/to/hotkey.toml}}
```

- Write the path of the first selected file to this file and exit:

```
spf {{[--cf|--chooser-file]}} {{tmp/chooser-result}}
```

- Show internal configuration and data directory paths:

```
spf {{[pl|path-list]}}
```

spfquery

Query Sender Policy Framework records to validate e-mail senders.

More information: https://manned.org/spfquery.

- Check if an IP address is allowed to send an e-mail from the specified e-mail address:

```
spfquery -ip {{8.8.8.8}} -sender {{sender@example.com}}
```

- Turn on debugging output:

```
spfquery -ip {{8.8.8.8}} -sender {{sender@example.com}} --debug
```

sphinx-build

Sphinx documentation generator.

More information: https://www.sphinx-doc.org/en/master/man/sphinx-build.html.

- Build documentation:

```
sphinx-build {{[-b|--builder]}} {{html|epub|text|latex|man|...}} {{path/to/source_directory}} {{path/to/build_directory}}
```

- Build documentations intended for readthedocs.io (requires the sphinx-rtd-theme pip package):

```
sphinx-build {{[-b|--builder]}} {{html}} {{path/to/docs_directory}} {{path/to/build_directory}}
```

spicetify

Customize the Spotify client UI and functionality.

More information: https://spicetify.app.

- Generate a configuration file on very first run and display help:

  ```
  spicetify
  ```

- Backup and preprocess Spotify application files:

  ```
  spicetify backup
  ```

- Print all configuration fields and values:

  ```
  spicetify config
  ```

- Change the value of a configuration field:

  ```
  spicetify config {{field}} {{value}}
  ```

- Apply the customization changes to Spotify:

  ```
  spicetify apply
  ```

- Restore Spotify to its original state:

  ```
  spicetify restore
  ```

spike

A fully featured static website generator written in JavaScript.

More information: https://spike.readme.io/docs/command-line-interface.

- Create a new project using the default template:

  ```
  spike new {{project_name}}
  ```

- Compile your project, watch for changes, and auto-reload the browser:

  ```
  spike watch
  ```

- Compile your project once to the "public" directory:

  ```
  spike compile
  ```

- Remove the output directory:

  ```
  spike clean
  ```

split

Split a file into pieces.

More information: https://www.gnu.org/software/coreutils/manual/html_node/split-invocation.html.

- Split a file, each split having 10 lines (except the last split):

```
split {{[-l|--lines]}} 10 {{path/to/file}}
```

- Split a file into 5 files. File is split such that each split has same size (except the last split):

```
split {{[-n|--number]}} 5 {{path/to/file}}
```

- Split a file with 512 bytes in each split (except the last split; use 512k for kilobytes and 512m for megabytes):

```
split {{[-b|--bytes]}} 512 {{path/to/file}}
```

- Split a file with at most 512 bytes in each split without breaking lines:

```
split {{[-C|--line-bytes]}} 512 {{path/to/file}}
```

- Split into multiple files from `stdin`:

```
gzip {{[-cd|--stdout --decompress]}} {{path/to/
compressed_file.gz}} | split {{[-l|--lines]}} {{1000}} -
{{path/to/output}}
```

sponge

Soak up the input before writing the output file.

More information: https://manned.org/sponge.

- Append file content to the source file:

```
cat {{path/to/file}} | sponge -a {{path/to/file}}
```

- Remove all lines starting with # in a file:

```
grep {{[-v|--invert-match]}} '^{{#}}' {{path/to/file}} |
sponge {{path/to/file}}
```

spotdl

Download Spotify playlists and songs along with metadata.

More information: https://github.com/spotDL/spotify-downloader.

- Download songs from the provided URLs and embed metadata:

```
spotdl {{open.spotify.com/playlist/playlistId1
open.spotify.com/track/trackId2 ...}}
```

- Start a web interface to download individual songs:

```
spotdl web
```

- Save only the metadata without downloading anything:

```
spotdl save {{open.spotify.com/playlist/playlistId1
open.spotify.com/track/trackId2 ...}} --save-file {{path/to/
save_file.spotdl}}
```

spotify_player

A TUI Spotify client that implements all features of the official Spotify app.

More information: https://github.com/aome510/spotify-player#commands.

- Start a daemon that plays music in the background:

```
spotify_player {{[-d|--daemon]}}
```

- Start the TUI (controls the daemon if available, otherwise starts its own client):

```
spotify_player
```

- Use the specified theme:

```
spotify_player {{[-t|--theme]}} {{theme_name}}
```

- Use configuration files (`app.toml`, `keymap.toml` and `theme.toml`) in the specified directory:

```
spotify_player {{[-c|--config-folder]}} {{path/to/directory}}
```

- Like the currently playing track:

```
spotify_player like
```

- Display a list of keybindings:

```
<?>
```

spottopgm

Convert a SPOT satellite image to PGM format.

More information: https://netpbm.sourceforge.net/doc/spottopgm.html.

- Convert the specified SPOT image to PGM format:

```
spottopgm {{path/to/file.spot}} > {{path/to/output.pgm}}
```

- Extract the specified color channel:

```
spottopgm -{{1|2|3}} {{path/to/file.spot}} > {{path/to/
output.pgm}}
```

- Extract the specified rectangle from the input image:

```
spottopgm {{first_col first_row last_col last_row}} {{path/
to/file.spot}} > {{path/to/output.pgm}}
```

sputoppm

Convert an Atari uncompressed Spectrum image to a PPM image.

More information: https://netpbm.sourceforge.net/doc/sputoppm.html.

- Convert an SPU file to a PPM image:

```
sputoppm {{path/to/input.spu}} > {{path/to/output.ppm}}
```

sqlite-utils

Manipulate SQLite databases in a number of different ways.

More information: https://sqlite-utils.datasette.io/en/stable/cli.html.

- Create a database:

```
sqlite-utils create-database {{path/to/database.db}}
```

- Create a table:

```
sqlite-utils create-table {{path/to/database.db}}
{{table_name}} {{id integer name text height float photo blob
--pk id}}
```

- List tables:

```
sqlite-utils tables {{path/to/database.db}}
```

- Upsert a record:

```
{{echo '[ {"id": 1, "name": "Linus Torvalds"}, {"id": 2,
"name": "Steve Wozniak"}, {"id": 3, "name": "Tony
Hoare"} ]'}} | sqlite-utils upsert {{path/to/database.db}}
{{table_name}} - {{--pk id}}
```

- Select records:

```
sqlite-utils rows {{path/to/database.db}} {{table_name}}
```

- Delete a record:

```
sqlite-utils query {{path/to/database.db}} "{{delete from
table_name where name = 'Tony Hoare'}}"
```

- Drop a table:

```
sqlite-utils drop-table {{path/to/database.db}}
{{table_name}}
```

- Display help:

```
sqlite-utils {{[-h|--help]}}
```

sqlite3

Interface to SQLite 3, which is a self-contained file-based embedded SQL engine.

More information: https://sqlite.org/cli.html.

- Start an interactive shell with a new database:

```
sqlite3
```

- Open an interactive shell against an existing database:

```
sqlite3 {{path/to/database.sqlite3}}
```

- Execute an SQL statement against a database and then exit:

```
sqlite3 {{path/to/database.sqlite3}} '{{SELECT * FROM some_table;}}'
```

sqlmap

Detect and exploit SQL injection flaws.

More information: https://sqlmap.org.

- Run sqlmap against a single target URL:

```
python sqlmap.py {{[-u|--url]}} "{{http://www.example.com/vuln.php?id=1}}"
```

- Send data in a POST request (`--data` implies POST request):

```
python sqlmap.py {{[-u|--url]}} "{{http://www.example.com/vuln.php}}" --data="{{id=1}}"
```

- Change the parameter delimiter (& is the default):

```
python sqlmap.py {{[-u|--url]}} "{{http://www.example.com/vuln.php}}" --data="{{query=foobar;id=1}}" --param-del="{{;}}"
```

- Select a random `User-Agent` from `./txt/user-agents.txt` and use it:

```
python sqlmap.py {{[-u|--url]}} "{{http://www.example.com/vuln.php}}" --random-agent
```

- Provide user credentials for HTTP protocol authentication:

```
python sqlmap.py {{[-u|--url]}} "{{http://www.example.com/vuln.php}}" --auth-type {{Basic}} --auth-cred "{{testuser:testpass}}"
```

sqlx

Utility for SQLx, the Rust SQL toolkit.

More information: https://github.com/launchbadge/sqlx/blob/main/sqlx-cli/README.md.

- Create the database specified in the DATABASE_URL environment variable:

```
sqlx database create
```

- Drop the specified database:

```
sqlx database drop {{[-D|--database-url]}} {{database_url}}
```

- Create a new pair of up and down migration files with the given description in the "migrations" directory:

```
sqlx migrate add -r {{migration_description}}
```

- Run all pending migrations for the specified database:

```
sqlx migrate run {{[-D|--database-url]}} {{database_url}}
```

- Revert the latest migration for the specified database:

```
sqlx migrate revert {{[-D|--database-url]}} {{database_url}}
```

sqsc

An AWS Simple Queue Service client.

More information: https://github.com/yongfei25/sqsc.

- List all queues:

```
sqsc lq {{queue_prefix}}
```

- List all messages in a queue:

```
sqsc ls {{queue_name}}
```

- Copy all messages from one queue to another:

```
sqsc cp {{source_queue}} {{destination_queue}}
```

- Move all messages from one queue to another:

```
sqsc mv {{source_queue}} {{destination_queue}}
```

- Describe a queue:

```
sqsc describe {{queue_name}}
```

- Query a queue with SQL syntax:

```
sqsc query "SELECT body FROM {{queue_name}} WHERE body LIKE '%user%'"
```

- Pull all messages from a queue into a local SQLite database in your present working directory:

```
sqsc pull {{queue_name}}
```

sr

This command is an alias of **surfraw**.

- View documentation for the original command:

```
tldr surfraw
```

srftopam

Convert an SRF file to Netpbm format.

See also: **pamtosrf**.

More information: https://netpbm.sourceforge.net/doc/srftopam.html.

- Convert an SRF file to Netpbm format:

```
srftopam {{path/to/file.srf}} > {{path/to/output.pam}}
```

srm

Securely remove files or directories.

Overwrites the existing data one or multiple times. Drop in replacement for rm.

More information: https://srm.sourceforge.net/srm.html.

- Remove a file after a single-pass overwriting with random data:

```
srm {{[-s|--simple]}} {{path/to/file}}
```

- Remove a file after seven passes of overwriting with random data:

```
srm -m {{path/to/file}}
```

- Recursively remove a directory and its contents overwriting each file with a single-pass of random data:

```
srm {{[-r|--recursive]}} {{[-s|--simple]}} {{path/to/directory}}
```

- Prompt before every removal:

```
srm {{[-i|--interactive]}} {{\*}}
```

srun

Run a command under the Slurm workload manager.

More information: https://slurm.schedmd.com/srun.html.

- Run a simple command interactively:

```
srun hostname
```

- Run a job with 4 tasks (CPUs):

```
srun {{[-n|--ntasks]}} 4 {{path/to/program}}
```

- Allocate 8 GB of memory:

```
srun --mem 8G {{path/to/program}}
```

- Run a job on a specific partition:

```
srun {{[-p|--partition]}} gpu {{path/to/program}}
```

- Run a job and save the output to a file:

```
srun {{path/to/program}} > {{path/to/output}}
```

ss-local

Run a Shadowsocks client as a SOCKS5 proxy.

More information: https://github.com/shadowsocks/shadowsocks-libev/blob/master/doc/ss-local.asciidoc.

- Run a Shadowsocks proxy by specifying the host, server port, local port, password, and encryption method:

```
ss-local -s {{host}} -p {{server_port}} -l {{local port}} -k {{password}} -m {{encrypt_method}}
```

- Run a Shadowsocks proxy by specifying the configuration file:

```
ss-local -c {{path/to/config_file.json}}
```

- Use a plugin to run the proxy client:

```
ss-local --plugin {{plugin_name}} --plugin-opts {{plugin_options}}
```

- Enable TCP fast open:

```
ss-local --fast-open
```

ssh-add

Manage loaded SSH keys in the **`ssh-agent`**.

Ensure that **`ssh-agent`** is up and running for the keys to be loaded in it.

More information: https://man.openbsd.org/ssh-add.

- Add the default SSH keys in `~/.ssh` to the ssh-agent:

```
ssh-add
```

- Add a specific key to the ssh-agent:

```
ssh-add {{path/to/private_key}}
```

- List fingerprints of currently loaded keys:

```
ssh-add -l
```

- Delete a key from the ssh-agent:

```
ssh-add -d {{path/to/private_key}}
```

- Delete all currently loaded keys from the ssh-agent:

```
ssh-add -D
```

- Add a key to the ssh-agent and the keychain:

```
ssh-add -K {{path/to/private_key}}
```

ssh-agent

Spawn an SSH Agent process.

An SSH Agent holds SSH keys decrypted in memory until removed or the process is killed.

See also: **`ssh-add`**, which can add and manage keys held by an SSH Agent.

More information: https://man.openbsd.org/ssh-agent.

- Start an SSH Agent for the current shell:

```
eval $(ssh-agent)
```

- Kill the currently running agent:

```
ssh-agent -k
```

ssh-copy-id

Install your public key in a remote machine's authorized_keys.

More information: https://manned.org/ssh-copy-id.

- Copy your keys to the remote machine:

```
ssh-copy-id {{username}}@{{remote_host}}
```

- Copy the given public key to the remote:

```
ssh-copy-id -i {{path/to/certificate}} {{username}}
@{{remote_host}}
```

- Copy the given public key to the remote with specific port:

```
ssh-copy-id -i {{path/to/certificate}} -p {{port}}
{{username}}@{{remote_host}}
```

ssh-keygen

Generate SSH keys used for authentication, password-less logins, and other things.

See also: **`ssh-copy-id`** for installing SSH keys on remote hosts.

More information: https://man.openbsd.org/ssh-keygen.

- Generate a key interactively:

```
ssh-keygen
```

- Generate an ed25519 key with 32 key derivation function rounds and save the key to a specific file:

```
ssh-keygen -t {{ed25519}} -a {{32}} -f {{~/.ssh/filename}}
```

- Generate an RSA 4096-bit key with email as a comment:

```
ssh-keygen -t {{rsa}} -b {{4096}} -C "{{comment|email}}"
```

- Remove the keys of a host from the known_hosts file (useful when a known host has a new key):

```
ssh-keygen -R {{remote_host}}
```

- Retrieve the fingerprint of a key in MD5 Hex:

```
ssh-keygen -l -E {{md5}} -f {{~/.ssh/filename}}
```

- Change the password of a key:

```
ssh-keygen -p -f {{~/.ssh/filename}}
```

- Change the type of the key format (for example from OPENSSH format to PEM), the file will be rewritten in-place:

```
ssh-keygen -p -N "" -m {{PEM}} -f {{~/.ssh/
OpenSSH_private_key}}
```

- Retrieve public key from secret key:

```
ssh-keygen -y -f {{~/.ssh/OpenSSH_private_key}}
```

ssh-keyscan

Get the public SSH keys of remote hosts.

More information: https://man.openbsd.org/ssh-keyscan.

- Retrieve all public SSH keys of a remote host:

```
ssh-keyscan {{hostname}}
```

- Retrieve all public SSH keys of a remote host listening on a specific port:

```
ssh-keyscan -p {{port}} {{hostname}}
```

- Retrieve certain types of public SSH keys of a remote host:

```
ssh-keyscan -t {{rsa,dsa,ecdsa,ed25519}} {{hostname}}
```

- Manually update the SSH known_hosts file with the fingerprint of a given host:

```
ssh-keyscan -H {{hostname}} >> ~/.ssh/known_hosts
```

ssh

Secure Shell is a protocol used to securely log onto remote systems.

It can be used for logging or executing commands on a remote server.

More information: https://man.openbsd.org/ssh.

- Connect to a remote server:

```
ssh {{username}}@{{remote_host}}
```

- Connect to a remote server with a specific [i]dentity (private key):

```
ssh -i {{path/to/key_file}} {{username}}@{{remote_host}}
```

- Connect to a remote server with IP `10.0.0.1` and using a specific [p]ort (Note: `10.0.0.1` can be shortened to `10.1`):

```
ssh {{username}}@10.0.0.1 -p {{2222}}
```

- Run a command on a remote server with a [t]ty allocation allowing interaction with the remote command:

```
ssh {{username}}@{{remote_host}} -t {{command}} {{command_arguments}}
```

- SSH tunneling: [D]ynamic port forwarding (SOCKS proxy on `localhost:1080`):

```
ssh -D {{1080}} {{username}}@{{remote_host}}
```

- SSH tunneling: Forward a specific port (`localhost:9999` to `example.org:80`) along with disabling pseudo-[T]ty allocation and executio[N] of remote commands:

```
ssh -L {{9999}}:{{example.org}}:{{80}} -N -T {{username}}@{{remote_host}}
```

- SSH [J]umping: Connect through a jumphost to a remote server (Multiple jump hops may be specified separated by comma characters):

```
ssh -J {{username}}@{{jump_host}} {{username}}@{{remote_host}}
```

- Close a hanged session:

```
<Enter><~><.>
```

sshd

Secure Shell Daemon - allows remote machines to securely log in to the current machine.

Remote machines can execute commands as it is executed at this machine.

More information: https://man.openbsd.org/sshd.

- Start daemon in the background:

```
sshd
```

- Run sshd in the foreground:

```
sshd -D
```

- Run with verbose output (for debugging):

```
sshd -D -d
```

- Run on a specific port:

```
sshd -p {{port}}
```

sshfs

Filesystem client based on SSH.

More information: https://github.com/libfuse/sshfs.

- Mount remote directory:

```
sshfs {{username}}@{{remote_host}}:{{remote_directory}} {{mountpoint}}
```

- Unmount remote directory:

```
umount {{mountpoint}}
```

- Mount remote directory from server with specific port:

```
sshfs {{username}}@{{remote_host}}:{{remote_directory}} -p {{2222}}
```

- Use compression:

```
sshfs {{username}}@{{remote_host}}:{{remote_directory}} -C
```

- Follow symbolic links:

```
sshfs -o follow_symlinks {{username}}@{{remote_host}}:{{remote_directory}} {{mountpoint}}
```

sshpass

An SSH password provider.

It works by creating a TTY, feeding the password into it, and then redirecting **`stdin`** to the SSH session.

More information: https://manned.org/sshpass.

- Connect to a remote server using a password supplied on a file descriptor (in this case, `stdin`):

```
sshpass -d {{0}} ssh {{user}}@{{hostname}}
```

- Connect to a remote server with the password supplied as an option, and automatically accept unknown SSH keys:

```
sshpass -p {{password}} ssh -o StrictHostKeyChecking=no
{{user}}@{{hostname}}
```

- Connect to a remote server using the first line of a file as the password, automatically accept unknown SSH keys, and launch a command:

```
sshpass -f {{path/to/file}} ssh -o StrictHostKeyChecking=no
{{user}}@{{hostname}} "{{command}}"
```

sshuttle

Transparent proxy server that tunnels traffic over an SSH connection.

Doesn't require root or any special setup on the remote SSH server, though root access on the local machine is prompted for.

More information: https://manned.org/sshuttle.

- Forward all IPv4 TCP traffic via a remote SSH server:

```
sshuttle {{[-r|--remote]}} {{username}}@{{sshserver}}
{{0.0.0.0/0}}
```

- Also forward all DNS traffic to the server's default DNS resolver:

```
sshuttle --dns {{[-r|--remote]}} {{username}}@{{sshserver}}
{{0.0.0.0/0}}
```

- Forward all traffic except that which is bound for a specific subnet:

```
sshuttle {{[-r|--remote]}} {{username}}@{{sshserver}}
{{0.0.0.0/0}} {{[-x|--exclude]}} {{192.168.0.1/24}}
```

- Use the tproxy method to forward all IPv4 and IPv6 traffic:

```
sshuttle --method tproxy {{[-r|--remote]}} {{username}}
@{{sshserver}} {{0.0.0.0/0}} {{::/0}} {{[-x|--exclude]}}
{{your_local_ip_address}} {{[-x|--exclude]}}
{{ssh_server_ip_address}}
```

sslscan

Check SSL/TLS protocols and ciphers supported by a server.

More information: https://github.com/rbsec/sslscan.

- Test a server on port 443:

```
sslscan {{example.com}}
```

- Test a specified port:

```
sslscan {{example.com}}:{{465}}
```

- Show certificate information:

```
testssl --show-certificate {{example.com}}
```

st-flash

Flash binary files to STM32 ARM Cortex microcontrollers.

More information: https://manned.org/st-flash.

- Read 4096 bytes from the device starting from 0x8000000:

```
st-flash read {{firmware}}.bin 0x8000000 4096
```

- Write firmware to device starting from 0x8000000:

```
st-flash write {{firmware}}.bin 0x8000000
```

- Erase firmware from device:

```
st-flash erase
```

st-info

Get information about connected STLink and STM32 devices.

More information: https://github.com/texane/stlink.

- Display amount of program memory available:

```
st-info --flash
```

- Display amount of SRAM memory available:

```
st-info --sram
```

- Display summarized information of the device:

```
st-info --probe
```

st-util

Run GDB (GNU Debugger) server to interact with STM32 ARM Cortex microcontoller.

More information: https://github.com/texane/stlink.

- Run GDB server on port 4500:

```
st-util {{[-p|--listen_port]}} {{4500}}
```

- Connect to GDB server within `gdb`:

```
target extended-remote {{localhost}}:{{4500}}
```

- Write firmware to device:

```
load {{firmware.elf}}
```

st

Print basic descriptive statistics from input numbers.

More information: https://github.com/nferraz/st.

- Print count, min, max, sum, mean, and standard deviation for numbers in a file:

```
st {{path/to/file}}
```

- Print statistics from `stdin`:

```
cat {{path/to/file}} | st
```

- Print only the sum of the numbers:

```
st {{[-s|--sum]}} {{path/to/file}}
```

- Print only the mean of the numbers:

```
st {{[-m|--mean]}} {{path/to/file}}
```

- Print only the standard deviation:

```
st {{[-s|--sd]}} {{path/to/file}}
```

- Transpose output (keys in one column, values in another):

```
st {{[-t|--transpose]}} {{path/to/file}}
```

st4topgm

Convert an SBIG ST-4 file to PGM.

More information: https://netpbm.sourceforge.net/doc/st4topgm.html.

- Convert an SBIG ST-4 file to a PGM file:

```
st4topgm {{path/to/input_file.st4}} > {{path/to/output.pgm}}
```

stack

Manage Haskell projects.

More information: https://github.com/commercialhaskell/stack.

- Create a new package:

```
stack new {{package}} {{template}}
```

- Compile a package:

```
stack build
```

- Run tests inside a package:

```
stack test
```

- Compile a project and re-compile every time a file changes:

```
stack build --file-watch
```

- Compile a project and execute a command after compilation:

```
stack build --exec "{{command}}"
```

- Run a program and pass an argument to it:

```
stack exec {{program}} -- {{argument}}
```

standard-version

Automate versioning and changelog generation, with SemVer and Conventional Commits.

More information: https://github.com/conventional-changelog/standard-version.

- Update the changelog file and tag a release:

  ```
  standard-version
  ```

- Tag a release without bumping the version:

  ```
  standard-version --first-release
  ```

- Update the changelog and tag an alpha release:

  ```
  standard-version --prerelease alpha
  ```

- Update the changelog and tag a specific release type:

  ```
  standard-version --release-as {{major|minor|patch}}
  ```

- Tag a release, preventing hooks from being verified during the commit step:

  ```
  standard-version --no-verify
  ```

- Tag a release committing all staged changes, not just files affected by `standard-version`:

  ```
  standard-version --commit-all
  ```

- Update a specific changelog file and tag a release:

  ```
  standard-version --infile {{path/to/file.md}}
  ```

- Display the release that would be performed without performing them:

  ```
  standard-version --dry-run
  ```

standard

The JavaScript Standard Style tool for linting and fixing JavaScript code.

More information: https://standardjs.com.

- Lint all JavaScript source files in the current directory:

  ```
  standard
  ```

- Lint specific JavaScript file(s):

  ```
  standard {{path/to/file1 path/to/file2 ...}}
  ```

- Apply automatic fixes during linting:

  ```
  standard --fix
  ```

- Declare any available global variables:

  ```
  standard --global {{variable}}
  ```

- Use a custom ESLint plugin when linting:

  ```
  standard --plugin {{plugin}}
  ```

- Use a custom JS parser when linting:

  ```
  standard --parser {{parser}}
  ```

- Use a custom ESLint environment when linting:

  ```
  standard --env {{environment}}
  ```

starship init

Print shell integration code for starship.

More information: https://starship.rs.

- Print the starship integration code for the specified shell:

```
starship init {{bash|elvish|fish|ion|powershell|tcsh|zsh}}
```

- Append the `starship` integration code to `~/.bashrc`:

```
starship init {{bash}} >> {{~/.bashrc}}
```

- Append the `starship` integration code to `~/.zshrc`:

```
starship init {{zsh}} >> {{~/.zshrc}}
```

- Display help:

```
starship init --help
```

starship

The minimal, blazing-fast, and infinitely customizable prompt for any shell.

Some subcommands such as **`init`** have their own usage documentation.

More information: https://starship.rs.

- Print the starship integration code for the specified shell:

```
starship init {{bash|elvish|fish|ion|powershell|tcsh|zsh|nu|xonsh|cmd}}
```

- Explain each part of the current prompt and show the time taken to render them:

```
starship explain
```

- Print the computed starship configuration (use `--default` to print default configuration instead):

```
starship print-config
```

- List supported modules:

```
starship module --list
```

- Edit the starship configuration in the default editor:

```
starship config
```

- Create a bug report GitHub issue pre-populated with information about the system and starship configuration:

```
starship bug-report
```

- Print the completion script for the specified shell:

```
starship completions {{bash|elvish|fish|powershell|zsh}}
```

- Display help for a subcommand:

```
starship {{subcommand}} --help
```

stat

Display file and filesystem information.

See also: **`file`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/stat-invocation.html.

- Display properties about a specific file such as size, permissions, creation and access dates among others:

```
stat {{path/to/file}}
```

- Display properties about a specific file, only showing the raw result data without labels:

```
stat {{[-t|--terse]}} {{path/to/file}}
```

- Display information about the filesystem where a specific file is located:

```
stat {{[-f|--file-system]}} {{path/to/file}}
```

- Show only octal file permissions:

```
stat {{[-c|--format]}} "%a %n" {{path/to/file}}
```

- Show the owner and group of a specific file:

```
stat {{[-c|--format]}} "%U %G" {{path/to/file}}
```

- Show the size of a specific file in bytes:

```
stat {{[-c|--format]}} "%s %n" {{path/to/file}}
```

stdbuf

Run a command with modified buffering operations for its standard streams.

More information: https://www.gnu.org/software/coreutils/manual/html_node/stdbuf-invocation.html.

- Change `stdin` buffer size to 512 KiB:

```
stdbuf {{[-i|--input]}} 512K {{command}}
```

- Change `stdout` buffer to line-buffered:

```
stdbuf {{[-o|--output]}} L {{command}}
```

- Change `stderr` buffer to unbuffered:

```
stdbuf {{[-e|--error]}} 0 {{command}}
```

steam

Video game platform by Valve.

More information: https://developer.valvesoftware.com/wiki/Command_Line_Options.

- Launch Steam, printing debug messages to `stdout`:

```
steam
```

- Launch Steam and enable its in-app debug console tab:

```
steam -console
```

- Launch Steam without opening the GUI:

```
steam -silent
```

- Enable and open the Steam console tab in a running Steam instance:

```
steam steam://open/console
```

- Log into Steam with the specified credentials:

```
steam -login {{username}} {{password}}
```

- Launch Steam in Big Picture Mode:

```
steam -tenfoot
```

- Exit Steam:

```
steam -shutdown
```

steamcmd

A command-line version of the Steam client.

More information: https://manned.org/steamcmd.

- Install or update an application anonymously:

```
steamcmd +login {{anonymous}} +app_update {{appid}} +quit
```

- Install or update an application using the specified credentials:

```
steamcmd +login {{username}} +app_update {{appid}} +quit
```

- Install an application for a specific platform:

```
steamcmd +@sSteamCmdForcePlatformType {{windows}} +login
{{anonymous}} +app_update {{appid}} validate +quit
```

- Clear cached login credentials for a user:

```
steamcmd +login {{username}} +logout +quit
```

step

An easy-to-use CLI tool for building, operating, and automating Public Key Infrastructure (PKI) systems and workflows.

See also: **`openssl`**.

More information: https://smallstep.com/docs/step-cli/.

- Inspect the contents of a certificate:

```
step certificate inspect {{path/to/certificate.crt}}
```

- Create a root CA certificate and a key (append `--no-password --insecure` to skip private key password protection):

```
step certificate create "{{Example Root CA}}" {{path/to/root-ca.crt}} {{path/to/root-ca.key}} --profile root-ca
```

- Generate a certificate for a specific hostname and sign it with the root CA (generating a CSR can be skipped for simplification):

```
step certificate create {{hostname.example.com}} {{path/to/hostname.crt}} {{path/to/hostname.key}} --profile leaf --ca {{path/to/root-ca.crt}} --ca-key {{path/to/root-ca.key}}
```

- Verify a certificate chain:

```
step certificate verify {{path/to/hostname.crt}} --roots {{path/to/root-ca.crt}} --verbose
```

- Convert a PEM format certificate to DER and write it to disk:

```
step certificate format {{path/to/certificate.pem}} --out {{path/to/certificate.der}}
```

- Install or uninstall a root certificate in the system's default trust store:

```
step certificate {{install|uninstall}} {{path/to/root-ca.crt}}
```

- Create a RSA/EC private and public keypair (append `--no-password --insecure` to skip private key password protection):

```
step crypto keypair {{path/to/public_key}} {{path/to/private_key}} --kty {{RSA|EC}}
```

- Display help for subcommands:

```
step {{path|base64|certificate|completion|context|crl|crypto|
oauth|ca|beta|ssh}} --help
```

stern

Tail multiple pods and containers from Kubernetes.

More information: https://github.com/stern/stern.

- Tail all pods within a current namespace:

```
stern .
```

- Tail all pods with a specific status:

```
stern . --container-state {{running|waiting|terminated}}
```

- Tail all pods that matches a given `regex`:

```
stern {{pod_query}}
```

- Tail matched pods from all namespaces:

```
stern {{pod_query}} --all-namespaces
```

- Tail matched pods from 15 minutes ago:

```
stern {{pod_query}} --since {{15m}}
```

- Tail matched pods with a specific label:

```
stern {{pod_query}} --selector {{release=canary}}
```

stl2gts

Convert STL files into the GTS (GNU triangulated surface library) file format.

More information: https://manned.org/stl2gts.

- Convert an STL file to a GTS file:

```
stl2gts < {{path/to/file.stl}} > {{path/to/file.gts}}
```

- Convert an STL file to a GTS file and revert face normals:

```
stl2gts < {{path/to/file.stl}} --revert > {{path/to/
file.gts}}
```

- Convert an STL file to a GTS file and do not merge vertices:

```
stl2gts < {{path/to/file.stl}} --nomerge > {{path/to/
file.gts}}
```

- Convert an STL file to a GTS file and display surface statistics:

```
stl2gts < {{path/to/file.stl}} --verbose > {{path/to/
file.gts}}
```

- Display help:

```
stl2gts --help
```

stolonctl

CLI for Stolon, a cloud native PostgreSQL manager for PostgreSQL high availability.

More information: https://github.com/sorintlab/stolon.

- Get cluster status:

```
stolonctl --cluster-name {{cluster_name}} --store-backend
{{store_backend}} --store-endpoints {{store_endpoints}}
status
```

- Get cluster data:

```
stolonctl --cluster-name {{cluster_name}} --store-backend
{{store_backend}} --store-endpoints {{store_endpoints}}
clusterdata
```

- Get cluster specification:

```
stolonctl --cluster-name {{cluster_name}} --store-backend
{{store_backend}} --store-endpoints {{store_endpoints}} spec
```

- Update cluster specification with a patch in JSON format:

```
stolonctl --cluster-name {{cluster_name}} --store-backend
{{store_backend}} --store-endpoints {{store_endpoints}}
update --patch '{{cluster_spec}}'
```

stormlock

Centralized locking system.

More information: https://github.com/tmccombs/stormlock.

- Acquire a lease for resource:

```
stormlock acquire {{resource}}
```

- Release the given lease for the given resource:

```
stormlock release {{resource}} {{lease_id}}
```

- Show information on the current lease for a resource, if any:

```
stormlock current {{resource}}
```

- Test if a lease for given resource is currently active:

```
stormlock is-held {{resource}} {{lease_id}}
```

stow

Symlink manager.

Often used to manage dotfiles.

See also: **`chezmoi`**, **`tuckr`**, **`vcsh`**, **`homeshick`**.

More information: https://www.gnu.org/software/stow/manual/stow.html#Invoking-Stow.

- Symlink all files recursively to a given directory:

```
stow {{[-t|--target]}} {{path/to/target_directory}} {{file1 directory1 file2 directory2}}
```

- Delete symlinks recursively from a given directory:

```
stow {{[-D|--delete]}} {{[-t|--target]}} {{path/to/target_directory}} {{file1 directory1 file2 directory2}}
```

- Simulate to see what the result would be like:

```
stow {{[-n|--simulate]}} {{[-t|--target]}} {{path/to/target_directory}} {{file1 directory1 file2 directory2}}
```

- Delete and resymlink:

```
stow {{[-R|--restow]}} {{[-t|--target]}} {{path/to/target_directory}} {{file1 directory1 file2 directory2}}
```

- Exclude files matching a `regex`:

```
stow --ignore={{regex}} {{[-t|--target]}} {{path/to/target_directory}} {{file1 directory1 file2 directory2}}
```

streamlink

Extracts streams from various services and pipes them into a video player of choice.

More information: https://streamlink.github.io.

- Attempt to extract streams from the URL specified, and if it's successful, print out a list of available streams to choose from:

```
streamlink {{example.com/stream}}
```

- Open a stream with the specified quality:

```
streamlink {{example.com/stream}} {{720p60}}
```

- Select the highest or lowest available quality:

```
streamlink {{example.com/stream}} {{best|worst}}
```

- Use a specific player to feed stream data to (VLC is used by default if found):

```
streamlink --player={{mpv}} {{example.com/stream}} {{best}}
```

- Skip a specific amount of time from the beginning of the stream. For live streams, this is a negative offset from the end of the stream (rewind):

```
streamlink --hls-start-offset {{[HH:]MM:SS}} {{example.com/
stream}} {{best}}
```

- Skip to the beginning of a live stream, or as far back as possible:

```
streamlink --hls-live-restart {{example.com/stream}} {{best}}
```

- Write stream data to a file instead of playing it:

```
streamlink --output {{path/to/file.ts}} {{example.com/
stream}} {{best}}
```

- Open the stream in the player, while at the same time writing it to a file:

```
streamlink --record {{path/to/file.ts}} {{example.com/
stream}} {{best}}
```

streamlit

Framework for creating interactive, data-driven web apps in Python.

More information: https://docs.streamlit.io/develop/api-reference/cli.

- Check for the Streamlit installation:

  ```
  streamlit hello
  ```

- Run a Streamlit application:

  ```
  streamlit run {{project_name}}
  ```

- Display help:

  ```
  streamlit --help
  ```

- Display version:

  ```
  streamlit --version
  ```

stress-ng

Load and stress a Linux system in various ways (CPU, memory, I/O, etc.).

Useful for benchmarking, hardware validation, and stability testing.

More information: https://manned.org/stress-ng.

- Stress all CPUs with 4 workers for 60 seconds:

```
stress-ng {{[-c|--cpu]}} 4 {{[-t|--timeout]}} 60s
```

- Stress virtual memory with 2 workers for 30 seconds:

```
stress-ng {{[-m|--vm]}} 2 --vm-bytes {{512M}} {{[-t|--timeout]}} 30s
```

- Stress the I/O subsystem with 3 workers for 45 seconds:

```
stress-ng {{[-i|--io]}} 3 {{[-t|--timeout]}} 45s
```

- Run all stress tests for 2 minutes:

```
stress-ng {{[-a|--all]}} {{1}} {{[-t|--timeout]}} 2m
```

stressapptest

Userspace memory and IO test.

More information: https://github.com/stressapptest/stressapptest.

- Test the given amount of memory (in Megabytes):

```
stressapptest -M {{memory}}
```

- Test memory as well as I/O for the given file:

```
stressapptest -M {{memory}} -f {{path/to/file}}
```

- Test specifying the verbosity level, where 0=lowest, 20=highest, 8=default:

```
stressapptest -M {{memory}} -v {{level}}
```

strings

Find printable strings in an object file or binary.

More information: https://manned.org/strings.

- Print all strings in a binary:

```
strings {{path/to/file}}
```

- Limit results to strings at least n characters long:

```
strings {{[-n|--bytes]}} {{n}} {{path/to/file}}
```

- Prefix each result with its offset within the file:

```
strings {{[-t|--radix]}} d {{path/to/file}}
```

- Prefix each result with its offset within the file in hexadecimal:

```
strings {{[-t|--radix]}} x {{path/to/file}}
```

strip-nondeterminism

Remove non-deterministic information (e.g. timestamps) from files.

More information: https://salsa.debian.org/reproducible-builds/strip-nondeterminism.

- Strip nondeterministic information from a file:

```
strip-nondeterminism {{path/to/file}}
```

- Strip nondeterministic information from a file manually specifying the filetype:

```
strip-nondeterminism --type {{filetype}} {{path/to/file}}
```

- Strip nondeterministic information from a file; instead of removing timestamps set them to the specified UNIX timestamp:

```
strip-nondeterminism --timestamp {{unix_timestamp}} {{path/to/file}}
```

stripe

Interact with a Stripe account.

More information: https://docs.stripe.com/stripe-cli.

- Follow the logs of activity on the account:

  ```
  stripe logs tail
  ```

- Listen for events, filtering on events with the name `charge.succeeded` and forwarding them to localhost:3000/events:

  ```
  stripe listen --events="{{charge.succeeded}}" --forward-to="{{localhost:3000/events}}"
  ```

- Send a test webhook event:

  ```
  stripe trigger {{charge.succeeded}}
  ```

- Create a customer:

  ```
  stripe customers create --email="{{test@example.com}}" --name="{{Jenny Rosen}}"
  ```

- Print to JSON:

  ```
  stripe listen --print-json
  ```

stty

Set options for a terminal device interface.

More information: https://www.gnu.org/software/coreutils/manual/html_node/stty-invocation.html.

- Display current terminal size:

```
stty size
```

- Display all settings for the current terminal:

```
stty {{[-a|--all]}}
```

- Set the number of rows or columns:

```
stty {{rows|cols}} {{count}}
```

- Get the actual transfer speed of a device:

```
stty {{[-F|--file]}} {{path/to/device_file}} speed
```

- Reset all modes to reasonable values for the current terminal:

```
stty sane
```

- Switch between raw and normal mode:

```
stty {{raw|cooked}}
```

- Turn character echoing off or on:

```
stty {{-echo|echo}}
```

- Display help:

```
stty --help
```

stun

Classic STUN client.

More information: https://manned.org/stun.1.

- Make a STUN request:

```
stun {{stun.1und1.de}}
```

- Make a STUN request and specify the source port:

```
stun {{stun.1und1.de}} -p {{4302}}
```

stylua

An opinionated Lua code formatter.

More information: https://github.com/JohnnyMorganz/StyLua.

- Auto-format a file or an entire directory:

```
stylua {{path/to/file_or_directory}}
```

- Check if a specific file has been formatted:

```
stylua --check {{path/to/file}}
```

- Run with a specific configuration file:

```
stylua --config-path {{path/to/config_file}} {{path/to/file}}
```

- Format code from `stdin` and output to `stdout`:

```
stylua - < {{path/to/file.lua}}
```

- Format a file or directory using spaces and preferring single quotes:

```
stylua --indent-type {{Spaces}} --quote-style
{{AutoPreferSingle}} {{path/to/file_or_directory}}
```

subfinder

Discover valid subdomains for websites.

Designed as a passive framework to be useful for bug bounties and safe for penetration testing.

More information: https://docs.projectdiscovery.io/opensource/subfinder/usage.

- Find subdomains for a specific domain:

```
subfinder {{[-d|-domain]}} {{example.com}}
```

- Show only the subdomains found:

```
subfinder -silent {{[-d|-domain]}} {{example.com}}
```

- Show only active subdomains:

```
subfinder {{[-nW|-active]}} {{[-d|-domain]}} {{example.com}}
```

- Use all sources for enumeration:

```
subfinder -all {{[-d|-domain]}} {{example.com}}
```

- Use a given comma-separated list of [r]esolvers:

```
subfinder -r {{8.8.8.8,1.1.1.1,...}} {{[-d|-domain]}}
{{example.com}}
```

subl

Sublime Text editor.

More information: https://www.sublimetext.com/docs/command_line.html.

- Open the current directory in Sublime Text:

```
subl .
```

- Open a file or directory in Sublime Text:

```
subl {{path/to/file_or_directory}}
```

- Open a file and jump to a specific line number:

```
subl {{path/to/file}}:{{line_number}}
```

- Open a file or directory in the currently open window:

```
subl {{[-a|--add]}} {{path/to/file_or_directory}}
```

- Open a file or directory in a new window:

```
subl {{[-n|--new-window]}} {{path/to/file_or_directory}}
```

subliminal

Python-based subtitle downloader.

More information: https://github.com/Diaoul/subliminal.

- Download English subtitles for a video:

```
subliminal download {{[-l|--language]}} {{en}} {{video.ext}}
```

sublist3r

Fast subdomains enumeration tool for penetration testers.

More information: https://github.com/aboul3la/Sublist3r.

- Find subdomains for a domain:

```
sublist3r --domain {{domain_name}}
```

- Find subdomains for a domain, also enabling brute force search:

```
sublist3r --domain {{domain_name}} --bruteforce
```

- Save the found subdomains to a text file:

```
sublist3r --domain {{domain_name}} --output {{path/to/output_file}}
```

- Display help:

```
sublist3r --help
```

sudo

Execute a single command as the superuser or another user.

See also: **`pkexec`**, **`run0`**, **`doas`**.

More information: https://www.sudo.ws/sudo.html.

- Run a command as the superuser:

```
sudo {{less /var/log/syslog}}
```

- Edit a file as the superuser with your default editor:

```
sudo {{[-e|--edit]}} {{/etc/fstab}}
```

- Run a command as another user and/or group:

```
sudo {{[-u|--user]}} {{user}} {{[-g|--group]}} {{group}} {{id -a}}
```

- Repeat the last command prefixed with `sudo` (only in Bash, Zsh, etc.):

```
sudo !!
```

- Launch the default shell with superuser privileges and run login-specific files (`.profile`, `.bash_profile`, etc.):

```
sudo {{[-i|--login]}}
```

- Launch the default shell with superuser privileges without changing the environment:

```
sudo {{[-s|--shell]}}
```

- Launch the default shell as the specified user, loading the user's environment and reading login-specific files (`.profile`, `.bash_profile`, etc.):

```
sudo {{[-i|--login]}} {{[-u|--user]}} {{user}}
```

- List the allowed (and forbidden) commands for the invoking user:

```
sudo {{[-ll|--list --list]}}
```

sudoedit

This command is an alias of **`sudo --edit`**.

- View documentation for the original command:

  ```
  tldr sudo
  ```

sui client faucet

Interact with the Sui faucet.

More information: https://docs.sui.io/references/cli/client#request-a-sui-coin-from-faucet.

- Get a SUI coin from the faucet associated with the active network:

  ```
  sui client faucet
  ```

- Get a SUI coin for the address (accepts also an alias):

  ```
  sui client faucet --address {{address}}
  ```

- Get a SUI coin from custom faucet:

  ```
  sui client faucet --url {{custom-faucet-url}}
  ```

sui client ptb

Create, sign and execute programmable transaction blocks.

More information: https://docs.sui.io/references/cli/ptb.

- Call a Move function from a package and module:

```
sui client ptb --move-call p::m::f "<{{type}}>" args
```

- Make a Move vector with two elements of type u64:

```
sui client ptb --make-move-vec "<u64>" "[1000,2000]"
```

- Split a gas coin and transfer it to address:

```
sui client ptb --split-coins gas "[1000]" --assign new_coins
--transfer-objects "[new_coins]" @{{address}}
```

- Transfer an object to an address:

```
sui client ptb --transfer-objects "[{{object_id}}]"
@{{address}}
```

- Publish a Move package, and transfer the upgrade capability to sender:

```
sui client ptb --move-call sui::tx_context::sender --assign
sender --publish "." --assign upgrade_cap --transfer-objects
"[upgrade_cap]" sender
```

sui client

Publish smart contracts, get object information, execute transactions, and more.

More information: https://docs.sui.io/references/cli/client.

- Create a new address with the ED25519 scheme:

```
sui client new-address ed25519 {{address-alias}}
```

- Create a new testnet environment with an RPC URL and alias:

```
sui client new-env --rpc https://fullnode.testnet.sui.io:443
--alias testnet
```

- Switch to the address of your choice (accepts also an alias):

```
sui client switch --address {{address-alias}}
```

- Switch to the given environment:

```
sui client switch --env {{env-alias}}
```

- Publish a smart contract:

```
sui client publish {{package-path}}
```

- Interact with the Sui faucet:

```
sui client faucet {{subcommand}}
```

- List the gas coins for the given address (accepts also an alias):

```
sui client gas {{address}}
```

- Create, sign, and execute programmable transaction blocks:

```
sui client ptb {{options}} {{subcommand}}
```

sui move

Work with Move source code.

More information: https://docs.sui.io/references/cli/move.

- Create a new Move project in the given folder:

  ```
  sui move new {{project_name}}
  ```

- Test the Move project in the current directory:

  ```
  sui move test
  ```

- Test with coverage and get a summary:

  ```
  sui move test --coverage; sui move coverage summary
  ```

- Find which parts of your code are covered from tests (i.e. explain coverage results):

  ```
  sui move coverage source --module {{module_name}}
  ```

- Build the Move project in the current directory:

  ```
  sui move build
  ```

- Build the Move project from the given path:

  ```
  sui move build --path {{path}}
  ```

- Migrate to Move 2024 for the package at the provided path:

  ```
  sui move migrate {{path}}
  ```

sui

Interact with the Sui network.

More information: https://docs.sui.io/references/cli/cheatsheet.

- Execute a Sui subcommand:

```
sui {{subcommand}}
```

- Build tools for a smart contract:

```
sui move {{subcommand}}
```

- Publish smart contracts, get object information, execute transactions and more:

```
sui client {{subcommand}}
```

- Start a local network:

```
sui start
```

- Update from source:

```
cargo install --locked --git https://github.com/MystenLabs/sui.git --branch testnet sui
```

sum

Compute checksums and the number of blocks for a file.

A predecessor to the more modern **`cksum`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/sum-invocation.html.

- Compute a checksum with BSD-compatible algorithm and 1024-byte blocks:

```
sum {{path/to/file}}
```

- Compute a checksum with System V-compatible algorithm and 512-byte blocks:

```
sum {{[-s|--sysv]}} {{path/to/file}}
```

sunicontopnm

Convert a Sun icon into a Netpbm image.

More information: https://netpbm.sourceforge.net/doc/sunicontopnm.html.

- Convert a Sun icon into a Netpbm image:

```
sunicontopnm {{path/to/input.ico}} > {{path/to/output.pbm}}
```

sup

Manage a RSS feed in the current directory.

See also: **`lb`**.

More information: https://github.com/LukeSmithxyz/lb.

- Add an article to the RSS feed:

```
sup {{path/to/file.html}}
```

supervisorctl

Supervisor, a client/server system that allows its users to control a number of processes on UNIX-like operating systems.

Supervisorctl is the command-line client piece of the supervisor which provides a shell-like interface.

More information: https://supervisord.org/running.html#running-supervisorctl.

- Show the status of a process (or all processes if `process_name` is not specified):

```
supervisorctl status {{process_name}}
```

- Start/stop/restart a process:

```
supervisorctl {{start|stop|restart}} {{process_name}}
```

- Start/stop/restart all processes in a group:

```
supervisorctl {{start|stop|restart}} {{group_name}}:*
```

- Show last 100 bytes of process `stderr`:

```
supervisorctl tail -100 {{process_name}} stderr
```

- Keep displaying `stdout` of a process:

```
supervisorctl tail -f {{process_name}} stdout
```

- Reload process configuration file to add/remove processes as necessary:

```
supervisorctl update
```

supervisord

Supervisor is a client/server system for controlling some processes on UNIX-like operating systems.

Supervisord is the server part of supervisor; it is primarily managed via a configuration file.

More information: https://supervisord.org/running.html#running-supervisord.

- Start `supervisord` with specified configuration file:

```
supervisord {{[-c|--configuration]}} {{path/to/file}}
```

- Run supervisord in the foreground:

```
supervisord {{[-n|--nodaemon]}}
```

surfraw

Query a variety of web search engines.

Consists of a collection of elvi, each of which knows how to search a website.

More information: https://manned.org/surfraw.

- Display the list of supported website search scripts (elvi):

```
surfraw -elvi
```

- Open the elvi's results page for a specific search in the browser:

```
surfraw {{elvi_name}} "{{search_terms}}"
```

- Display an elvi description and its specific options:

```
surfraw {{elvi_name}} {{[-lh|-local-help]}}
```

- Search using an elvi with specific options and open the results page in the browser:

```
surfraw {{elvi_name}} {{elvi_options}} "{{search_terms}}"
```

- Display the URL to the elvi's results page for a specific search:

```
surfraw -print {{elvi_name}} "{{search_terms}}"
```

- Search using the alias:

```
sr {{elvi_name}} "{{search_terms}}"
```

surge

Simple web publishing.

More information: https://surge.sh.

- Upload a new site to surge.sh:

```
surge {{path/to/my_project}}
```

- Deploy site to custom domain (note that the DNS records must point to the surge.sh subdomain):

```
surge {{path/to/my_project}} {{example.com}}
```

- List your surge projects:

```
surge list
```

- Remove a project:

```
surge teardown {{example.com}}
```

suspend

Suspend the execution of the current shell.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-suspend.

- Suspend the current shell (useful for when you are in nested shells like `su`):

```
{{bash}} <Enter> suspend
```

- Continue from suspension if `suspend` was used in a non-nested shell (run this in a separate terminal):

```
pkill -CONT {{bash}}
```

- Force suspension even if this would lock you out of the system:

```
suspend -f
```

sv

Control a running runsv service.

More information: https://manned.org/sv.8.

- Start a service:

```
sudo sv up {{path/to/service}}
```

- Stop a service:

```
sudo sv down {{path/to/service}}
```

- Get service status:

```
sudo sv status {{path/to/service}}
```

- Reload a service:

```
sudo sv reload {{path/to/service}}
```

- Start a service, but only if it's not running and don't restart it if it stops:

```
sudo sv once {{path/to/service}}
```

svgcleaner

SVG optimizing utility.

More information: https://github.com/RazrFalcon/svgcleaner.

- Optimize an SVG:

```
svgcleaner {{input.svg}} {{output.svg}}
```

- Optimize an SVG multiple times:

```
svgcleaner --multipass {{input.svg}} {{output.svg}}
```

svgo

SVG Optimizer: optimizing Scalable Vector Graphics files. Based in Node.js.

It applies a series of transformation rules (plugins), which can be toggled individually.

More information: https://github.com/svg/svgo.

- Optimize a file using the default plugins (overwrites the original file):

```
svgo {{test.svg}}
```

- Optimize a file and save the result to another file:

```
svgo {{test.svg}} {{[-o|--output]}} {{test.min.svg}}
```

- Optimize all SVG files within a directory (overwrites the original files):

```
svgo {{[-f|--folder]}} {{path/to/directory_with_svg_files}}
```

- Optimize all SVG files within a directory and save the resulting files to another directory:

```
svgo {{[-f|--folder]}} {{path/to/input_directory}} {{[-o|--output]}} {{path/to/output_directory}}
```

- Optimize SVG content passed from another command, and save the result to a file:

```
{{cat test.svg}} | svgo {{[-i|--input]}} - {{[-o|--output]}} {{test.min.svg}}
```

- Optimize a file and print out the result:

```
svgo {{test.svg}} {{[-o|--output]}} -
```

- Show available plugins:

```
svgo --show-plugins
```

svgr

Transform SVGs into React components.

More information: https://react-svgr.com/docs/options/.

- Transform a SVG file into a React component to `stdout`:

```
svgr -- {{path/to/file.svg}}
```

- Transform a SVG file into a React component using TypeScript to `stdout`:

```
svgr --typescript -- {{path/to/file.svg}}
```

- Transform a SVG file into a React component using JSX transform to `stdout`:

```
svgr --jsx-runtime automatic -- {{path/to/file.svg}}
```

- Transform all SVG files from a directory to React components into a specific directory:

```
svgr --out-dir {{path/to/output_directory}} {{path/to/input_directory}}
```

- Transform all SVG files from a directory to React components into a specific directory skipping already transformed files:

```
svgr --out-dir {{path/to/output_directory}} --ignore-existing {{path/to/input_directory}}
```

- Transform all SVG files from a directory to React components into a specific directory using a specific case for filenames:

```
svgr --out-dir {{path/to/output_directory}} --filename-case {{camel|kebab|pascal}} {{path/to/input_directory}}
```

- Transform all SVG files from a directory to React components into a specific directory without generating an index file:

```
svgr --out-dir {{path/to/output_directory}} --no-index {{path/to/input_directory}}
```

svgtopam

Convert an SVG file to Netpbm format.

More information: https://netpbm.sourceforge.net/doc/svgtopam.html.

- Convert an SVG file to Netpbm format:

```
svgtopam {{path/to/file.svg}} > {{path/to/output.pam}}
```

- Emit textual output describing the input image:

```
svgtopam {{[-t|-trace]}} {{path/to/file.svg}}
```

svn changelist

Associate a changelist with a set of files.

More information: https://svnbook.red-bean.com/en/1.7/svn-book.html#svn.ref.svn.c.changelist.

- Add files to a changelist, creating the changelist if it does not exist:

```
svn {{[cl|changelist]}} {{changelist_name}} {{path/to/file1 path/to/file2 ...}}
```

- Remove files from a changelist:

```
svn {{[cl|changelist]}} --remove {{path/to/file1 path/to/file2 ...}}
```

- Remove the whole changelist at once:

```
svn {{[cl|changelist]}} --remove {{[-R|--recursive]}} --changelist {{changelist_name}} .
```

- Add the contents of a space-separated list of directories to a changelist:

```
svn {{[cl|changelist]}} {{[-R|--recursive]}} {{changelist_name}} {{path/to/directory1 path/to/directory2 ...}}
```

- Commit a changelist:

```
svn commit --changelist {{changelist_name}}
```

svn

Subversion client tool.

More information: https://svnbook.red-bean.com/en/1.7/svn-book.html#svn.ref.svn.

- Check out a working copy from a repository:

```
svn {{[co|checkout]}} {{file:///path/to/repository}}
```

- Bring changes from the repository into the working copy:

```
svn {{[up|update]}}
```

- Put files and directories under version control, scheduling them for addition to repository. They will be added in next commit:

```
svn add {{PATH}}
```

- Send changes from your working copy to the repository:

```
svn {{[ci|commit]}} {{[-m|--message]}} {{commit_log_message}} [{{PATH}}]
```

- Display changes from the last 10 revisions, showing modified files for each revision:

```
svn log {{[-vl|--verbose --limit]}} {{10}}
```

- Display help:

```
svn {{[h|help]}}
```

swagger-codegen

Generate code and documentation for your REST api from a OpenAPI/swagger definition.

More information: https://github.com/swagger-api/swagger-codegen.

- Generate documentation and code from an OpenAPI/swagger file:

```
swagger-codegen generate {{[-i|--input-spec]}}
{{swagger_file}} {{[-l|--lang]}} {{language}}
```

- Generate Java code using the library retrofit2 and the option useRxJava2:

```
swagger-codegen generate {{[-i|--input-spec]}} {{http://
petstore.swagger.io/v2/swagger.json}} {{[-l|--lang]}}
{{java}} --library {{retrofit2}} -D{{useRxJava2}}={{true}}
```

- List available languages:

```
swagger-codegen langs
```

- Display help for a specific command:

```
swagger-codegen {{generate|config-help|meta|langs|version}}
--help
```

sway

A tiling Wayland compositor.

It uses the same config format as **`i3`**, with some Wayland-specific additions.

More information: https://github.com/swaywm/sway/wiki.

- Start `sway`:

  `sway`

- View documentation for `i3` (`sway` uses the same default keybindings):

  `tldr i3`

swaybg

Wallpaper tool for Wayland compositors.

More information: https://github.com/swaywm/swaybg/blob/master/swaybg.1.scd.

- Set the wallpaper to an image:

```
swaybg {{[-i|--image]}} {{path/to/image}}
```

- Set the wallpaper mode:

```
swaybg {{[-i|--image]}} {{path/to/image}} {{[-m|--mode]}}
{{stretch|fit|fill|center|tile|solid_color}}
```

- Set the wallpaper to a static color:

```
swaybg {{[-c|--color]}} "{{#rrggbb}}"
```

swayidle

Idle management daemon for Wayland.

Note: The configuration options are documented in its man page.

More information: https://github.com/swaywm/swayidle/blob/master/swayidle.1.scd.

- Listen for idle activity using the configuration in `$XDG_CONFIG_HOME/swayidle/config` or `$HOME/swayidle/config`:

  ```
  swayidle
  ```

- Specify an alternative path to the configuration file:

  ```
  swayidle -C {{path/to/file}}
  ```

swaylock

Screen locking utility for Wayland compositors.

More information: https://manned.org/swaylock.

- Lock the screen using the config in `$HOME/.swaylock/config` or `$XDG_CONFIG_HOME/swaylock/config`:

```
swaylock
```

- Lock the screen with a simple color background (`rrggbb` format):

```
swaylock {{[-c|--color]}} {{0000ff}}
```

- Lock the screen with a background image:

```
swaylock {{[-i|--image]}} {{path/to/image}}
```

- Lock the screen and disable the unlock indicator (removes feedback on keypress):

```
swaylock {{[-u|--no-unlock-indicator]}}
```

- Detach from the controlling terminal after locking (like `i3lock`):

```
swaylock {{[-f|--daemonize]}}
```

- Lock the screen with a background image tiled over all monitors:

```
swaylock {{[-i|--image]}} {{path/to/image}} {{[-t|--tiling]}}
```

- Lock the screen and show the number of failed login attempts:

```
swaylock {{[-F|--show-failed-attempts]}}
```

- Load the configuration from a specific file:

```
swaylock {{[-C|--config]}} {{path/to/config}}
```

swaymsg

Send messages to a running instance of Sway using IPC.

See https://github.com/swaywm/sway/blob/master/sway/sway.5.scd for available commands.

More information: https://github.com/swaywm/sway/blob/master/swaymsg/swaymsg.1.scd.

- Run a Sway command:

```
swaymsg {{command}}
```

- Display a list of workspaces:

```
swaymsg {{[-t|--type]}} get_workspaces
```

- Display a list of input devices:

```
swaymsg {{[-t|--type]}} get_inputs
```

- Display a list of output devices:

```
swaymsg {{[-t|--type]}} get_outputs
```

- Display a layout tree of all open windows, containers, outputs, and workspaces:

```
swaymsg {{[-t|--type]}} get_tree
```

SWC

JavaScript and TypeScript compiler written in Rust.

More information: https://swc.rs.

- Transpile a specified input file and output to `stdout`:

```
swc {{path/to/file}}
```

- Transpile the input file every time it is changed:

```
swc {{path/to/file}} --watch
```

- Transpile a specified input file and output to a specific file:

```
swc {{path/to/input_file}} --out-file {{path/to/output_file}}
```

- Transpile a specified input directory and output to a specific directory:

```
swc {{path/to/input_directory}} --out-dir {{path/to/
output_directory}}
```

- Transpile a specified input directory using a specific configuration file:

```
swc {{path/to/input_directory}} --config-file {{path/
to/.swcrc}}
```

- Ignore files in a directory specified using glob path:

```
swc {{path/to/input_directory}} --ignore {{path/to/
ignored_file1 path/to/ignored_file2 ...}}
```

swift

Create, run and build Swift projects.

More information: https://swift.org.

- Start a REPL (interactive shell):

```
swift repl
```

- Execute a program:

```
swift {{file.swift}}
```

- Start a new project with the package manager:

```
swift package init
```

- Generate an Xcode project file:

```
swift package generate-xcodeproj
```

- Update dependencies:

```
swift package update
```

- Compile project for release:

```
swift build -c release
```

swig

Generate bindings between C/C++ code and various high level languages such as JavaScript, Python, C#, and more.

It uses special **`.i`** or **`.swg`** files to generate the bindings (C/C++ with SWIG directives), then outputs a C/C++ file that contains all the wrapper code needed to build an extension module.

More information: https://www.swig.org.

- Generate a binding between C++ and Python:

```
swig -c++ -python -o {{path/to/output_wrapper.cpp}} {{path/to/swig_file.i}}
```

- Generate a binding between C++ and Go:

```
swig -go -cgo -intgosize 64 -c++ {{path/to/swig_file.i}}
```

- Generate a binding between C and Java:

```
swig -java {{path/to/swig_file.i}}
```

- Generate a binding between C and Ruby and prefix the Ruby module with `foo::bar::`:

```
swig -ruby -prefix "{{foo::bar::}}" {{path/to/swig_file.i}}
```

swipl

SWI-Prolog - A comprehensive free Prolog environment.

More information: https://www.swi-prolog.org/pldoc/man?section=cmdline.

- Start an interactive session:

```
swipl
```

- Execute a command without showing any output:

```
swipl --quiet -t "{{command}}"
```

- Execute a script:

```
swipl {{path/to/file.pl}}
```

- Print all shell configuration variables:

```
swipl --dump-runtime-variables
```

- Display version:

```
swipl --version
```

symfony

The console component for the Symfony framework.

More information: https://symfony.com/doc/current/setup.html.

- Create a new Symfony project:

  ```
  symfony new {{name}}
  ```

- Run a local web server:

  ```
  symfony serve
  ```

- Stop the local web server:

  ```
  symfony server:stop
  ```

- Check for security issues in the project's dependencies:

  ```
  symfony security:check
  ```

sync

Flushes all pending write operations to the appropriate disks.

More information: https://www.gnu.org/software/coreutils/manual/html_node/sync-invocation.html.

- Flush all pending write operations on all disks:

```
sync
```

- Flush all pending write operations on a single file to disk:

```
sync {{path/to/file}}
```

- Flush writes and drop file system caches (Linux only):

```
sync; echo 3 | sudo tee /proc/sys/vm/drop_caches
```

- Flush disk writes and attempts to clear inactive memory and filesystem caches (macOS only):

```
sync; sudo purge
```

syncthing

Continuous bidirectional decentralised folder synchronisation tool.

More information: https://docs.syncthing.net/users/syncthing.html.

- Start Syncthing:

```
syncthing
```

- Start Syncthing without opening a web browser:

```
syncthing --no-browser
```

- Change the home directory:

```
syncthing --home {{path/to/directory}}
```

- Run Syncthing with increased logging:

```
syncthing --verbose
```

- Pause all devices:

```
syncthing cli config devices pause --all
```

- Resume all devices:

```
syncthing cli config devices resume --all
```

- Change the address upon which the web interface listens:

```
syncthing --gui-address {{ip_address:port|path/to/
socket.sock}}
```

- Set the log level for output:

```
syncthing --log-level {{info|warning|error|debug}}
```

sysbench

Benchmark a System's CPU, IO and memory.

More information: https://github.com/akopytov/sysbench/.

- Run a CPU benchmark with 1 thread for 10 seconds:

```
sysbench cpu run
```

- Run a CPU benchmark with multiple threads for a specified time:

```
sysbench --threads={{number_of_threads}} --time={{seconds}}
```

- Run a memory benchmark with 1 thread for 10 seconds:

```
sysbench memory run
```

- Prepare a filesystem-level read benchmark:

```
sysbench fileio prepare
```

- Run a filesystem-level benchmark:

```
sysbench --file-test-mode={{rndrd|rndrw|rndwr|seqrd|seqrewr|
seqwr}} fileio run
```

tabula

Extract tables from PDF files.

More information: https://tabula.technology.

- Extract all tables from a PDF to a CSV file:

```
tabula -o {{file.csv}} {{file.pdf}}
```

- Extract all tables from a PDF to a JSON file:

```
tabula --format JSON -o {{file.json}} {{file.pdf}}
```

- Extract tables from pages 1, 2, 3, and 6 of a PDF:

```
tabula --pages {{1-3,6}} {{file.pdf}}
```

- Extract tables from page 1 of a PDF, guessing which portion of the page to examine:

```
tabula --guess --pages {{1}} {{file.pdf}}
```

- Extract all tables from a PDF, using ruling lines to determine cell boundaries:

```
tabula --spreadsheet {{file.pdf}}
```

- Extract all tables from a PDF, using blank space to determine cell boundaries:

```
tabula --no-spreadsheet {{file.pdf}}
```

tac

Display and concatenate files with lines in reversed order.

See also: **`cat`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/tac-invocation.html.

- Concatenate specific files in reversed order:

```
tac {{path/to/file1 path/to/file2 ...}}
```

- Display `stdin` in reversed order:

```
{{cat path/to/file}} | tac
```

- Use a specific separator:

```
tac {{[-s|--separator]}} {{separator}} {{path/to/file1 path/to/file2 ...}}
```

- Use a specific `regex` as a separator:

```
tac {{[-r|--regex]}} {{[-s|--separator]}} {{separator}} {{path/to/file1 path/to/file2 ...}}
```

- Use a separator before each file:

```
tac {{[-b|--before]}} {{path/to/file1 path/to/file2 ...}}
```

tail

Display the last part of a file.

See also: **head**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/tail-invocation.html.

- Show last 10 lines in a file:

```
tail {{path/to/file}}
```

- Show last 10 lines of multiple files:

```
tail {{path/to/file1 path/to/file2 ...}}
```

- Show last `count` lines in file:

```
tail {{[-n|--lines]}} {{count}} {{path/to/file}}
```

- Print a file from a specific line number:

```
tail {{[-n|--lines]}} +{{count}} {{path/to/file}}
```

- Print a specific count of bytes from the end of a given file:

```
tail {{[-c|--bytes]}} {{count}} {{path/to/file}}
```

- Print the last lines of a given file and keep reading it until `<Ctrl c>`:

```
tail {{[-f|--follow]}} {{path/to/file}}
```

- Keep reading file until `<Ctrl c>`, even if the file is inaccessible:

```
tail {{[-F|--retry --follow]}} {{path/to/file}}
```

- Show last `count` lines in a file and refresh every `seconds` seconds:

```
tail {{[-n|--lines]}} {{count}} {{[-s|--sleep-interval]}}
{{seconds}} {{[-f|--follow]}} {{path/to/file}}
```

tailscale file

Send files across connected devices on a Tailscale network.

It currently does not support sending files to devices owned by other users even on the same Tailscale network.

More information: https://tailscale.com/kb/1106/taildrop/.

- Send a file to a specific node:

```
tailscale file cp {{path/to/file}} {{hostname|ip}}:
```

- Store files that were sent to the current node into a specific directory:

```
tailscale file get {{path/to/directory}}
```

tailscale set

Change the specified preferences.

These options can also be changed during the initial setup.

This command can set all options documented in the page for **`tailscale up`**.

More information: https://tailscale.com/kb/1080/cli#set.

- View documentation for `tailscale up`:

```
tldr tailscale up
```

tailscale ssh

SSH to a Tailscale machine (Linux Only).

More information: https://tailscale.com/kb/1193/tailscale-ssh.

- Advertise/Disable SSH on the host:

```
tailscale up --ssh={{true|false}}
```

- SSH to a specific host which has Tailscale-SSH enabled:

```
tailscale ssh {{username}}@{{host}}
```

tailscale up

Connect the client to the Tailscale network.

Note: Run **`sudo tailscale set --operator $USER`** to allow the current user to run these commands.

All options described here can be changed later using **`tailscale set --option argument`**. Use **`--option=false`** to disable options that don't require arguments.

More information: https://tailscale.com/kb/1080/cli/#up.

- Connect to Tailscale:

```
tailscale up
```

- Connect and offer the current machine to be an exit node for internet traffic:

```
tailscale up --advertise-exit-node
```

- Connect using a specific node for internet traffic:

```
tailscale up --exit-node {{exit_node_ip}}
```

- Connect and block incoming connections to the current node:

```
tailscale up --shields-up
```

- Connect and don't accept DNS configuration from the admin panel (defaults to `true`):

```
tailscale up --accept-dns=false
```

- Connect and configure Tailscale as a subnet router:

```
tailscale up --advertise-routes
{{10.0.0.0/24,10.0.1.0/24,...}}
```

- Connect and accept subnet routes from Tailscale:

```
tailscale up --accept-routes
```

- Reset unspecified settings to their default values and connect:

```
tailscale up --reset
```

tailscale

A private WireGuard network service.

Some subcommands such as **up** have their own usage documentation.

More information: https://tailscale.com/kb/1080/cli.

- Allow the current user to operate on the Tailscale daemon:

```
sudo tailscale set --operator $USER
```

- Connect to Tailscale:

```
tailscale up
```

- Disconnect from Tailscale:

```
tailscale down
```

- Display all devices connected to Tailscale (with their IP addresses):

```
tailscale status
```

- Ping a peer node at the Tailscale layer and display which route it took for each response:

```
tailscale ping {{ip|hostname}}
```

- Analyze the local network conditions and display the result:

```
tailscale netcheck
```

- Start a web server for controlling the Tailscale daemon:

```
tailscale web
```

- Display a shareable identifier to help diagnose issues:

```
tailscale bugreport
```

takeout

A Docker-based development-only dependency manager.

More information: https://github.com/tighten/takeout.

- Display a list of available services:

```
takeout enable
```

- Enable a specific service:

```
takeout enable {{name}}
```

- Enable a specific service with the default parameters:

```
takeout enable --default {{name}}
```

- Display a list of enabled services:

```
takeout disable
```

- Disable a specific service:

```
takeout disable {{name}}
```

- Disable all services:

```
takeout disable --all
```

- Start a specific container:

```
takeout start {{container_id}}
```

- Stop a specific container:

```
takeout stop {{container_id}}
```

talosctl

Interact with Talos Linux, a minimal and immutable Kubernetes distribution.

See also: **`kubectl`**.

More information: https://www.talos.dev/latest/reference/cli/.

- Apply a config to a fresh node:

```
talosctl apply-config {{[-i|--insecture]}} {{[-n|--nodes]}}
{{control_plane_ip}} {{[-f|--file]}} {{path/to/
control_plane.yaml}}
```

- Bootstrap the `etcd` cluster on a node:

```
talosctl bootstrap {{[-n|--nodes]}} {{node_ip}}
```

- Edit an API resource:

```
talosctl edit {{resource_to_edit}} {{[-n|--nodes]}}
{{node_ip}}
```

- Get resources:

```
talosctl get {{resource_to_get}} {{[-n|--nodes]}} {{node_ip}}
```

- Download the admin kube configuration from a node:

```
talosctl kubeconfig {{[-n|--nodes]}} {{node_ip}}
```

- Reset a node:

```
talosctl reset {{[-n|--nodes]}} {{node_ip}}
```

tar

Archiving utility.

Often combined with a compression method, such as **gzip** or **bzip2**.

More information: https://www.gnu.org/software/tar/manual/tar.html.

- [c]reate an archive and write it to a [f]ile:

```
tar cf {{path/to/target.tar}} {{path/to/file1 path/to/file2 ...}}
```

- [c]reate a g[z]ipped archive and write it to a [f]ile:

```
tar czf {{path/to/target.tar.gz}} {{path/to/file1 path/to/file2 ...}}
```

- [c]reate a g[z]ipped (compressed) archive from a directory using relative paths:

```
tar czf {{path/to/target.tar.gz}} {{[-C|--directory]}} {{path/to/directory}} .
```

- E[x]tract a (compressed) archive [f]ile into the current directory [v]erbosely:

```
tar xvf {{path/to/source.tar[.gz|.bz2|.xz]}}
```

- E[x]tract a (compressed) archive [f]ile into the target directory:

```
tar xf {{path/to/source.tar[.gz|.bz2|.xz]}} {{[-C|--directory]}} {{path/to/directory}}
```

- [c]reate a compressed archive and write it to a [f]ile, using the file extension to [a]utomatically determine the compression program:

```
tar caf {{path/to/target.tar.xz}} {{path/to/file1 path/to/file2 ...}}
```

- Lis[t] the contents of a tar [f]ile [v]erbosely:

```
tar tvf {{path/to/source.tar}}
```

- E[x]tract files matching a pattern from an archive [f]ile:

```
tar xf {{path/to/source.tar}} --wildcards "{{*.html}}"
```

tarsnap-keygen

Generate a key file for use with Tarsnap, an online backup service.

More information: https://www.tarsnap.com/man-tarsnap-keygen.1.html.

- Register a machine with the Tarsnap server:

```
sudo tarsnap-keygen --keyfile {{path/to/file.key}} --user
{{user_email}} --machine {{machine_name}}
```

- Encrypt the key file (a passphrase will be requested twice):

```
sudo tarsnap-keygen --keyfile {{path/to/file.key}} --user
{{user_email}} --machine {{machine_name}} --passphrased
```

tarsnap

Manipulate remote Tarsnap encrypted backups.

Note: You don't need to specify the key file and the cache directory if you configure them in **`/usr/local/etc/tarsnap.conf`** or **`~/.tarsnaprc`**.

See also: **`tarsnap-keygen`**.

More information: https://www.tarsnap.com/man-tarsnap.1.html.

- [c]reate a backup archive of one or more files or directories, specifying the cryptographic key and the cache directory:

```
tarsnap -c --keyfile {{path/to/key_file}} --cachedir {{path/to/cache_directory}} -f {{archive_name}} {{path/to/file_or_directory1 path/to/file_or_directory2 ...}}
```

- Display how much data would be uploaded:

```
tarsnap -c --dry-run --print-stats --keyfile {{path/to/key_file}} --cachedir {{path/to/cache_directory}} -f {{archive_name}} {{path/to/file_or_directory1 path/to/file_or_directory2 ...}}
```

- List stored archives:

```
tarsnap --list-archives --keyfile {{path/to/key_file}}
```

- [d]elete a specific archive:

```
tarsnap -d --keyfile {{path/to/key_file}} --cachedir {{path/to/cache_directory}} -f {{archive_name}}
```

- Lis[t] the contents of a specific archive in [v]erbose mode:

```
tarsnap -tv --keyfile {{path/to/key_file}} -f {{archive_name}}
```

- Restore one or more files or directories from a specific archive:

```
tarsnap -x --keyfile {{path/to/key_file}} -f {{archive_name}} {{path/to/file_or_directory1 path/to/file_or_directory2 ...}}
```

- Copy an archive:

```
tarsnap -c --keyfile {{path/to/key_file}} -f {{new_archive_name}} @@{{source_archive_name}}
```

task

To-do list manager.

More information: https://taskwarrior.org/docs/.

- Add a new task which is due tomorrow:

```
task add {{description}} due:{{tomorrow}}
```

- Update a task's priority:

```
task {{task_id}} modify priority:{{H|M|L}}
```

- Complete a task:

```
task {{task_id}} done
```

- Delete a task:

```
task {{task_id}} delete
```

- List all open tasks:

```
task list
```

- List open tasks due before the end of the week:

```
task list due.before:{{eow}}
```

- Show a graphical burndown chart, by day:

```
task burndown.daily
```

- List all reports:

```
task reports
```

tb

Manage tasks and notes across multiple boards.

More information: https://github.com/klaussinani/taskbook.

- Add a new task to a board:

  ```
  tb --task {{task_description}} @{{board_name}}
  ```

- Add a new note to a board:

  ```
  tb --note {{note_description}} @{{board_name}}
  ```

- Edit item's priority:

  ```
  tb --priority @{{item_id}} {{priority}}
  ```

- Check/uncheck item:

  ```
  tb --check {{item_id}}
  ```

- Archive all checked items:

  ```
  tb --clear
  ```

- Move item to a board:

  ```
  tb --move @{{item_id}} {{board_name}}
  ```

tbl

Table preprocessor for the groff (GNU Troff) document formatting system.

See also: **`groff`**, **`troff`**.

More information: https://manned.org/tbl.

- Process input with tables, saving the output for future typesetting with groff to PostScript:

```
tbl {{path/to/input_file}} > {{path/to/output.roff}}
```

- Typeset input with tables to PDF using the [me] macro package:

```
tbl -T {{pdf}} {{path/to/input.tbl}} | groff -{{me}} -T
{{pdf}} > {{path/to/output.pdf}}
```

tcc

A tiny C compiler that can run C source files as scripts and otherwise has command-line options similar to **`gcc`**.

More information: https://bellard.org/tcc/tcc-doc.html.

- Compile and link 2 source files to generate an executable:

```
tcc -o {{executable_name}} {{path/to/file1.c}} {{path/to/file2.c}}
```

- Directly run an input file like a script and pass arguments to it:

```
tcc -run {{path/to/source_file.c}} {{arguments}}
```

- Interpret C source files with a shebang inside the file:

```
#!/{{path/to/tcc}} -run
```

tcpdump

Dump traffic on a network.

More information: https://www.tcpdump.org.

- List available network interfaces:

```
tcpdump {{[-D|--list-interfaces]}}
```

- Capture the traffic of a specific interface:

```
sudo tcpdump {{[-i|--interface]}} {{eth0}}
```

- Capture all TCP traffic showing contents ([A]SCII) in console:

```
sudo tcpdump -A tcp
```

- Capture the traffic from or to a host:

```
sudo tcpdump host {{www.example.com}}
```

- Capture the traffic from a specific interface, source, destination and destination port:

```
sudo tcpdump {{[-i|--interface]}} {{eth0}} src {{192.168.1.1}} and dst {{192.168.1.2}} and dst port {{80}}
```

- Capture the traffic of a network:

```
sudo tcpdump net {{192.168.1.0/24}}
```

- Capture all traffic except traffic over port 22 and [w]rite to a dump file:

```
sudo tcpdump -w {{dumpfile.pcap}} port not {{22}}
```

- [r]ead from a given dump file:

```
tcpdump -r {{dumpfile.pcap}}
```

tcpreplay

Replay network traffic stored in a **pcap** file.

More information: https://tcpreplay.appneta.com/wiki/tcpreplay-man.html.

- List available network interfaces:

```
tcpreplay --listnics
```

- Replay traffic to interface:

```
tcpreplay {{[-i|--intf1]}} {{eth0}} {{traffic.pcap}}
```

- Replay traffic to interface and `stdout`:

```
tcpreplay {{[-i|--intf1]}} {{eth0}} {{[-v|--verbose]}}
{{traffic.pcap}}
```

- Replay traffic to interface as fast as possible:

```
tcpreplay {{[-i|--intf1]}} {{eth0}} {{[-t|--topspeed]}}
{{traffic.pcap}}
```

- Replay traffic to interface at given Mbps:

```
tcpreplay {{[-i|--intf1]}} {{eth0}} {{[-M|--mbps]}} {{10}}
{{traffic.pcap}}
```

- Replay traffic to interface several times:

```
tcpreplay {{[-i|--intf1]}} {{eth0}} {{[-l|--loop]}}
{{num_times}} {{traffic.pcap}}
```

tcsh

C shell with file name completion and command-line editing.

See also: **`csh`**.

More information: https://manned.org/tcsh.

- Start an interactive shell session:

```
tcsh
```

- Start an interactive shell session without loading startup configs:

```
tcsh -f
```

- Execute specific [c]ommands:

```
tcsh -c "{{echo 'tcsh is executed'}}"
```

- Execute a specific script:

```
tcsh {{path/to/script.tcsh}}
```

- Check a specific script for syntax errors:

```
tcsh -n {{path/to/script.tcsh}}
```

- Execute specific commands from `stdin`:

```
{{echo "echo 'tcsh is executed'"}} | tcsh
```

tea

Interact with Gitea servers.

More information: https://gitea.com/gitea/tea.

- Log into a Gitea server:

```
tea login add --name "{{name}}" --url "{{url}}" --token "{{token}}"
```

- Display all repositories:

```
tea repos ls
```

- Display a list of issues:

```
tea issues ls
```

- Display a list of issues for a specific repository:

```
tea issues ls --repo "{{repository}}"
```

- Create a new issue:

```
tea issues create --title "{{title}}" --body "{{body}}"
```

- Display a list of open pull requests:

```
tea pulls ls
```

- Open the current repository in a browser:

```
tea open
```

tectonic

A modern, self-contained TeX/LaTeX engine.

More information: https://tectonic-typesetting.github.io/book/latest.

- Compile a standalone TeX/LaTeX file:

```
tectonic -X compile {{path/to/file.tex}}
```

- Compile a standalone TeX/LaTeX file with synctex data:

```
tectonic -X compile --synctex {{path/to/file.tex}}
```

- Initialize a tectonic project in the current directory:

```
tectonic -X init
```

- Initialize a tectonic project in the specified directory:

```
tectonic -X new {{project_name}}
```

- Build the project in the current directory:

```
tectonic -X build
```

- Start a watcher to build the project in the current directory on change:

```
tectonic -X watch
```

tee

Read from **`stdin`** and write to **`stdout`** and files (or commands).

More information: https://www.gnu.org/software/coreutils/manual/html_node/tee-invocation.html.

- Copy `stdin` to each file, and also to `stdout`:

```
echo "example" | tee {{path/to/file}}
```

- Append to the given files, do not overwrite:

```
echo "example" | tee {{[-a|--append]}} {{path/to/file}}
```

- Print `stdin` to the terminal, and also pipe it into another program for further processing:

```
echo "example" | tee {{/dev/tty}} | {{xargs printf "[%s]"}}
```

- Create a directory called "example", count the number of characters in "example" and write "example" to the terminal:

```
echo "example" | tee >(xargs mkdir) >(wc {{[-c|--bytes]}})
```

telegram-desktop

Instant messenger with open source clients, chats and stickers.

More information: https://manned.org/telegram-desktop.

- Start GUI:

```
telegram-desktop
```

- Run GUI as an autostart if allowed:

```
telegram-desktop -autostart
```

- Run GUI minimized to tray:

```
telegram-desktop -startintray
```

telnet

Connect to a specified port of a host using the telnet protocol.

More information: https://manned.org/telnet.

- Telnet to the default port of a host:

```
telnet {{host}}
```

- Telnet to a specific port of a host:

```
telnet {{ip_address}} {{port}}
```

- Exit a telnet session:

```
quit
```

- Emit the default escape character combination for terminating the session:

```
<Ctrl ]>
```

- Start `telnet` with "x" as the session termination character:

```
telnet {{[-e|--escape]}} {{x}} {{ip_address}} {{port}}
```

- Telnet to Star Wars animation:

```
telnet {{towel.blinkenlights.nl}}
```

termdown

Countdown timer and stopwatch.

More information: https://github.com/trehn/termdown.

- Start a stopwatch:

```
termdown
```

- Start a 1 minute and 30 seconds countdown:

```
termdown {{1m30s}}
```

- Start a 1 minute 30 seconds countdown with blinking the terminal at the end:

```
termdown {{1m30s}} --blink
```

- Show a title above countdown:

```
termdown {{1m30s}} --title "{{Interesting title}}"
```

- Display current time:

```
termdown --time
```

terminalizer

Record the terminal, generate animated GIFs and a web player, and optionally upload to https://terminalizer.com.

See also: **asciinema**.

More information: https://www.terminalizer.com/docs#getting-started.

- Create the global configuration directory:

```
terminalizer init
```

- Record the terminal and create a recording file:

```
terminalizer record {{path/to/recording.gif}}
```

- Play a recorded file on the terminal:

```
terminalizer play {{path/to/recording.gif}}
```

- Render a recording file as an animated GIF image:

```
terminalizer render {{path/to/recording.gif}}
```

- Upload a video to https://terminalizer.com:

```
terminalizer share {{path/to/recording.gif}}
```

terraform fmt

Format configuration according to Terraform language style conventions.

More information: https://developer.hashicorp.com/terraform/cli/commands/fmt.

- Format the configuration in the current directory:

```
terraform fmt
```

- Format the configuration in the current directory and subdirectories:

```
terraform fmt -recursive
```

- Display diffs of formatting changes:

```
terraform fmt -diff
```

- Do not list files that were formatted to `stdout`:

```
terraform fmt -list=false
```

terraform output

Export structured data about your Terraform resources.

More information: https://developer.hashicorp.com/terraform/cli/commands/output.

- With no additional arguments, `output` will display all outputs for the root module:

  ```
  terraform output
  ```

- Output only a value with specific name:

  ```
  terraform output {{name}}
  ```

- Convert the output value to a raw string (useful for shell scripts):

  ```
  terraform output -raw
  ```

- Format the outputs as a JSON object, with a key per output (useful with jq):

  ```
  terraform output -json
  ```

terraform plan

Generate and show Terraform execution plans.

More information: https://developer.hashicorp.com/terraform/cli/commands/plan.

- Generate and show the execution plan in the currently directory:

```
terraform plan
```

- Show a plan to destroy all remote objects that currently exist:

```
terraform plan -destroy
```

- Show a plan to update the Terraform state and output values:

```
terraform plan -refresh-only
```

- Specify values for input variables:

```
terraform plan -var '{{name1}}={{value1}}' -var '{{name2}}
={{value2}}'
```

- Focus Terraform's attention on only a subset of resources:

```
terraform plan -target {{resource_type.resource_name[instance
index]}}
```

- Output a plan as JSON:

```
terraform plan -json
```

- Write a plan to a specific file:

```
terraform plan -no-color > {{path/to/file}}
```

terraform

Create and deploy infrastructure as code to cloud providers.

More information: https://developer.hashicorp.com/terraform/cli/commands.

- Initialize a new or existing Terraform configuration:

  ```
  terraform init
  ```

- Verify that the configuration files are syntactically valid:

  ```
  terraform validate
  ```

- Format configuration according to Terraform language style conventions:

  ```
  terraform fmt
  ```

- Generate and show an execution plan:

  ```
  terraform plan
  ```

- Build or change infrastructure:

  ```
  terraform apply
  ```

- Destroy Terraform-managed infrastructure:

  ```
  terraform destroy
  ```

terragrunt

Keep your Terraform CLI arguments DRY.

More information: https://terragrunt.gruntwork.io.

- Generate and show an execution plan:

  ```
  terragrunt plan
  ```

- Build or change infrastructure:

  ```
  terragrunt apply
  ```

- Show current deployment (from state):

  ```
  terragrunt show
  ```

- Show module output values:

  ```
  terragrunt output
  ```

- Destroy Terraform-managed infrastructure:

  ```
  terragrunt destroy
  ```

- Build or change infrastructure from a tree of Terragrunt modules (stack):

  ```
  terragrunt run-all apply
  ```

tesseract

OCR (Optical Character Recognition) engine.

More information: https://github.com/tesseract-ocr/tesseract/blob/main/doc/tesseract.1.asc.

- Recognize text in an image and save it to the given path (a `.txt` extension is added automatically):

```
tesseract {{path/to/image.png}} {{path/to/output_file}}
```

- Specify a custom [l]anguage (default is English) with an ISO 639-2 code (e.g. deu = Deutsch = German):

```
tesseract -l deu {{path/to/image.png}} {{path/to/output}}
```

- List the ISO 639-2 codes of installed languages:

```
tesseract --list-langs
```

- Specify a custom [p]age [s]egmentation [m]ode (default is 3):

```
tesseract --psm {{0..13}} {{path/to/image.png}} {{path/to/output}}
```

- List page segmentation modes and their descriptions:

```
tesseract --help-psm
```

test

Check file types and compare values.

Returns 0 if the condition evaluates to true, 1 if it evaluates to false.

More information: https://www.gnu.org/software/coreutils/manual/html_node/test-invocation.html.

- Test if a given variable is equal to a given string:

```
test "{{$MY_VAR}}" = "{{/bin/zsh}}"
```

- Test if a given variable is empty ([z]ero length):

```
test -z "{{$GIT_BRANCH}}"
```

- Test if a [f]ile exists:

```
test -f "{{path/to/file_or_directory}}"
```

- Test if a [d]irectory does not exist:

```
test ! -d "{{path/to/directory}}"
```

- If A is true, then do B, or C in the case of an error (notice that C may run even if A fails):

```
test {{condition}} && {{echo "true"}} || {{echo "false"}}
```

- Use `test` in a conditional statement:

```
if test -f "{{path/to/file}}"; then echo "File exists"; else echo "File does not exist"; fi
```

testssl

Check SSL/TLS protocols and ciphers supported by a server.

More information: https://testssl.sh/doc/testssl.1.html.

- Test a server (run every check) on port 443:

```
testssl {{example.com}}
```

- Test a different port:

```
testssl {{example.com:465}}
```

- Only check available protocols:

```
testssl --protocols {{example.com}}
```

- Only check vulnerabilities:

```
testssl --vulnerable {{example.com}}
```

- Only check HTTP security headers:

```
testssl --headers {{example.com}}
```

- Test other STARTTLS enabled protocols:

```
testssl --starttls {{ftp|smtp|pop3|imap|xmpp|sieve|xmpp-server|telnet|ldap|irc|lmtp|nntp|postgres|mysql}} {{example.com}}:{{port}}
```

tex-fmt

Format LaTeX source code.

More information: https://github.com/WGUNDERWOOD/tex-fmt.

- Format a file, overwriting the original:

```
tex-fmt {{path/to/file.tex}}
```

- Check if a file is correctly formatted:

```
tex-fmt --check {{path/to/file.tex}}
```

- Format a file read from `stdin` and print to `stdout`:

```
cat {{path/to/file.tex}} | tex-fmt --stdin
```

tex

Compile a DVI document from TeX source files.

More information: https://www.tug.org/begin.html.

- Compile a DVI document:

```
tex {{source.tex}}
```

- Compile a DVI document, specifying an output directory:

```
tex -output-directory={{path/to/directory}} {{source.tex}}
```

- Compile a DVI document, exiting on each error:

```
tex -halt-on-error {{source.tex}}
```

texcount

Count words in TeX documents omitting macros.

Note: If the TeX document uses **`\include`** or **`\input`** and you want to count the included files, **`texcount`** must be run in the directory of the root TeX file.

More information: https://app.uio.no/ifi/texcount/howto.html.

- Count words in a TeX file:

```
texcount {{path/to/file.tex}}
```

- Count words in a document and subdocuments built with `\input` or `\include`:

```
texcount -merge {{file.tex}}
```

- Count words in a document and subdocuments, listing each file separately (and a total count):

```
texcount -inc {{file.tex}}
```

- Count words in a document and subdocuments, producing subcounts by chapter (instead of subsection):

```
texcount -merge -sub=chapter {{file.tex}}
```

- Count words with verbose output:

```
texcount -v {{path/to/file.tex}}
```

texdoc

Search for appropriate documentation for (La)TeX commands or packages.

More information: https://manned.org/texdoc.

- Open the first search result in the default PDF viewer:

```
texdoc {{search}}
```

- List the best search results:

```
texdoc {{[-l|--list]}} {{search}}
```

- Open full documentation of texdoc:

```
texdoc {{texdoc}}
```

texliveonfly

Downloads missing TeX Live packages while compiling **`.tex`** files.

More information: https://ctan.org/tex-archive/support/texliveonfly.

- Download missing packages while compiling:

```
texliveonfly {{source.tex}}
```

- Use a specific compiler (defaults to `pdflatex`):

```
texliveonfly {{[-c|--compiler]}} {{compiler}} {{source.tex}}
```

- Use a custom TeX Live `bin` folder:

```
texliveonfly --texlive_bin={{path/to/texlive_bin}}
{{source.tex}}
```

textql

Execute SQL against structured text like CSV or TSV files.

More information: https://github.com/dinedal/textql.

- Print the lines in the specified CSV file that match an SQL query to `stdout`:

```
textql -sql "{{SELECT * FROM filename}}" {{path/to/
filename.csv}}
```

- Query a TSV file:

```
textql -dlm=tab -sql "{{SELECT * FROM filename}}" {{path/to/
filename.tsv}}
```

- Query file with header row:

```
textql -dlm={{delimiter}} -header -sql "{{SELECT * FROM
filename}}" {{path/to/filename.csv}}
```

- Read data from `stdin`:

```
cat {{path/to/file}} | textql -sql "{{SELECT * FROM stdin}}"
```

- Join two files on a specified common column:

```
textql -header -sql "SELECT * FROM {{path/to/file1}} JOIN
{{file2}} ON {{path/to/file1}}.{{c1}} = {{file2}}.{{c1}}
LIMIT {{10}}" -output-header {{path/to/file1.csv}} {{path/to/
file2.csv}}
```

- Format output using an output delimiter with an output header line:

```
textql -output-dlm={{delimiter}} -output-header -sql "SELECT
{{column}} AS {{alias}} FROM {{filename}}" {{path/to/
filename.csv}}
```

tgatoppm

Convert a TrueVision Targa file to a Netpbm image.

More information: https://netpbm.sourceforge.net/doc/tgatoppm.html.

- Convert a TrueVision Targa file to a PPM image:

```
tgatoppm {{path/to/file.tga}} > {{path/to/output.ppm}}
```

- Dump information from the TGA header to `stdout`:

```
tgatoppm {{[-h|-headerdump]}} {{path/to/file.tga}} > {{path/
to/output.ppm}}
```

- Write the transparency channel values of the input image to the specified file:

```
tgatoppm {{[-a|-alphaout]}} {{path/to/transparency_file.pgm}}
{{path/to/file.tga}} > {{path/to/output.ppm}}
```

- Display version:

```
tgatoppm {{[-v|-version]}}
```

tgcloud

Manage a Telegram account.

See also: **`tgsend`**, **`tginfo`**, **`tgutil`**.

More information: https://pypi.org/project/telegram-cloud/.

- Upload a file to a chat with a caption:

```
tgcloud {{[-m|--mode]}} upload {{[-n|--name]}}
{{session_name}} {{[-u|--username]}} {{chat_id}} {{[-p|--
path]}} {{path_of_the_file}} {{[-c|--caption]}} {{caption}}
```

- Download a file from a chat and store it in `path/to/store` with a caption:

```
tgcloud {{[-m|--mode]}} download {{[-n|--name]}}
{{session_name}} {{[-u|--username]}} {{chat_id}} {{[-p|--
path]}} {{path/to/store}} {{[-c|--caption]}} {{caption}}
```

tginfo

Manage a Telegram account.

See also: **`tgcloud`**, **`tgsend`**, **`tgutil`**.

More information: https://pypi.org/project/telegram-cloud/.

- Search for all files inside a Telegram conversation based on their filename or caption:

```
tginfo {{[-n|--name]}} {{session_name}} {{[-u|--username]}} {{chat_id}} {{[-s|--search]}} "{{query}}"
```

- Get info for all media inside a Telegram chat:

```
tginfo {{[-n|--name]}} {{session_name}} {{[-u|--username]}} {{chat_id}}
```

tgpt

Talk to an AI chatbot without the need for API keys.

Available providers: **`openai`**, **`opengpts`**, **`koboldai`**, **`phind`**, **`llama2`**, **`blackboxai`**.

More information: https://github.com/aandrew-me/tgpt.

- Chat with the default provider (GPT-3.5-turbo):

```
tgpt "{{prompt}}"
```

- Start multi-line interactive mode:

```
tgpt {{[-m|--multiline]}}
```

- Generate images and save them to the current directory:

```
tgpt {{[-img|--image]}} "{{prompt}}"
```

- Generate code with the default provider (GPT-3.5-turbo):

```
tgpt {{[-c|--code]}} "{{prompt}}"
```

- Chat with a specific provider quietly (without animations):

```
tgpt --provider {{openai|opengpts|koboldai|phind|llama2|
blackboxai}} {{[-q|--quiet]}} {{[-w|--whole]}} "{{prompt}}"
```

- Generate and execute shell commands using a specific provider (with a confirmation prompt):

```
tgpt --provider {{llama2}} {{[-s|--shell]}} "{{prompt}}"
```

- Prompt with an API key, model, max response length, temperature, and `top_p` (required when using `openai` provider):

```
tgpt --provider openai --key "{{api_key}}" --model
"{{gpt-3.5-turbo}}" --max-length {{10}} --temperature {{0.7}}
--top_p {{0.9}} "{{prompt}}"
```

- Feed a file as additional pre-prompt input:

```
tgpt < {{path/to/file}} --provider {{blackboxai}}
"{{prompt}}"
```

tgsend

Manage a Telegram account.

See also: **`tgcloud`**, **`tginfo`**, **`tgutil`**.

More information: https://pypi.org/project/telegram-cloud/.

- Send a message to a chat:

```
tgsend {{[-n|--name]}} {{session_name}} {{[-u|--username]}}
{{chat_id}} "{{message}}"
```

- Send a message to chat with silent notification:

```
tgsend {{[-n|--name]}} {{session_name}} {{[-u|--username]}}
{{chat_id}} --silent "{{message}}"
```

tgutil

Manage a Telegram account.

See also: **`tgcloud`**, **`tgsend`**, **`tginfo`**.

More information: https://pypi.org/project/telegram-cloud/.

- Edit the last text message on a chat with new text:

```
tgutil {{[-n|--name]}} {{session_name}} {{[-u|--username]}}
{{chat_id}} {{[-m|--mode]}} edit --text "{{current_text}}" --
newtext "{{new_text}}"
```

- Edit all text messages that contain `current_text` with `new_text`:

```
tgutil {{[-n|--name]}} {{session_name}} {{[-u|--username]}}
{{chat_id}} {{[-m|--mode]}} editall --text "{{current_text}}"
--newtext "{{new_text}}"
```

- Delete the last text message on a chat with search value:

```
tgutil {{[-n|--name]}} {{session_name}} {{[-u|--username]}}
{{chat_id}} {{[-m|--mode]}} delete --text "{{current_text}}"
```

- Delete all messages containing the `query` value:

```
tgutil {{[-n|--name]}} {{session_name}} {{[-u|--username]}}
{{chat_id}} {{[-m|--mode]}} deleteall --text "{{query}}"
```

theHarvester

A tool designed to be used in the early stages of a penetration test.

More information: https://github.com/laramies/theHarvester.

- Gather information on a domain using Google:

```
theHarvester --domain {{domain_name}} --source google
```

- Gather information on a domain using multiple sources:

```
theHarvester --domain {{domain_name}} --source
{{duckduckgo,bing,crtsh}}
```

- Change the limit of results to work with:

```
theHarvester --domain {{domain_name}} --source {{google}} --
limit {{200}}
```

- Save the output to two files in XML and HTML format:

```
theHarvester --domain {{domain_name}} --source {{google}} --
file {{output_file_name}}
```

- Display help:

```
theHarvester --help
```

then

This shell keyword is used in combination with **`if`** and **`elif`** to separate branches and branch conditions.

- View documentation for `if` command:

```
tldr if
```

thinkjettopbm

Convert a HP ThinkJet printer commands file to a PBM file.

More information: https://netpbm.sourceforge.net/doc/thinkjettopbm.html.

- Convert a HP ThinkJet printer commands file to a PBM file:

```
thinkjettopbm {{path/to/input}} > {{path/to/output.pbm}}
```

- Print debug information to `stderr`:

```
thinkjettopbm {{[-d|-debug]}} {{path/to/input}} > {{path/to/
output.pbm}}
```

thunderbird

Email client and RSS reader.

More information: https://manned.org/thunderbird.

- Open thunderbird:

```
thunderbird
```

- Use a specific user profile:

```
thunderbird -P {{profile_name}}
```

- Use a specific user profile directory:

```
thunderbird --profile {{path/to/profile_directory}}
```

tidy

Clean up and pretty print HTML, XHTML and XML files.

Note: **`tidy`** cannot preserve original indentation.

More information: https://api.html-tidy.org/tidy/tidylib_api_next/group__options__cli.html#gad7a9fcaf7b2a712a82e625e84c042b28.

- Pretty print an HTML file:

```
tidy {{path/to/file.html}}
```

- Enable indentation, wrapping lines in 100, saving to `output.html`:

```
tidy {{[-i|--indent]}} y {{[-w|--wrap]}} 100 {{[-o|-output]}}
{{path/to/output.html}} {{path/to/file.html}}
```

- Modify an HTML file in-place using a configuration file:

```
tidy -config {{path/to/configuration}} {{[-m|-modify]}}
{{path/to/file.html}}
```

tifftopnm

Convert a TIFF image to a PNM image.

More information: https://netpbm.sourceforge.net/doc/tifftopnm.html.

- Convert a TIFF to a PNM file:

```
tifftopnm {{path/to/input_file.tiff}} > {{path/to/
output_file.pnm}}
```

- Create a PGM file containing the alpha channel of the input image:

```
tifftopnm {{[-a|-alphaout]}} {{path/to/alpha_file.pgm}}
{{path/to/input_file.tiff}} > {{path/to/output_file.pnm}}
```

- Respect the `fillorder` tag in the input TIFF image:

```
tifftopnm {{[-r|-respectfillorder]}} {{path/to/
input_file.tiff}} > {{path/to/output_file.pnm}}
```

- Print TIFF header information to `stderr`:

```
tifftopnm {{[-h|-headerdump]}} {{path/to/input_file.tiff}} >
{{path/to/output_file.pnm}}
```

tig

A configurable **ncurses**-based TUI for Git.

See also: **gitui**, **git-gui**.

More information: https://jonas.github.io/tig/doc/manual.html.

- Show the sequence of commits starting from the current one in reverse chronological order:

```
tig
```

- Show the history of a specific branch:

```
tig {{branch}}
```

- Show the history of specific files or directories:

```
tig {{path1 path2 ...}}
```

- Show the difference between two references (such as branches or tags):

```
tig {{base_ref}}..{{compared_ref}}
```

- Browse git blame interactively (press `<,>` to jump to parent):

```
tig blame {{path/to/file}}
```

- Display commits from all branches and stashes:

```
tig --all
```

- Start in stash view, displaying all saved stashes:

```
tig stash
```

- Display help in TUI:

```
<h>
```

time

Measure how long a command took to run.

Note: **`time`** can either exist as a shell builtin, a standalone program or both.

See also: **`times`**.

More information: https://manned.org/time.

- Run the `command` and print the time measurements to `stdout`:

```
time {{command}}
```

- Create a very simple stopwatch (only works in Bash):

```
time read
```

timeout

Run a command with a time limit.

More information: https://www.gnu.org/software/coreutils/manual/html_node/timeout-invocation.html.

- Run `sleep 10` and terminate it after 3 seconds:

```
timeout 3s sleep 10
```

- Send a signal to the command after the time limit expires (`TERM` by default, `kill -l` to list all signals):

```
timeout {{[-s|--signal]}} {{INT|HUP|KILL|...}} {{5s}} {{sleep 10}}
```

- Send verbose output to `stderr` showing signal sent upon timeout:

```
timeout {{[-v|--verbose]}} {{0.5s|1m|1h|1d|...}} {{command}}
```

- Preserve the exit status of the command regardless of timing out:

```
timeout {{[-p|--preserve-status]}} {{1s|1m|1h|1d|...}} {{command}}
```

- Send a forceful `KILL` signal after certain duration if the command ignores initial signal upon timeout:

```
timeout {{[-k|--kill-after]}} {{5m}} {{30s}} {{command}}
```

times

Print the cumulative CPU usage time of the current shell.

See also: **`time`**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-times.

- Print CPU usage. First line is current shell CPU usage for User and System. Second is all child processes:

```
times
```

timetrap

Simple time tracker written in Ruby.

More information: https://github.com/samg/timetrap.

- Create a new timesheet:

```
timetrap sheet {{timesheet}}
```

- Check in an entry started 5 minutes ago:

```
timetrap in --at "{{5 minutes ago}}" {{entry_notes}}
```

- Display the current timesheet:

```
timetrap display
```

- Edit the last entry's end time:

```
timetrap edit --end {{time}}
```

timew

A time tracking tool used to measure the duration of activities.

More information: https://timewarrior.net/docs.

- Start tracking an activity:

```
timew start
```

- Tag the current activity:

```
timew tag {{activity_tag}}
```

- Start tracking and tag a new activity:

```
timew start {{activity_tag}}
```

- Stop the current activity:

```
timew stop
```

- Track an activity in the past:

```
timew track {{start_time}} - {{end_time}} {{activity_tag}}
```

- View tracked items of the day:

```
timew summary
```

- View report for the last day, week, current month, etc.:

```
timew summary :{{today|yesterday|week|lastweek|month|
lastmonth|year|lastyear}}
```

timidity

Play and convert MIDI files.

More information: https://timidity.sourceforge.net.

- Play a MIDI file:

```
timidity {{path/to/file.mid}}
```

- Play a MIDI file in a loop:

```
timidity --loop {{path/to/file.mid}}
```

- Play a MIDI file in a specific key (0 = C major/A minor, -1 = F major/D minor, +1 = G major/E minor, etc.):

```
timidity --force-keysig={{-flats|+sharps}} {{path/to/file.mid}}
```

- Convert a MIDI file to PCM (WAV) audio:

```
timidity --output-mode={{w}} --output-file={{path/to/file.wav}} {{path/to/file.mid}}
```

- Convert a MIDI file to FLAC audio:

```
timidity --output-mode={{F}} --output-file={{path/to/file.flac}} {{path/to/file.mid}}
```

tldr-lint

Lint and format **`tldr`** pages.

Note: **`tldrl`** can be used as an alias for **`tldr-lint`**.

More information: https://github.com/tldr-pages/tldr-lint#usage.

- Lint a single page or all pages in a directory:

```
tldr-lint {{path/to/page_or_directory}}
```

- Ignore specific `tldr-lint` error codes while linting:

```
tldr-lint {{[-I|--ignore]}} {{TLDR001,TLDR002,...}}
```

- Format a specific page to `stdout`:

```
tldr-lint {{[-f|--format]}} {{path/to/page.md}}
```

- Format a page in place:

```
tldr-lint {{[-f|--format]}} {{[-i|--in-place]}} {{path/to/page.md}}
```

tldr

Display simple help pages for command-line tools from the tldr-pages project.

Note: The **`--language`** and **`--list`** options are not required by the client specification, but most clients implement them.

More information: https://github.com/tldr-pages/tldr/blob/main/CLIENT-SPECIFICATION.md#command-line-interface.

- Print the tldr page for a specific command (hint: this is how you got here!):

```
tldr {{command}}
```

- Print the tldr page for a specific subcommand:

```
tldr {{command}} {{subcommand}}
```

- Print the tldr page for a command in the given language (if available, otherwise fall back to English):

```
tldr {{[-L|--language]}} {{language_code}} {{command}}
```

- Print the tldr page for a command from a specific platform:

```
tldr {{[-p|--platform]}} {{android|cisco-ios|common|dos|
freebsd|linux|netbsd|openbsd|osx|sunos|windows}} {{command}}
```

- Update the local cache of tldr pages:

```
tldr {{[-u|--update]}}
```

- List all pages for the current platform and `common`:

```
tldr {{[-l|--list]}}
```

- List all available subcommand pages for a command:

```
tldr {{[-l|--list]}} | grep {{command}} | column
```

- Print the tldr page for a random command:

```
tldr {{[-l|--list]}} | shuf {{[-n|--head-count]}} 1 | xargs
tldr
```

tldrl

This command is an alias of **`tldr-lint`**.

- View documentation for the original command:

```
tldr tldr-lint
```

tlmgr arch

This command is an alias of **`tlmgr platform`**.

- View documentation for the original command:

```
tldr tlmgr platform
```

tlmgr backup

Manage backups of TeX Live packages.

The default backup directory is specified by the **`backupdir`** option, and can be obtained with **`tlmgr option`**.

More information: https://www.tug.org/texlive/doc/tlmgr.html#backup.

- Make a backup of one or more packages:

```
tlmgr backup {{package1 package2 ...}}
```

- Make a backup of all packages:

```
tlmgr backup --all
```

- Make a backup to a custom directory:

```
tlmgr backup {{package}} --backupdir {{path/to/
backup_directory}}
```

- Remove a backup of one or more packages:

```
tlmgr backup clean {{package1 package2 ...}}
```

- Remove all backups:

```
tlmgr backup clean --all
```

tlmgr candidates

Get available candidate repositories from which a TeX Live package can be installed.

More information: https://www.tug.org/texlive/doc/tlmgr.html#candidates-pkg.

- List all available repositories from which a package can be installed:

```
tlmgr candidates {{package}}
```

tlmgr check

Check the consistency of a TeX Live installation.

More information: https://www.tug.org/texlive/doc/tlmgr.html#check-option...-depends-executes-files-runfiles-texmfdbs-all.

- Check the consistency of the whole TeX Live installation:

```
tlmgr check all
```

- Check the consistency of the whole TeX Live information in verbose mode:

```
tlmgr check all -v
```

- Check for missing dependencies:

```
tlmgr check depends
```

- Check if all TeX Live executables are present:

```
tlmgr check executes
```

- Check if all files listed in the local TLPDB are present:

```
tlmgr check files
```

- Check for duplicate filenames in the runfiles sections:

```
tlmgr check runfiles
```

tlmgr conf

Manage the TeX Live configuration.

More information: https://www.tug.org/texlive/doc/tlmgr.html#conf.

- Show the current TeX Live configuration:

```
tlmgr conf
```

- Show the current `texmf`, `tlmgr`, or `updmap` configuration:

```
tlmgr conf {{texmf|tlmgr|updmap}}
```

- Show only a specific configuration option:

```
tlmgr conf {{texmf|tlmgr|updmap}} {{configuration_key}}
```

- Set a specific configuration option:

```
tlmgr conf {{texmf|tlmgr|updmap}} {{configuration_key}}
{{value}}
```

- Delete a specific configuration option:

```
tlmgr conf {{texmf|tlmgr|updmap}} --delete
{{configuration_key}}
```

- Disable the execution of system calls via `\write18`:

```
tlmgr conf texmf {{shell_escape}} {{0}}
```

- Show all additional `texmf` trees:

```
tlmgr conf auxtrees show
```

tlmgr dump-tlpdb

Dump the TeX Live package database.

More information: https://www.tug.org/texlive/doc/tlmgr.html#dump-tlpdb-option...---json.

- Dump the local package database:

```
tlmgr dump-tlpdb --local
```

- Dump the remote package database:

```
tlmgr dump-tlpdb --remote
```

- Dump the local package database as JSON:

```
tlmgr dump-tlpdb --local --json
```

tlmgr generate

Remake configuration files from information stored locally.

More information: https://www.tug.org/texlive/doc/tlmgr.html#generate.

- Remake the configuration file storing into a specific location:

```
tlmgr generate --dest {{output_file}}
```

- Remake the configuration file using a local configuration file:

```
tlmgr generate --localcfg {{local_configuration_file}}
```

- Run necessary programs after rebuilding configuration files:

```
tlmgr generate --rebuild-sys
```

tlmgr gui

Start a graphical user interface for **`tlmgr`**.

`tlmgr gui` depends on the package **`perl-tk`**, which has to be installed manually.

More information: https://www.tug.org/texlive/doc/tlmgr.html#gui.

- Start a GUI for `tlmgr`:

```
sudo tlmgr gui
```

- Start a GUI specifying the background color:

```
sudo tlmgr gui -background "{{#f39bc3}}"
```

- Start a GUI specifying the foreground color:

```
sudo tlmgr gui -foreground "{{#0ef3bd}}"
```

- Start a GUI specifying the font and font size:

```
sudo tlmgr gui -font "{{helvetica 18}}"
```

- Start a GUI setting a specific geometry:

```
sudo tlmgr gui -geometry {{width}}x{{height}}-{{xpos}}+{{ypos}}
```

- Start a GUI passing an arbitrary X resource string:

```
sudo tlmgr gui -xrm {{xresource}}
```

tlmgr info

Show information about TeX Live packages.

More information: https://www.tug.org/texlive/doc/tlmgr.html#info.

- List all available TeX Live packages, prefexing installed ones with `i`:

```
tlmgr info
```

- List all available collections:

```
tlmgr info collections
```

- List all available schemes:

```
tlmgr info scheme
```

- Show information about a specific package:

```
tlmgr info {{package}}
```

- List all files contained in a specific package:

```
tlmgr info {{package}} --list
```

- List all installed packages:

```
tlmgr info --only-installed
```

- Show only specific information about a package:

```
tlmgr info {{package}} --data "{{name}},{{category}},
{{installed}},{{size}},{{depends}},{{...}}"
```

- Print all available packages as JSON encoded array:

```
tlmgr info --json
```

tlmgr install

Install TeX Live packages.

More information: https://www.tug.org/texlive/doc/tlmgr.html#install-option...-pkg.

- Install a package and its dependencies:

```
sudo tlmgr install {{package}}
```

- Reinstall a package:

```
sudo tlmgr install --reinstall {{package}}
```

- Simulate installing a package without making any changes:

```
tlmgr install --dry-run {{package}}
```

- Install a package without its dependencies:

```
sudo tlmgr install --no-depends {{package}}
```

- Install a package from a specific file:

```
sudo tlmgr install --file {{path/to/package}}
```

tlmgr key

Manage GPG keys used to verify TeX Live databases.

More information: https://www.tug.org/texlive/doc/tlmgr.html#key.

- List all keys for TeX Live:

```
tlmgr key list
```

- Add a key from a specific file:

```
sudo tlmgr key add {{path/to/key.gpg}}
```

- Add a key from `stdin`:

```
cat {{path/to/key.gpg}} | sudo tlmgr key add -
```

- Remove a specific key by its ID:

```
sudo tlmgr key remove {{key_id}}
```

tlmgr option

TeX Live settings manager.

More information: https://www.tug.org/texlive/doc/tlmgr.html#option.

- List all TeX Live settings:

```
tlmgr option showall
```

- List all currently set Tex Live settings:

```
tlmgr option show
```

- Print all TeX Live settings in JSON format:

```
tlmgr option showall --json
```

- Show the value of a specific TeX Live setting:

```
tlmgr option {{setting}}
```

- Modify the value of a specific TeX Live setting:

```
tlmgr option {{setting}} {{value}}
```

- Set TeX Live to get future updates from the internet after installing from DVD:

```
tlmgr option {{repository}} {{https://mirror.ctan.org/
systems/texlive/tlnet}}
```

tlmgr paper

Manage paper size options of an TeX Live installation.

More information: https://www.tug.org/texlive/doc/tlmgr.html#paper.

- Show the default paper size used by all TeX Live programs:

```
tlmgr paper
```

- Set the default paper size for all TeX Live programs to A4:

```
sudo tlmgr paper {{a4}}
```

- Show the default paper size used by a specific TeX Live program:

```
tlmgr {{pdftex}} paper
```

- Set the default paper size for a specific TeX Live program to A4:

```
sudo tlmgr {{pdftex}} paper {{a4}}
```

- List all available paper sizes for a specific TeX Live program:

```
tlmgr {{pdftex}} paper --list
```

- Dump the default paper size used by all TeX Live programs in JSON format:

```
tlmgr paper --json
```

tlmgr path

Add or remove symlinks for TeX Live executables, man pages and info pages.

This command has to be re-run for files added in the future.

More information: https://www.tug.org/texlive/doc/tlmgr.html#path.

- Add symlinks to TeX Live files:

```
sudo tlmgr path add
```

- Remove symlinks to TeX Live files:

```
sudo tlmgr path remove
```

tlmgr pinning

The pinning action manages the pinning file.

More information: https://www.tug.org/texlive/doc/tlmgr.html#pinning.

- Show the current pinning data:

```
tlmgr pinning show
```

- Pin the matching the packages to the given repository:

```
tlmgr pinning add {{repository}} {{package1 package2 ...}}
```

- Remove any packages recorded in the pinning file matching the packages for the given repository:

```
tlmgr pinning remove {{repository}} {{package1 package2 ...}}
```

- Remove all pinning data for the given repository:

```
tlmgr pinning remove {{repository}} --all
```

tlmgr platform

Manage TeX Live platforms.

More information: https://www.tug.org/texlive/doc/tlmgr.html#platform.

- List all available platforms in the package repository:

```
tlmgr platform list
```

- Add the executables for a specific platform:

```
sudo tlmgr platform add {{platform}}
```

- Remove the executables for a specific platform:

```
sudo tlmgr platform remove {{platform}}
```

- Auto-detect and switch to the current platform:

```
sudo tlmgr platform set auto
```

- Switch to a specific platform:

```
sudo tlmgr platform set {{platform}}
```

tlmgr recreate-tlpdb

Recreate the TeX Live package database.

This command has a lack of documentation and should be used with caution.

More information: https://www.tug.org/texlive/tlmgr.html.

- Recreate the `texlive.tlpdb` database file and dump it to `stdout`:

```
sudo tlmgr recreate-tlpdb
```

tlmgr remove

Uninstall TeX Live packages.

By default, removed packages will be backed up to **`./tlpkg/backups`** under the TL installation directory.

More information: https://www.tug.org/texlive/doc/tlmgr.html#remove-option...-pkg.

- Uninstall a TeX Live package:

```
sudo tlmgr remove {{package}}
```

- Simulate uninstalling a package without making any changes:

```
tlmgr remove --dry-run {{package}}
```

- Uninstall a package without its dependencies:

```
sudo tlmgr remove --no-depends {{package}}
```

- Uninstall a package and back it up to a specific directory:

```
sudo tlmgr remove --backupdir {{path/to/directory}} {{package}}
```

- Uninstall all of TeX Live, asking for confirmation:

```
sudo tlmgr remove --all
```

tlmgr repository

Manage repositories of a TeX Live installation.

More information: https://www.tug.org/texlive/doc/tlmgr.html#repository.

- List all configured repositories and their tags (if set):

```
tlmgr repository list
```

- List all packages available in a specific repository:

```
tlmgr repository list {{path|url|tag}}
```

- Add a new repository with a specific tag (the tag is not required):

```
sudo tlmgr repository add {{path|url}} {{tag}}
```

- Remove a specific repository:

```
sudo tlmgr repository remove {{path|url|tag}}
```

- Set a new list of repositories, overwriting the previous list:

```
sudo tlmgr repository set {{path|url|tag}}#{{tag}} {{path|
url|tag}}#{{tag}} {{...}}
```

- Show the verification status of all configured repositories:

```
tlmgr repository status
```

tlmgr restore

Restore package backups created with **`tlmgr backup`**.

The default backup directory is specified by the **`backupdir`** option, and can be obtained with **`tlmgr option`**.

More information: https://www.tug.org/texlive/doc/tlmgr.html#restore.

- List all available backup revisions for all packages:

```
tlmgr restore
```

- List all available backup revisions for a specific package:

```
tlmgr restore {{package}}
```

- Restore a specific revision of a specific package:

```
tlmgr restore {{package}} {{revision}}
```

- Restore the latest revision of all backed-up packages:

```
tlmgr restore --all
```

- Restore a package from a custom backup directory:

```
tlmgr restore {{package}} {{revision}} --backupdir {{path/to/backup_directory}}
```

- Perform a dry-run and print all taken actions without making them:

```
tlmgr restore --dry-run {{package}} {{revision}}
```

tlmgr search

Search for TeX Live packages using (Perl) **regex**.

More information: https://www.tug.org/texlive/doc/tlmgr.html#search.

- Search for a package name and descriptions of all locally installed packages from a specific `regex`:

```
tlmgr search "{{regex}}"
```

- Search for all file names of all locally installed packages from a `regex`:

```
tlmgr search --file "{{regex}}"
```

- Search for all file names, package names, and descriptions of all locally installed packages from a `regex`:

```
tlmgr search --all "{{regex}}"
```

- Search the TeX Live database, instead of the local installation:

```
tlmgr search --global "{{regex}}"
```

- Restrict the matches for package names and descriptions (but not for file names) to whole words:

```
tlmgr search --all --word "{{regex}}"
```

tlmgr shell

Start an interactive shell of the native TeX Live manager.

More information: https://www.tug.org/texlive/doc/tlmgr.html#shell.

- Start an interactive shell of `tlmgr`:

```
tlmgr shell
```

- Run any `tlmgr` subcommand in the interactive shell:

```
{{subcommand}} {{arguments}}
```

- Quit the interactive shell:

```
quit
```

- List all TeX Live variables:

```
get
```

- Get the value of a TeX Live variable:

```
get {{variable}}
```

- Set the value of a TeX Live variable:

```
set {{variable}} {{value}}
```

- Restart the interactive shell:

```
restart
```

- Display the version of the current protocol:

```
protocol
```

tlmgr update

Update TeX Live packages.

More information: https://www.tug.org/texlive/doc/tlmgr.html#update-option...-pkg.

- Update all TeX Live packages:

```
sudo tlmgr update --all
```

- Update tlmgr itself:

```
sudo tlmgr update --self
```

- Update a specific package:

```
sudo tlmgr update {{package}}
```

- Update all except a specific package:

```
sudo tlmgr update --all --exclude {{package}}
```

- Update all packages, making a backup of the current packages:

```
sudo tlmgr update --all --backup
```

- Update a specific package without updating its dependencies:

```
sudo tlmgr update --no-depends {{package}}
```

- Simulate updating all packages without making any changes:

```
sudo tlmgr update --all --dry-run
```

tlmgr

Manage packages and configuration options of an existing TeX Live installation.

Some subcommands such as **`paper`** have their own usage documentation.

More information: https://www.tug.org/texlive/doc/tlmgr.html#ACTIONS.

- Install a package and its dependencies:

```
tlmgr install {{package}}
```

- Remove a package and its dependencies:

```
tlmgr remove {{package}}
```

- Display information about a package:

```
tlmgr info {{package}}
```

- Update all packages:

```
tlmgr update --all
```

- Show possible updates without updating anything:

```
tlmgr update --list
```

- Start a GUI version of tlmgr:

```
tlmgr gui
```

- List all TeX Live configurations:

```
tlmgr conf
```

tmpmail

A temporary email right from your terminal written in POSIX sh.

More information: https://github.com/sdushantha/tmpmail#usage.

- Create a temporary inbox:

```
tmpmail {{[-g|--generate]}}
```

- List messages and their numeric ID:

```
tmpmail
```

- Display the most recent received email:

```
tmpmail {{[-r|--recent]}}
```

- Open a specific message:

```
tmpmail {{email_id}}
```

- View email as raw text without HTML tags:

```
tmpmail {{[-t|--text]}}
```

- Open email with a specific browser (default is w3m):

```
tmpmail {{[-b|--browser]}} {{browser}}
```

tmsu

Simple tool for tagging files.

More information: https://tmsu.org.

- Tag a specific file with multiple tags:

```
tmsu tag {{path/to/file.mp3}} {{music}} {{big-jazz}} {{mp3}}
```

- Tag multiple files:

```
tmsu tag --tags "{{music mp3}}" {{*.mp3}}
```

- List tags of specified file(s):

```
tmsu tags {{*.mp3}}
```

- List files with specified tag(s):

```
tmsu files {{big-jazz}} {{music}}
```

- List files with tags matching boolean expression:

```
tmsu files "{{(year >= 1990 and year <= 2000)}} and
{{grunge}}"
```

- Mount tmsu virtual filesystem to an existing directory:

```
tmsu mount {{path/to/directory}}
```

tmux

Terminal multiplexer.

It allows multiple sessions with windows, panes, and more.

See also: **`zellij`**, **`screen`**.

More information: https://github.com/tmux/tmux.

- Start a new session:

  `tmux`

- Start a new named [s]ession:

  `tmux {{[new|new-session]}} -s {{name}}`

- List existing sessions:

  `tmux {{[ls|list-sessions]}}`

- Attach to the most recently used session:

  `tmux {{[a|attach]}}`

- Detach from the current session (inside a tmux session):

  `<Ctrl b><d>`

- Create a new window (inside a tmux session):

  `<Ctrl b><c>`

- Switch between sessions and windows (inside a tmux session):

  `<Ctrl b><w>`

- Kill a session by [t]arget name:

  `tmux kill-session -t {{name}}`

tmuxinator

Create and manage tmux sessions easily.

More information: https://github.com/tmuxinator/tmuxinator.

- Create a new project:

```
tmuxinator new {{project}}
```

- Edit a project:

```
tmuxinator edit {{project}}
```

- List projects:

```
tmuxinator list
```

- Start a tmux session based on project:

```
tmuxinator start {{project}}
```

- Stop a project's tmux session:

```
tmuxinator stop {{project}}
```

todo

A simple, standards-based, cli todo manager.

More information: https://todoman.readthedocs.io.

- List startable tasks:

```
todo list --startable
```

- Add a new task to the work list:

```
todo new {{thing_to_do}} --list {{work}}
```

- Add a location to a task with a given ID:

```
todo edit --location {{location_name}} {{task_id}}
```

- Show details about a task:

```
todo show {{task_id}}
```

- Mark tasks with the specified IDs as completed:

```
todo done {{task_id1 task_id2 ...}}
```

- Delete a task:

```
todo delete {{task_id}}
```

- Delete done tasks and reset the IDs of the remaining tasks:

```
todo flush
```

todo.sh

Simple and extensible shell script for managing your **`todo.txt`** file.

More information: https://github.com/todotxt/todo.txt-cli.

- List every item:

```
todo.sh ls
```

- Add an item with project and context tags:

```
todo.sh add '{{description}} +{{project}} @{{context}}'
```

- Mark an item as [do]ne:

```
todo.sh do {{item_no}}
```

- Remove an item:

```
todo.sh rm {{item_no}}
```

- Set an item's [pri]ority (A-Z):

```
todo.sh pri {{item_no}} {{priority}}
```

- Replace an item:

```
todo.sh replace {{item_no}} '{{new_description}}'
```

todoist

Access https://todoist.com from the command-line.

More information: https://github.com/sachaos/todoist.

- Add a task:

```
todoist add "{{task_name}}"
```

- Add a high priority task with a label, project, and due date:

```
todoist add "{{task_name}}" --priority {{1}} --label-ids
"{{label_id}}" --project-name "{{project_name}}" --date
"{{tmr 9am}}"
```

- Add a high priority task with a label, project, and due date in quick mode:

```
todoist quick '#{{project_name}} "{{tmr 9am}}" p{{1}}
{{task_name}} @{{label_name}}'
```

- List all tasks with a header and color:

```
todoist --header --color list
```

- List all high priority tasks:

```
todoist list --filter p{{1}}
```

- List today's tasks with high priority that have the specified label:

```
todoist list --filter '(@{{label_name}} | {{today}}) &
p{{1}}'
```

todoman

This command is an alias of **todo**.

- View documentation for the original command:

```
tldr todo
```

tofu fmt

Format configuration according to OpenTofu language style conventions.

More information: https://opentofu.org/docs/cli/commands/fmt/.

- Format the configuration in the current directory:

```
tofu fmt
```

- Format the configuration in the current directory and subdirectories:

```
tofu fmt -recursive
```

- Display diffs of formatting changes:

```
tofu fmt -diff
```

- Do not list files that were formatted to `stdout`:

```
tofu fmt -list=false
```

tofu output

Export structured data about your OpenTofu resources.

More information: https://opentofu.org/docs/cli/commands/output/.

- With no additional arguments, `output` will display all outputs for the root module:

  ```
  tofu output
  ```

- Output only a value with specific name:

  ```
  tofu output {{name}}
  ```

- Convert the output value to a raw string (useful for shell scripts):

  ```
  tofu output -raw
  ```

- Format the outputs as a JSON object, with a key per output (useful with `jq`):

  ```
  tofu output -json
  ```

tofu plan

Generate and show OpenTofu execution plans.

More information: https://opentofu.org/docs/cli/commands/plan/.

- Generate and show the execution plan in the currently directory:

```
tofu plan
```

- Show a plan to destroy all remote objects that currently exist:

```
tofu plan -destroy
```

- Show a plan to update the Tofu state and output values:

```
tofu plan -refresh-only
```

- Specify values for input variables:

```
tofu plan -var '{{name1}}={{value1}}' -var '{{name2}}
={{value2}}'
```

- Focus Tofu's attention on only a subset of resources:

```
tofu plan -target {{resource_type.resource_name[instance
index]}}
```

- Output a plan as JSON:

```
tofu plan -json
```

- Write a plan to a specific file:

```
tofu plan -no-color > {{path/to/file}}
```

tofu

Create and deploy infrastructure as code to cloud providers. Open-source fork of Terraform.

More information: https://opentofu.org/docs/cli/commands/.

- Initialize a new or existing OpenTofu configuration:

  ```
  tofu init
  ```

- Verify that the configuration files are syntactically valid:

  ```
  tofu validate
  ```

- Format configuration according to OpenTofu language style conventions:

  ```
  tofu fmt
  ```

- Generate and show an execution plan:

  ```
  tofu plan
  ```

- Build or change infrastructure:

  ```
  tofu apply
  ```

- Destroy Tofu-managed infrastructure:

  ```
  tofu destroy
  ```

toipe

Yet another typing test, but crab flavoured.

A trusty terminal typing tester.

More information: https://github.com/Samyak2/toipe.

- Start the typing test with the default wordlist:

```
toipe
```

- Use a specific wordlist:

```
toipe {{[-w|--wordlist]}} {{wordlist_name}}
```

- Use a custom wordlist:

```
toipe {{[-f|--file]}} {{path/to/file}}
```

- Specify the number of words on each test:

```
toipe {{[-n|--num]}} {{number_of_words}}
```

- Include punctuation:

```
toipe {{[-p|--punctuation]}}
```

tokei

Display statistics about code.

More information: https://github.com/XAMPPRocky/tokei.

- Display a report for the code in a directory and all subdirectories:

```
tokei {{path/to/directory}}
```

- Display a report for a directory excluding `.min.js` files:

```
tokei {{path/to/directory}} {{[-e|--exclude]}} {{*.min.js}}
```

- Display statistics for individual files in a directory:

```
tokei {{path/to/directory}} {{[-f|--files]}}
```

- Display a report for all files of type Rust and Markdown:

```
tokei {{path/to/directory}} {{[-t|--type]}} {{Rust}},
{{Markdown}}
```

topgrade

Update all applications on the system.

More information: https://github.com/r-darwish/topgrade.

- Run updates:

```
topgrade
```

- Say yes to all updates:

```
topgrade {{[-y|--yes]}}
```

- Cleanup temporary/old files:

```
topgrade {{[-c|--cleanup]}}
```

- Disable a certain update operation:

```
topgrade --disable {{operation}}
```

- Only perform a certain update operation:

```
topgrade --only {{operation}}
```

- Edit the configuration file with default editor:

```
topgrade --edit-config
```

topydo

A to-do list application that uses the todo.txt format.

More information: https://github.com/topydo/topydo.

- Add a to-do to a specific project with a given context:

```
topydo add "{{todo_message}} +{{project_name}}
@{{context_name}}"
```

- Add a to-do with a due date of tomorrow with a priority of `A`:

```
topydo add "(A) {{todo _message}} due:{{1d}}"
```

- Add a to-do with a due date of Friday:

```
topydo add "{{todo_message}} due:{{fri}}"
```

- Add a non-strict repeating to-do (next due = now + rec):

```
topydo add "water flowers due:{{mon}} rec:{{1w}}"
```

- Add a strict repeating to-do (next due = current due + rec):

```
topydo add "{{todo_message}} due:{{2020-01-01}} rec:{{+1m}}"
```

- Revert the last `topydo` command executed:

```
topydo revert
```

touch

Create files and set access/modification times.

More information: https://manned.org/touch.

- Create specific files:

```
touch {{path/to/file1 path/to/file2 ...}}
```

- Set the file [a]ccess or [m]odification times to the current one and don't create file if it doesn't exist:

```
touch {{[-c|--no-create]}} -{{a|m}} {{path/to/file1 path/to/file2 ...}}
```

- Set the file [t]ime to a specific value and don't create file if it doesn't exist:

```
touch {{[-c|--no-create]}} -t {{YYYYMMDDHHMM.SS}} {{path/to/file1 path/to/file2 ...}}
```

- Set the files' timestamp to the reference file's timestamp, and do not create the file if it does not exist:

```
touch {{[-c|--no-create]}} {{[-r|--reference]}} {{path/to/reference_file}} {{path/to/file1 path/to/file2 ...}}
```

- Set the timestamp by parsing a string:

```
touch {{[-d|--date]}} "{{last year|5 hours|next thursday|nov 14|...}}" {{path/to/file}}
```

- Create multiple files with an increasing number:

```
touch {{path/to/file{1..10}}}
```

- Create multiple files with a letter range:

```
touch {{path/to/file{a..z}}}
```

tox

Automate Python testing across multiple Python versions.

Use tox.ini to configure environments and test command.

More information: https://github.com/tox-dev/tox.

- Run tests on all test environments:

  ```
  tox
  ```

- Create a `tox.ini` configuration:

  ```
  tox-quickstart
  ```

- List the available environments:

  ```
  tox {{[-a|--listenvs-all]}}
  ```

- Run tests on a specific environment (e.g. Python 3.6):

  ```
  tox -e {{py36}}
  ```

- Force the virtual environment to be recreated:

  ```
  tox {{[-r|--recreate]}} -e {{py27}}
  ```

tpp

Command-Line based presentation tool.

More information: https://github.com/cbbrowne/tpp.

- View a presentation:

```
tpp {{path/to/file}}
```

- Output a presentation:

```
tpp -t {{type}} -o {{path/to/output}} {{path/to/file}}
```

tput

View and modify terminal settings and capabilities.

More information: https://manned.org/tput.

- Move the cursor to a screen location:

```
tput cup {{row}} {{column}}
```

- Set foreground (af) or background (ab) color:

```
tput {{setaf|setab}} {{ansi_color_code}}
```

- Reverse text and background colors:

```
tput rev
```

- Reset all terminal text attributes:

```
tput sgr0
```

- Show number of columns, lines, or colors:

```
tput {{cols|lines|colors}}
```

- Enable or disable word wrap:

```
tput {{smam|rmam}}
```

- Hide or show the terminal cursor:

```
tput {{civis|cnorm}}
```

- Save or restore terminal text status (smcup also captures scroll wheel events):

```
tput {{smcup|rmcup}}
```

tqdm

Show progress over time of a command.

More information: https://tqdm.github.io/docs/cli/.

- Show iterations per second and use `stdout` afterwards:

```
{{seq 10000000}} | tqdm | {{command}}
```

- Create a progress bar:

```
{{seq 10000000}} | tqdm --total {{10000000}} | {{command}}
```

- Create an archive out of a directory and use the file count of that directory to create a progress bar:

```
zip {{[-r|--recurse-paths]}} {{path/to/archive.zip}} {{path/to/directory}} | tqdm --total $(find {{path/to/directory}} | wc {{[-l|--lines]}}) --unit files --null
```

- Create an archive with tar and create a progress bar (system agnostic, GNU tar uses `stdout` while BSD tar uses `stderr`):

```
tar vzcf {{path/to/archive.tar.gz}} {{path/to/directory}} 2>&1 | tqdm --total $(find {{path/to/directory}} | wc {{[-l|--lines]}}) --unit files --null
```

tr

Translate characters: run replacements based on single characters and character sets.

More information: https://www.gnu.org/software/coreutils/manual/html_node/tr-invocation.html.

- Replace all occurrences of a character in a file, and print the result:

```
tr < {{path/to/file}} {{find_character}}
{{replace_character}}
```

- Replace all occurrences of a character from another command's output:

```
echo {{text}} | tr {{find_character}} {{replace_character}}
```

- Map each character of the first set to the corresponding character of the second set:

```
tr < {{path/to/file}} '{{abcd}}' '{{jkmn}}'
```

- Delete all occurrences of the specified set of characters from the input:

```
tr < {{path/to/file}} {{[-d|--delete]}}
'{{input_characters}}'
```

- Compress a series of identical characters to a single character:

```
tr < {{path/to/file}} {{[-s|--squeeze-repeats]}}
'{{input_characters}}'
```

- Translate the contents of a file to upper-case:

```
tr < {{path/to/file}} "[:lower:]" "[:upper:]"
```

- Strip out non-printable characters from a file:

```
tr < {{path/to/file}} {{[-cd|--complement --delete]}}
"[:print:]"
```

traceroute

Print the route packets trace to network host.

See also: **`mtr`**.

More information: https://manned.org/traceroute.

- Traceroute to a host:

```
traceroute {{example.com}}
```

- Disable IP address and host name mapping:

```
traceroute -n {{example.com}}
```

- Specify wait time in seconds for response:

```
traceroute {{[-w|--wait]}} {{0.5}} {{example.com}}
```

- Specify number of queries per hop:

```
traceroute {{[-q|--queries]}} {{5}} {{example.com}}
```

- Specify size in bytes of probing packet:

```
traceroute {{example.com}} {{42}}
```

- Determine the MTU to the destination:

```
traceroute --mtu {{example.com}}
```

- Use ICMP instead of UDP for tracerouting:

```
traceroute {{[-I|--icmp]}} {{example.com}}
```

traefik

An HTTP reverse proxy and load balancer.

More information: https://traefik.io.

- Start the server with the default configuration:

  ```
  traefik
  ```

- Start the server with a custom configuration file:

  ```
  traefik --ConfigFile {{config_file.toml}}
  ```

- Start the server with cluster mode enabled:

  ```
  traefik --cluster
  ```

- Start server with web UI enabled:

  ```
  traefik --web
  ```

trafilatura

A Python tool for web scraping and crawling that extracts main text, metadata, and comments from web pages.

Designed for creating text corpora and extracting structured content.

More information: https://trafilatura.readthedocs.io/en/latest/usage-cli.html#further-information.

- Extract text from a URL:

  ```
  trafilatura {{[-u|--URL]}} {{url}}
  ```

- Extract text and save to a file:

  ```
  trafilatura {{[-u|--URL]}} {{url}} {{[-o|--output-dir]}} {{path/to/output.txt}}
  ```

- Extract text in JSON format:

  ```
  trafilatura {{[-u|--URL]}} {{url}} --json
  ```

- Extract text from multiple URLs listed in a file:

  ```
  trafilatura {{[-i|--input-file]}} {{path/to/url_list.txt}}
  ```

- Crawl a website using its sitemap:

  ```
  trafilatura --sitemap {{url_to_sitemap.xml}}
  ```

- Extract text while preserving HTML formatting:

  ```
  trafilatura {{[-u|--URL]}} {{url}} --formatting
  ```

- Extract text including comments:

  ```
  trafilatura {{[-u|--URL]}} {{url}} --with-comments
  ```

- Display help:

  ```
  trafilatura {{[-h|--help]}}
  ```

trans

Translate Shell, a command-line translator.

More information: https://github.com/soimort/translate-shell.

- Translate a word (language is detected automatically):

```
trans "{{word_or_sentence_to_translate}}"
```

- Get a brief translation:

```
trans {{[-b|-brief]}} "{{word_or_sentence_to_translate}}"
```

- Translate a word into french:

```
trans :{{fr}} {{word}}
```

- Translate a word from German to English:

```
trans {{de}}:{{en}} {{Schmetterling}}
```

- Behave like a dictionary to get the meaning of a word:

```
trans {{[-d|-dictionary]}} {{word}}
```

transcode

Transcode video and audio codecs, and convert between media formats.

More information: https://manned.org/transcode.

- Create stabilization file to be able to remove camera shakes:

```
transcode -J stabilize -i {{input_file}}
```

- Remove camera shakes after creating stabilization file, transform video using XviD:

```
transcode -J transform -i {{input_file}} -y xvid -o {{output_file}}
```

- Resize the video to 640x480 pixels and convert to MPEG4 codec using XviD:

```
transcode -Z 640x480 -i {{input_file}} -y xvid -o {{output_file}}
```

transcrypt

Transparently encrypt files within a Git repository.

More information: https://github.com/elasticdog/transcrypt.

- Initialize an unconfigured repository:

```
transcrypt
```

- List the currently encrypted files:

```
git ls-crypt
```

- Display the credentials of a configured repository:

```
transcrypt {{[-d|--display]}}
```

- Initialize and decrypt a fresh clone of a configured repository:

```
transcrypt {{[-c|--cipher]}} {{cipher}}
```

- Rekey to change the encryption cipher or password:

```
transcrypt {{[-r|--rekey]}}
```

transfersh

An unofficial client for transfer.sh.

More information: https://github.com/AlpixTM/transfersh.

- Upload a file to transfer.sh:

```
transfersh {{path/to/file}}
```

- Upload a file showing a progress bar (requires Python package `requests_toolbelt`):

```
transfersh --progress {{path/to/file}}
```

- Upload a file using a different file name:

```
transfersh --name {{filename}} {{path/to/file}}
```

- Upload a file to a custom transfer.sh server:

```
transfersh --servername {{upload.server.name}} {{path/to/file}}
```

- Upload all files from a directory recursively:

```
transfersh --recursive {{path/to/directory}}/
```

- Upload a specific directory as an uncompressed tar:

```
transfersh -rt {{path/to/directory}}
```

transmission-cli

A lightweight, command-line BitTorrent client.

This tool has been deprecated, please see **`transmission-remote`**.

More information: https://manned.org/transmission-cli.

- Download a specific torrent:

```
transmission-cli {{url|magnet|path/to/file}}
```

- Download a torrent to a specific directory:

```
transmission-cli {{[-w|--download-dir]}} {{path/to/
download_directory}} {{url|magnet|path/to/file}}
```

- Create a torrent file from a specific file or directory:

```
transmission-cli --new {{path/to/source_file_or_directory}}
```

- Specify the download speed limit (in KB/s):

```
transmission-cli {{[-d|--downlimit]}} {{50}} {{url|magnet|
path/to/file}}
```

- Specify the upload speed limit (in KB/s):

```
transmission-cli {{[-u|--uplimit]}} {{50}} {{url|magnet|path/
to/file}}
```

- Use a specific port for connections:

```
transmission-cli {{[-p|--port]}} {{port_number}} {{url|
magnet|path/to/file}}
```

- Force encryption for peer connections:

```
transmission-cli {{[-er|--encryption-required]}} {{url|
magnet|path/to/file}}
```

- Use a Bluetack-formatted peer blocklist:

```
transmission-cli {{[-b|--blocklist]}} {{blocklist_url|path/
to/blocklist}} {{url|magnet|path/to/file}}
```

transmission-create

Create BitTorrent **.torrent** files.

See also: **transmission**.

More information: https://manned.org/transmission-create.

- Create a torrent with a specific piece size (in KB):

```
transmission-create {{[-o|--outfile]}} {{path/to/
example.torrent}} {{[-t|--tracker]}} {{tracker_announce_url}}
{{[-s|--piecesize]}} {{2048}} {{path/to/file_or_directory}}
```

- Create a private torrent with a specific piece size (in KB):

```
transmission-create {{[-p|--private]}} {{[-o|--outfile]}}
{{path/to/example.torrent}} {{[-t|--tracker]}}
{{tracker_announce_url}} {{[-s|--piecesize]}} {{2048}}
{{path/to/file_or_directory}}
```

- Create a torrent with a comment:

```
transmission-create {{[-o|--outfile]}} {{path/to/
example.torrent}} {{[-t|--tracker]}} {{tracker_url1}} {{[-
c|--comment]}} {{comment}} {{path/to/file_or_directory}}
```

- Create a torrent with multiple trackers:

```
transmission-create {{[-o|--outfile]}} {{path/to/
example.torrent}} {{[-t|--tracker]}} {{tracker_url1}} {{[-
t|--tracker]}} {{tracker_url2}} {{path/to/file_or_directory}}
```

- Display help page:

```
transmission-create {{[-h|--help]}}
```

transmission-daemon

Daemon controlled with **`transmission-remote`** or its web interface.

See also: **`transmission`**.

More information: https://manned.org/transmission-daemon.

- Start a headless `transmission` session:

```
transmission-daemon
```

- Start and watch a specific directory for new torrents:

```
transmission-daemon {{[-c|--watch-dir]}} {{path/to/
directory}}
```

- Dump daemon settings in JSON format:

```
transmission-daemon {{[-d|--dump-settings]}} > {{path/to/
file.json}}
```

- Start with specific settings for the web interface:

```
transmission-daemon {{[-t|--auth]}} {{[-u|--username]}}
{{username}} {{[-v|--password]}} {{password}} {{[-p|--port]}}
{{9091}} {{[-a|--allowed]}} {{127.0.0.1}}
```

transmission-edit

Modify announce URLs from torrent files.

See also: **`transmission`**.

More information: https://manned.org/transmission-edit.

- Add a URL to a torrent's announce list:

```
transmission-edit {{[-a|--add]}} {{http://example.com}} {{path/to/file.torrent}}
```

- Remove a URL from a torrent's announce list:

```
transmission-edit {{[-d|--delete]}} {{http://example.com}} {{path/to/file.torrent}}
```

- Update a tracker's passcode in a torrent file:

```
transmission-edit {{[-r|--replace]}} {{old-passcode}} {{new-passcode}} {{path/to/file.torrent}}
```

transmission-remote

Remote control utility for **`transmission-daemon`** and **`transmission`**.

More information: https://manned.org/transmission-remote.

- Add a torrent file or magnet link to Transmission and download to a specified directory:

```
transmission-remote {{hostname}} {{[-a|--all]}} {{torrent|
url}} {{[-w|--download-dir]}} /{{path/to/download_directory}}
```

- Change the default download directory:

```
transmission-remote {{hostname}} {{[-w|--download-dir]}} /
{{path/to/download_directory}}
```

- List all torrents:

```
transmission-remote {{hostname}} {{[-l|--list]}}
```

- Start torrent 1 and 2, stop torrent 3:

```
transmission-remote {{hostname}} {{[-t|--torrent]}}
"1,2" {{[-s|--start]}} {{[-t|--torrent]}} 3 {{[-S|--stop]}}
```

- Remove torrent 1 and 2, and also delete local data for torrent 2:

```
transmission-remote {{hostname}} {{[-t|--torrent]}} 1 {{[-
r|--remove]}} {{[-t|--torrent]}} 2 {{[-rad|--remove-and-
delete]}}
```

- Stop all torrents:

```
transmission-remote {{hostname}} {{[-t|--torrent]}} {{all}}
{{[-S|--stop]}}
```

- Move torrents 1-10 and 15-20 to a new directory (which will be created if it does not exist):

```
transmission-remote {{hostname}} {{[-t|--torrent]}}
"1-10,15-20" --move /{{path/to/new_directory}}
```

transmission-show

Get information about a torrent file.

See also: **`transmission`**.

More information: https://manned.org/transmission-show.

- Display metadata for a specific torrent:

```
transmission-show {{path/to/file.torrent}}
```

- Generate a magnet link for a specific torrent:

```
transmission-show {{[-m|--magnet]}} {{path/to/file.torrent}}
```

- Query a torrent's trackers and print the current number of peers:

```
transmission-show {{[-s|--scrape]}} {{path/to/file.torrent}}
```

transmission

A simple torrent client.

Transmission isn't a command, but a set of commands. See the pages below.

More information: https://transmissionbt.com/.

- View documentation for running Transmission's daemon:

```
tldr transmission-daemon
```

- View documentation for interacting with the daemon:

```
tldr transmission-remote
```

- View documentation for creating torrent files:

```
tldr transmission-create
```

- View documentation for modifying torrent files:

```
tldr transmission-edit
```

- View documentation for getting info about torrent files:

```
tldr transmission-show
```

- View documentation for the deprecated method for interacting with the daemon:

```
tldr transmission-cli
```

trap

Execute a command upon an event.

More information: https://manned.org/trap.1posix.

- List the commands and the names of the expected events:

```
trap
```

- Execute a command when a signal is received:

```
trap 'echo "Caught signal {{SIGHUP}}"' {{HUP}}
```

- Remove commands:

```
trap - {{HUP}} {{INT}}
```

trash-cli

This command is an alias of **`trash`**.

- View documentation for the original command:

  ```
  tldr trash
  ```

travis

Interface with Travis CI.

More information: https://github.com/travis-ci/travis.rb.

- Display the client version:

```
travis version
```

- Authenticate the CLI client against the server, using an authentication token:

```
travis login
```

- List repositories the user has permissions on:

```
travis repos
```

- Encrypt values in `.travis.yml`:

```
travis encrypt {{token}}
```

- Generate a `.travis.yml` file and enable the project:

```
travis init
```

trawl

Print out network interface information to the console, much like ifconfig/ipconfig/ip/ifdata.

More information: https://github.com/robphoenix/trawl.

- Show column names:

```
trawl -n
```

- Filter interface names using a case-insensitive `regex`:

```
trawl -f wi
```

- List available interfaces:

```
trawl -i
```

- Include the loopback interface:

```
trawl -l
```

trdsql

Execute SQL on CSV, LTSV, JSON, YAML, and TBLN files.

More information: https://noborus.github.io/trdsql/.

- Convert object data from multiple JSON files to a CSV file with header (`-oh`) and double quote:

```
trdsql -ocsv -oh "SELECT * FROM {{path/to/directory/*.json}}"
| sed 's/\([^,]*\)/"&"/g' > {{path/to/file.csv}}
```

- Interpret JSON list as a table and put objects inside as columns (path/to/file.json: `{"list":[{"age":"26","name":"Tanaka"}]}`):

```
trdsql "SELECT * FROM {{path/to/file.json}}::.list"
```

- Manipulate complex SQL query with data from multiple CSV files with first line is header (`-ih`):

```
trdsql -icsv -ih "SELECT {{column1,column2}} FROM {{path/to/
file*.csv}} WHERE column2 != '' ORDER BY column1 GROUP BY
column1"
```

- Merge content of 2 CSV files to one CSV file:

```
trdsql "SELECT {{column1,colum2}} FROM {{path/to/file1.csv}}
UNION SELECT {{column1,column2}} FROM {{path/to/file2.csv}}"
```

- Connect to PostgreSQL database:

```
trdsql -driver postgres -dsn "host={{hostname}} port={{5433}}
dbname={{database_name}}" "SELECT 1"
```

- Create table data to MySQL database from CSV file:

```
trdsql -driver mysql -dsn "{{username}}:{{password}}
@{{hostname}}/{{database}}" -ih "CREATE TABLE {{table}}
({{column1}} int, {{colum2}} varchar(20)) AS SELECT
{{column3}} AS {{column1}},{{column2}} FROM {{path/to/
header_file.csv}}"
```

- Show data from compress log files:

```
trdsql -iltsv "SELECT * FROM {{path/to/access.log.gz}}"
```

tre

Show the contents of the current directory as a tree.

Respects the **`.gitignore`** settings by default.

More information: https://github.com/dduan/tre.

- Print directories only:

```
tre --directories
```

- Print JSON containing files in the tree hierarchy instead of the normal tree diagram:

```
tre --json
```

- Print files and directories up to the specified depth limit (where 1 means the current directory):

```
tre --limit {{depth}}
```

- Print all hidden files and directories using the specified colorization mode:

```
tre --all --color {{automatic|always|never}}
```

- Print files within the tree hierarchy, assigning a shell alias to each file that, when called, will open the associated file using the provided `command` (or in `$EDITOR` by default):

```
tre --editor {{command}}
```

- Print files within the tree hierarchy, excluding all paths that match the provided `regex`:

```
tre --exclude {{regex}}
```

- Display version:

```
tre --version
```

- Display help:

```
tre --help
```

tred

Compute the transitive reduction of directed graphs.

Graphviz filters: **`acyclic`**, **`bcomps`**, **`comps`**, **`edgepaint`**, **`gvcolor`**, **`gvpack`**, **`mingle`**, **`nop`**, **`sccmap`**, **`tred`**, & **`unflatten`**.

More information: https://www.graphviz.org/pdf/tred.1.pdf.

- Construct the transitive reduction graph of one or more directed graphs:

```
tred {{path/to/input1.gv path/to/input2.gv ...}} > {{path/to/output.gv}}
```

- Display help:

```
tred -?
```

tree

Show the contents of the current directory as a tree.

More information: https://manned.org/tree.

- Print files and directories up to `num` levels of depth (where 1 means the current directory):

```
tree -L {{num}}
```

- Print directories only:

```
tree -d
```

- Print hidden files too with colorization on:

```
tree -a -C
```

- Print the tree without indentation lines, showing the full path instead (use `-N` to not escape non-printable characters):

```
tree -i -f
```

- Print the size of each file and the cumulative size of each directory, in human-readable format:

```
tree -s -h --du
```

- Print files within the tree hierarchy, using a wildcard (glob) pattern, and pruning out directories that don't contain matching files:

```
tree -P '{{*.txt}}' --prune
```

- Print directories within the tree hierarchy, using the wildcard (glob) pattern, and pruning out directories that aren't ancestors of the wanted one:

```
tree -P {{directory_name}} --matchdirs --prune
```

- Print the tree ignoring the given directories:

```
tree -I '{{directory_name1|directory_name2}}'
```

trip

A network diagnostic tool.

Combines the functionality of **`traceroute`** and **`ping`**.

Designed to assist with the analysis of networking issues.

More information: https://trippy.rs/reference/cli/.

- Basic usage with default parameters:

```
sudo trip {{example.com}}
```

- Trace without requiring elevated privileges (supported platforms only):

```
trip {{example.com}} --unprivileged
```

- Trace using `IPv6` only:

```
sudo trip {{example.com}} --ipv6
```

- Trace using the `udp` protocol:

```
sudo trip {{example.com}} --protocol {{udp}}
```

- Use custom destination port `443` for `tcp` tracing:

```
sudo trip {{example.com}} --protocol {{tcp}} --target-port
{{443}}
```

- Use custom source port `5000` for `udp` tracing:

```
sudo trip {{example.com}} --protocol {{udp}} --source-port
{{5000}}
```

trivy

Scanner for vulnerabilities in container images, file systems, and Git repositories, as well as for configuration issues.

More information: https://aquasecurity.github.io/trivy.

- Scan a Docker image for vulnerabilities and exposed secrets:

```
trivy image {{image:tag}}
```

- Scan a Docker image filtering the output by severity:

```
trivy image {{[-s|--severity]}} {{HIGH,CRITICAL}} {{alpine:3.15}}
```

- Scan a Docker image ignoring any unfixed/unpatched vulnerabilities:

```
trivy image --ignore-unfixed {{alpine:3.15}}
```

- Scan the filesystem for vulnerabilities and misconfigurations:

```
trivy fs --security-checks {{vuln,config}} {{path/to/project_directory}}
```

- Scan a IaC (Terraform, CloudFormation, ARM, Helm and Dockerfile) directory for misconfigurations:

```
trivy config {{path/to/iac_directory}}
```

- Scan a local or remote Git repository for vulnerabilities:

```
trivy repo {{path/to/local_repository_directory|remote_repository_URL}}
```

- Scan a Git repository up to a specific commit hash:

```
trivy repo --commit {{commit_hash}} {{repository}}
```

- Generate output with a SARIF template:

```
trivy image {{[-f|--format]}} {{template}} {{[-t|--template]}} "{{@sarif.tpl}}" {{[-o|--output]}} {{path/to/report.sarif}} {{image:tag}}
```

troff

Typesetting processor for the groff (GNU Troff) document formatting system.

See also: **`groff`**.

More information: https://manned.org/troff.

- Format output for a PostScript printer, saving the output to a file:

```
troff {{path/to/input.roff}} | grops > {{path/to/output.ps}}
```

- Format output for a PostScript printer using the me macro package, saving the output to a file:

```
troff -{{me}} {{path/to/input.roff}} | grops > {{path/to/output.ps}}
```

- Format output as ASCII text using the man macro package:

```
troff -T {{ascii}} -{{man}} {{path/to/input.roff}} | grotty
```

- Format output as a pdf file, saving the output to a file:

```
troff -T {{pdf}} {{path/to/input.roff}} | gropdf > {{path/to/output.pdf}}
```

true

Returns a successful exit status code of 0.

Use this with the || operator to make a command always exit with 0.

More information: https://www.gnu.org/software/coreutils/manual/html_node/true-invocation.html.

- Return a successful exit code:

```
true
```

truffle

Develop smart contracts for running services on the Ethereum blockchain.

More information: https://www.trufflesuite.com/docs/truffle/reference/truffle-commands.

- Download a pre-built Truffle project (Truffle Box):

  ```
  truffle unbox {{box_name}}
  ```

- Compile contract source files in the current directory:

  ```
  truffle compile
  ```

- Run JavaScript and Solidity tests:

  ```
  truffle test
  ```

- Run migrations to deploy contracts:

  ```
  truffle migrate
  ```

- Display help for a subcommand:

  ```
  truffle help {{subcommand}}
  ```

trufflehog

Find and verify credentials in files, Git repositories, S3 buckets, and Docker images.

More information: https://github.com/trufflesecurity/trufflehog.

- Scan a Git repository for verified secrets:

```
trufflehog git {{https://github.com/trufflesecurity/test_keys}} --only-verified
```

- Scan a GitHub organization for verified secrets:

```
trufflehog github --org {{trufflesecurity}} --only-verified
```

- Scan a GitHub repository for verified keys and get JSON output:

```
trufflehog git {{https://github.com/trufflesecurity/test_keys}} --only-verified --json
```

- Scan a GitHub repository along with its Issues and Pull Requests:

```
trufflehog github --repo {{https://github.com/trufflesecurity/test_keys}} --issue-comments --pr-comments
```

- Scan an S3 bucket for verified keys:

```
trufflehog s3 --bucket {{bucket name}} --only-verified
```

- Scan S3 buckets using IAM Roles:

```
trufflehog s3 --role-arn {{iam-role-arn}}
```

- Scan individual files or directories:

```
trufflehog filesystem {{path/to/file_or_directory1 path/to/file_or_directory2 ...}}
```

- Scan a Docker image for verified secrets:

```
trufflehog docker --image {{trufflesecurity/secrets}} --only-verified
```

truncate

Shrink or extend the size of a file to the specified size.

More information: https://www.gnu.org/software/coreutils/manual/html_node/truncate-invocation.html.

- Set a size of 10 GB to an existing file, or create a new file with the specified size:

```
truncate {{[-s|--size]}} 10G {{path/to/file}}
```

- Extend the file size by 50 MiB, fill with holes (which reads as zero bytes):

```
truncate {{[-s|--size]}} +50M {{path/to/file}}
```

- Shrink the file by 2 GiB, by removing data from the end of file:

```
truncate {{[-s|--size]}} -2G {{path/to/file}}
```

- Empty the file's content:

```
truncate {{[-s|--size]}} 0 {{path/to/file}}
```

- Empty the file's content, but do not create the file if it does not exist:

```
truncate {{[-cs|--no-create --size]}} 0 {{path/to/file}}
```

trunk

Bundle and serve Rust web apps with CI/CD pipelines.

More information: https://docs.trunk.io/references/cli.

- Start local/production server with hot reloading:

```
trunk serve --port {{port}} --release --proxy-backend {{URL}}
```

- Build for production at root or subdirectory:

```
trunk build --release --dist {{path/to/distribution}} --public-url /{{path/to/app_subdirectory}}
```

- List all available tools in the repo and if they are enabled:

```
trunk tools list
```

- Enable/disable a tool at a specific version:

```
trunk tools {{enable|disable}} {{tool}}@{{version}}
```

- Print an action's execution history:

```
trunk actions history {{action}}
```

ts-node

Run TypeScript code directly, without any compiling.

More information: https://typestrong.org/ts-node/docs/options.

- Execute a TypeScript file without compiling (`node` + `tsc`):

```
ts-node {{path/to/file.ts}}
```

- Execute a TypeScript file without loading `tsconfig.json`:

```
ts-node --skipProject {{path/to/file.ts}}
```

- Evaluate TypeScript code passed as a literal:

```
ts-node {{[-e|--eval]}} '{{console.log("Hello World")}}'
```

- Execute a TypeScript file in script mode:

```
ts-node --script-mode {{path/to/file.ts}}
```

- Transpile a TypeScript file to JavaScript without executing it:

```
ts-node {{[-T|--transpileOnly]}} {{path/to/file.ts}}
```

- Display help:

```
ts-node {{[-h|--help]}}
```

ts

Add timestamps to every line from **`stdin`**.

More information: https://manned.org/ts.

- Add a timestamp to the beginning of each line:

```
{{command}} | ts
```

- Add timestamps with microsecond precision:

```
{{command}} | ts "{{%b %d %H:%M:%.S}}"
```

- Add [i]ncremental timestamps with microsecond precision, starting from zero:

```
{{command}} | ts -i "{{%H:%M:%.S}}"
```

- Convert existing timestamps in a text file (eg. a log file) into [r]elative format:

```
cat {{path/to/file}} | ts -r
```

tsc

TypeScript compiler.

More information: https://www.typescriptlang.org/docs/handbook/compiler-options.html.

- Initialize a TypeScript project:

```
tsc --init
```

- Compile a TypeScript file into a JavaScript file with the same name:

```
tsc {{path/to/file.ts}}
```

- Compile a TypeScript file into JavaScript using a specific target syntax (default is ES3):

```
tsc {{[-t|--target]}} {{ES5|ES2015|ES2016|ES2017|ES2018|
ESNEXT|...}} {{path/to/file.ts}}
```

- Compile a TypeScript file into a JavaScript file with a custom name:

```
tsc --outFile {{path/to/output_file.js}} {{path/to/
input_file.ts}}
```

- Compile all `.ts` files of a TypeScript project defined in a `tsconfig.json` file (`--build` can be omitted to build the project in the current working directory):

```
tsc {{[-b|--build]}} {{path/to/tsconfig.json}}
```

- Run the compiler using command-line options and arguments fetched from a text file:

```
tsc @{{args.txt}}
```

- Type-check multiple JavaScript files, and output only the errors:

```
tsc --allowJs --checkJs --noEmit {{src/**/*.js}}
```

- Run the compiler in watch mode, which automatically recompiles code when it changes:

```
tsc {{[-w|--watch]}}
```

tslint

A pluggable linting utility for TypeScript.

More information: https://palantir.github.io/tslint.

- Create TSLint config:

```
tslint --init
```

- Lint on a given set of files:

```
tslint {{path/to/file1.js path/to/file2.js ...}}
```

- Fix lint issues:

```
tslint --fix
```

- Lint with the configuration file in the project root:

```
tslint --project {{path/to/project_root}}
```

tsort

Perform a topological sort.

A common use is to show the dependency order of nodes in a directed acyclic graph.

More information: https://www.gnu.org/software/coreutils/manual/html_node/tsort-invocation.html.

- Perform a topological sort consistent with a partial sort per line of input separated by blanks:

```
tsort {{path/to/file}}
```

- Perform a topological sort consistent on strings:

```
echo -e "{{UI Backend\nBackend Database\nDocs UI}}" | tsort
```

tspin

A log file highlighter based on the **`less`** pager and basically behaves like any pager.

More information: https://github.com/bensadeh/tailspin.

- Read from file and view in `less`:

```
tspin {{path/to/application.log}}
```

- Read from another command and print to `stdout`:

```
journalctl {{[-b|--boot]}} {{[-f|--follow]}} | tspin
```

- Read from file and print to `stdout`:

```
tspin {{path/to/application.log}} {{[-p|--print]}}
```

- Read from `stdin` and print to `stdout`:

```
echo "2021-01-01 12:00:00 [INFO] This is a log message" |
tspin
```

tsv-filter

Filter lines of a TSV file by running tests against individual fields.

More information: https://github.com/eBay/tsv-utils#tsv-filter.

- Print the lines where a specific column is numerically equal to a given number:

```
tsv-filter -H --eq {{field_name}}:{{number}} {{path/to/tsv_file}}
```

- Print the lines where a specific column is [eq]ual/[n]on [e]qual/[l]ess [t]han/[l]ess than or [e]qual/[g]reater [t]han/[g]reater than or [e]qual to a given number:

```
tsv-filter --{{eq|ne|lt|le|gt|ge}} {{column_number}}:{{number}} {{path/to/tsv_file}}
```

- Print the lines where a specific column is [eq]ual/[n]ot [e]qual/part of/not part of a given string:

```
tsv-filter --str-{{eq|ne|in-fld|not-in-fld}} {{column_number}}:{{string}} {{path/to/tsv_file}}
```

- Filter for non-empty fields:

```
tsv-filter --not-empty {{column_number}} {{path/to/tsv_file}}
```

- Print the lines where a specific column is empty:

```
tsv-filter --invert --not-empty {{column_number}} {{path/to/tsv_file}}
```

- Print the lines that satisfy two conditions:

```
tsv-filter --eq {{column_number1}}:{{number}} --str-eq {{column_number2}}:{{string}} {{path/to/tsv_file}}
```

- Print the lines that match at least one condition:

```
tsv-filter --or --eq {{column_number1}}:{{number}} --str-eq {{column_number2}}:{{string}} {{path/to/tsv_file}}
```

- Count matching lines, interpreting first line as a [H]eader:

```
tsv-filter --count -H --eq {{field_name}}:{{number}} {{path/to/tsv_file}}
```

tt

A terminal based typing test.

More information: https://github.com/lemnos/tt.

- Start quote mode with the builtin quote list in English:

```
tt -quotes {{en}}
```

- Produce a test consisting of 50 randomly drawn words in 5 groups of 10 words each:

```
tt -n {{10}} -g {{5}}
```

- Start a timed test lasting 10 seconds:

```
tt -t {{10}}
```

- Start `tt` with no theming and showing your WPM as you type:

```
tt -showwpm -notheme
```

tts

Synthesize speech.

More information: https://github.com/coqui-ai/TTS#command-line-tts.

- Run text-to-speech with the default models, writing the output to "tts_output.wav":

```
tts --text "{{text}}"
```

- List provided models:

```
tts --list_models
```

- Query info for a model by idx:

```
tts --model_info_by_idx {{model_type/model_query_idx}}
```

- Query info for a model by name:

```
tts --model_info_by_name {{model_type/language/dataset/
model_name}}
```

- Run a text-to-speech model with its default vocoder model:

```
tts --text "{{text}}" --model_name {{model_type/language/
dataset/model_name}}
```

- Run your own text-to-speech model (using the Griffin-Lim vocoder):

```
tts --text "{{text}}" --model_path {{path/to/model.pth}} --
config_path {{path/to/config.json}} --out_path {{path/to/
file.wav}}
```

tty

Returns terminal name.

More information: https://www.gnu.org/software/coreutils/manual/html_node/tty-invocation.html.

- Print the file name of this terminal:

```
tty
```

tuc

Cut text (or bytes) where a delimiter matches, then keep the desired parts.

A more user-friendly and powerful version of **`cut`** with sensible defaults.

More information: https://github.com/riquito/tuc.

- Cut and rearrange fields:

```
echo "foo bar baz" | tuc {{[-d|--delimiter]}} '{{ }}' {{[-f|--fields]}} {{3,2,1}}
```

- Replace the delimiter space with an arrow:

```
echo "foo bar baz" | tuc {{[-d|--delimiter]}} ' ' {{[-r|--replace-delimiter]}} ' ➡ '
```

- Keep a range of fields:

```
echo "foo bar baz" | tuc {{[-d|--delimiter]}} ' ' {{[-f|--fields]}} {{2:}}
```

- Cut using regex:

```
echo "a,b, c" | tuc {{[-e|--regex]}} '{{[, ]+}}' {{[-f|--fields]}} {{1,3}}
```

- Emit JSON output:

```
echo "foo bar baz" | tuc {{[-d|--delimiter]}} '{{ }}' --json
```

tuckr

Dotfile manager written in Rust.

See also: **`chezmoi`**, **`vcsh`**, **`homeshick`**, **`stow`**.

More information: https://github.com/RaphGL/Tuckr.

- Check dotfile status:

```
tuckr status
```

- Add all dotfiles to system:

```
tuckr add \*
```

- Add all dotfiles except specified programs:

```
tuckr add \* -e {{program1}},{{program2}}
```

- Remove all dotfiles from the system:

```
tuckr rm \*
```

- Add a program dotfile and run its setup script:

```
tuckr set {{program}}
```

tuir

A text user-interface (TUI) to view and interact with Reddit from your terminal.

Navigate with the Vim keys.

More information: https://gitlab.com/ajak/tuir.

- Launch tuir:

```
tuir
```

- Open a subreddit:

```
</>{{subreddit_name}}
```

- Open a link:

```
<o>
```

- Open a specific subreddit on launch:

```
tuir -s {{subreddit_name}}
```

- Open external links using programs defined in the mailcap config:

```
tuir --enable-media
```

turbo

High-performance build system for JavaScript and TypeScript codebases.

See also: **nx**.

More information: https://turborepo.org/docs/reference/command-line-reference.

- Log in using the default web browser with a Vercel account:

```
turbo login
```

- Link the current directory to a Vercel organization and enable remote caching:

```
turbo link
```

- Build the current project:

```
turbo run build
```

- Run a task without concurrency:

```
turbo run {{task_name}} --concurrency={{1}}
```

- Run a task ignoring cached artifacts and forcibly re-execute all tasks:

```
turbo run {{task_name}} --force
```

- Run a task in parallel across packages:

```
turbo run {{task_name}} --parallel --no-cache
```

- Unlink the current directory from your Vercel organization and disable Remote Caching:

```
turbo unlink
```

- Generate a Dot graph of a specific task execution (the output file format can be controlled with the filename):

```
turbo run {{task_name}} --graph={{path/to/file.html|jpg|json|pdf|png|svg}}
```

tv

Cross-platform, fast, extensible fuzzy-finder tool.

Works with "channels" (e.g., files, env, git repos) or from **`stdin`**.

More information: https://alexpasmantier.github.io/television/.

- Launch with the default channel:

```
tv
```

- Open a specific channel:

```
tv {{files|env|git-repos|...}}
```

- List all available channels:

```
tv list-channels
```

- Fuzzy-pick from lines piped on `stdin`:

```
{{command}} | tv
```

- Pipe and show a live preview for each entry:

```
{{command}} | tv --preview '{{preview_command}}'
```

- Build a picker from commands (source + preview):

```
tv --source-command '{{source_command}}' --preview-command
'{{preview_command}}' --preview-size {{70}}
```

- Update and install community-maintained channels:

```
tv update-channels
```

twine

Utility for publishing Python packages on PyPI.

More information: https://twine.readthedocs.io/en/stable/#commands.

- Upload to PyPI:

```
twine upload dist/*
```

- Upload to the Test PyPI repository to verify things look right:

```
twine upload {{[-r|--repository]}} testpypi dist/*
```

- Upload to PyPI with a specified username and password:

```
twine upload {{[-u|--username]}} {{username}} {{[-p|--
password]}} {{password}} dist/*
```

- Upload to an alternative repository URL:

```
twine upload --repository-url {{repository_url}} dist/*
```

- Check that your distribution's long description should render correctly on PyPI:

```
twine check dist/*
```

- Upload using a specific pypirc configuration file:

```
twine upload --config-file {{configuration_file}} dist/*
```

- Continue uploading files if one already exists (only valid when uploading to PyPI):

```
twine upload --skip-existing dist/*
```

- Upload to PyPI showing detailed information:

```
twine upload --verbose dist/*
```

twm

A window manager for the X Window system.

More information: https://manned.org/twm.

- Connect to the default X server:

```
twm
```

- Connect to a specific X server:

```
twm {{[-d|-display]}} {{display}}
```

- Only manage the default screen:

```
twm {{[-s|-single]}}
```

- Use a specific startup file:

```
twm {{[-f|-file]}} {{path/to/file}}
```

- Enable verbose mode and print unexpected errors in X:

```
twm {{[-v|-verbose]}}
```

twopi

Render an image of a **radial** network graph from a **graphviz** file.

Layouts: **dot**, **neato**, **twopi**, **circo**, **fdp**, **sfdp**, **osage** & **patchwork**.

More information: https://graphviz.org/doc/info/command.html.

- Render a PNG image with a filename based on the input filename and output format (uppercase -O):

```
twopi -T {{png}} -O {{path/to/input.gv}}
```

- Render a SVG image with the specified output filename (lowercase -o):

```
twopi -T {{svg}} -o {{path/to/image.svg}} {{path/to/input.gv}}
```

- Render the output in PS, PDF, SVG, Fig, PNG, GIF, JPEG, JSON, or DOT format:

```
twopi -T {{format}} -O {{path/to/input.gv}}
```

- Render a GIF image using `stdin` and `stdout`:

```
echo "{{digraph {this -> that} }}" | twopi -T {{gif}} > {{path/to/image.gif}}
```

- Display help:

```
twopi -?
```

twurl

Curl-like command but tailored specifically for the Twitter API.

More information: https://github.com/twitter/twurl.

- Authorize `twurl` to access a Twitter account:

```
twurl authorize {{[-c|--consumer-key]}} {{twitter_api_key}}
{{[-s|--consumer-secret]}} {{twitter_api_secret}}
```

- Make a GET request to an API endpoint:

```
twurl {{[-X|--request-method]}} GET {{twitter_api_endpoint}}
```

- Make a POST request to an API endpoint:

```
twurl {{[-X|--request-method]}} POST {{[-d|--data]}}
'{{endpoint_params}}' {{twitter_api_endpoint}}
```

- Upload media to Twitter:

```
twurl {{[-H|--host]}} "{{twitter_upload_url}}" {{[-X|--
request-method]}} POST "{{twitter_upload_endpoint}}" {{[-f|--
file]}} "{{path/to/media.jpg}}" {{[-F|--file-field]}} "media"
```

- Access a different Twitter API host:

```
twurl {{[-H|--host]}} {{twitter_api_url}} {{[-X|--request-
method]}} GET {{twitter_api_endpoint}}
```

- Create an alias for a requested resource:

```
twurl alias {{alias_name}} {{resource}}
```

tye

Develop, test, and deploy microservices and distributed applications easily.

More information: https://github.com/dotnet/tye.

- Scaffold a `tye.yaml` file representing the application:

  ```
  tye init
  ```

- Run an application locally:

  ```
  tye run
  ```

- Build an application's containers:

  ```
  tye build
  ```

- Push an application's containers:

  ```
  tye push
  ```

- Deploy an application to Kubernetes:

  ```
  tye deploy
  ```

- Remove a deployed application from Kubernetes:

  ```
  tye undeploy
  ```

type

Display the type of command the shell will execute.

Note: All examples are not POSIX compliant.

See also: **`whereis`**, **`which`**, **`whatis`**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-type.

- Display the type of a command:

```
type {{command}}
```

- Display all locations containing the specified executable (works only in Bash/fish/Zsh shells):

```
type -a {{command}}
```

- Display the name of the disk file that would be executed (works only in Bash/fish/Zsh shells):

```
type -p {{command}}
```

- Display the type of a specific command, alias/keyword/function/builtin/file (works only in Bash/fish shells):

```
type -t {{command}}
```

typeinc

An **`ncurses`** based terminal typing speed test program, written in Python.

Try out different difficulty levels and improve your typing speed.

More information: https://github.com/AnirudhG07/Typeinc.

- Enter the typing test:

  `typeinc`

- Display the top 10 rank list for input difficulty level:

  `typeinc {{[-r|--ranklist]}} {{difficulty_level}}`

- Get random English words present in our wordlist:

  `typeinc {{[-w|--words]}} {{word_count}}`

- Calculate hypothetical Typeinc score:

  `typeinc {{[-s|--score]}}`

typeorm

A JavaScript ORM that can run on Node.js, browser, Cordova, Ionic, React Native, NativeScript, and Electron platforms.

More information: https://typeorm.io/docs/advanced-topics/using-cli/#initialize-a-new-typeorm-project.

- Generate a new initial TypeORM project structure:

  ```
  typeorm init
  ```

- Create an empty migration file:

  ```
  typeorm migration:create --name {{migration_name}}
  ```

- Create a migration file with the SQL statements to update the schema:

  ```
  typeorm migration:generate --name {{migration_name}}
  ```

- Run all pending migrations:

  ```
  typeorm migration:run
  ```

- Create a new entity file in a specific directory:

  ```
  typeorm entity:create --name {{entity}} --dir {{path/to/directory}}
  ```

- Display the SQL statements to be executed by `typeorm schema:sync` on the default connection:

  ```
  typeorm schema:log
  ```

- Execute a specific SQL statement on the default connection:

  ```
  typeorm query {{sql_sentence}}
  ```

- Display help for a subcommand:

  ```
  typeorm {{subcommand}} --help
  ```

typeset

This command is an alias of **`declare`**.

- View documentation for the original command:

```
tldr declare
```

typst

Compile a Typst file to PDF.

Note: Specifying the output location is optional.

More information: https://github.com/typst/typst.

- Initialize a new Typst project in a given directory using a template (e.g., `@preview/charged-ieee`):

  ```
  typst init "{{template}}" {{path/to/directory}}
  ```

- Compile a Typst file:

  ```
  typst compile {{path/to/source.typ}} {{path/to/output.pdf}}
  ```

- Watch a Typst file and recompile on changes:

  ```
  typst watch {{path/to/source.typ}} {{path/to/output.pdf}}
  ```

- List all discoverable fonts in the system and the given directory:

  ```
  typst --font-path {{path/to/fonts_directory}} fonts
  ```

tzconfig

Manage and set your local timezone.

It accepts no parameters.

More information: https://manned.org/tzconfig.

- Launch timezone configuration utility:

```
tzconfig
```

u3d

A set of tools to interact with Unity.

More information: https://github.com/DragonBox/u3d.

- Open the project in the current directory using the correct Unity version:

  ```
  u3d
  ```

- List installed versions of Unity:

  ```
  u3d list
  ```

- List available versions of Unity that can be downloaded:

  ```
  u3d available
  ```

- Download and install latest stable Unity version:

  ```
  u3d install latest_stable
  ```

- Download and install Unity version and editor [p]ackages:

  ```
  u3d install {{2021.2.0f1}} -p {{Unity,iOS,Android}}
  ```

ufraw-batch

Convert RAW files from cameras into standard image files.

More information: https://manned.org/ufraw-batch.

- Simply convert RAW files to JPEG:

```
ufraw-batch --out-type=jpg {{input_file(s)}}
```

- Simply convert RAW files to PNG:

```
ufraw-batch --out-type=png {{input_file(s)}}
```

- Extract the preview image from the raw file:

```
ufraw-batch --embedded-image {{input_file(s)}}
```

- Save the file with size up to the given maximums MAX1 and MAX2:

```
ufraw-batch --size=MAX1,MAX2 {{input_file(s)}}
```

ugrep

Ultra fast search tool with query TUI.

More information: https://github.com/Genivia/ugrep.

- Start a query TUI to search files in the current directory recursively (`<Ctrl z>` for help):

```
ugrep {{[-Q|--query]}}
```

- Search the current directory recursively for files containing a `regex` search pattern:

```
ugrep "{{search_pattern}}"
```

- Search in a specific file or in all files in a specific directory, showing line numbers of matches:

```
ugrep {{[-n|--line-number]}} "{{search_pattern}}" {{path/to/file_or_directory}}
```

- Search in all files in the current directory recursively and print the name of each matching file:

```
ugrep {{[-l|--files-with-matches]}} "{{search_pattern}}"
```

- Fuzzy search files with up to 3 extra, missing or mismatching characters in the pattern:

```
ugrep {{[-Z|--fuzzy=]}}{{3}} "{{search_pattern}}"
```

- Also search compressed files, Zip and tar archives recursively:

```
ugrep {{[-z|--decompress]}} "{{search_pattern}}"
```

- Search only files whose filenames match a specific glob pattern:

```
ugrep {{[-g |--glob=]}}"{{glob_pattern}}" "{{search_pattern}}"
```

- Search only C++ source files (use `--file-type=list` to list all file types):

```
ugrep {{[-t |--file-type=]}}cpp "{{search_pattern}}"
```

ulimit

Get and set resource limits for user processes.

It is a shell builtin hence not shell-agnostic.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-ulimit.

- Get the properties of all the user limits:

```
ulimit -a
```

- Get hard limit for the number of simultaneously opened files:

```
ulimit -H -n
```

- Get soft limit for the number of simultaneously opened files:

```
ulimit -S -n
```

- Set max per-user process limit:

```
ulimit -u {{30}}
```

- Display help (Bash only):

```
help ulimit
```

umask

Manage the read/write/execute permissions that are masked out (i.e. restricted) for newly created files by the user.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-umask.

- Display the current mask in octal notation:

```
umask
```

- Display the current mask in symbolic (human-readable) mode:

```
umask -S
```

- Change the mask symbolically to allow read permission for all users (the rest of the mask bits are unchanged):

```
umask {{a+r}}
```

- Set the mask (using octal) to restrict no permissions for the file's owner, and restrict all permissions for everyone else:

```
umask {{077}}
```

umount

Unlink a filesystem from its mount point, making it no longer accessible.

A filesystem cannot be unmounted when it is busy.

More information: https://man.openbsd.org/umount.

- Unmount a filesystem, by passing the path to the source it is mounted from:

```
umount {{path/to/device_file}}
```

- Unmount a filesystem, by passing the path to the target where it is mounted:

```
umount {{path/to/mounted_directory}}
```

- Unmount all mounted filesystems (except the `proc` filesystem):

```
umount -a
```

unalias

Remove aliases.

See also: **alias**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-unalias.

- Remove an alias:

```
unalias {{alias_name}}
```

- Remove all aliases:

```
unalias -a
```

uname

Print details about the current machine and the operating system running on it.

See also: **`lsb_release`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/uname-invocation.html.

- Print kernel name:

```
uname
```

- Print all available system information:

```
uname {{[-a|--all]}}
```

- Print system architecture and processor information:

```
uname {{[-mp|--machine --processor]}}
```

- Print kernel name, kernel release and kernel version:

```
uname {{[-srv|--kernel-name --kernel-release --kernel-version]}}
```

- Print system hostname:

```
uname {{[-n|--nodename]}}
```

- Print the current operating system name:

```
uname {{[-o|--operating-system]}}
```

- Display help:

```
uname --help
```

unar

Extract contents from archive files.

More information: https://manned.org/unar.

- Extract an archive to the current directory:

```
unar {{path/to/archive}}
```

- Extract an archive to the specified directory:

```
unar {{[-o|-output-directory]}} {{path/to/directory}} {{path/to/archive}}
```

- Force overwrite if files to be unpacked already exist:

```
unar {{[-f|-force-overwrite]}} {{path/to/archive}}
```

- Force rename if files to be unpacked already exist:

```
unar {{[-r|-force-rename]}} {{path/to/archive}}
```

- Force skip if files to be unpacked already exist:

```
unar {{[-s|-force-skip]}} {{path/to/archive}}
```

unclutter

Hides the mouse cursor.

More information: https://manned.org/unclutter.1x.

- Hide mouse cursor after 3 seconds:

```
unclutter -idle {{3}}
```

uncrustify

C, C++, C#, D, Java and Pawn source code formatter.

More information: https://github.com/uncrustify/uncrustify.

- Format a single file:

```
uncrustify -f {{path/to/file.cpp}} -o {{path/to/output.cpp}}
```

- Read filenames from `stdin`, and take backups before writing output back to the original filepaths:

```
find . -name "*.cpp" | uncrustify -F - --replace
```

- Don't make backups (useful if files are under version control):

```
find . -name "*.cpp" | uncrustify -F - --no-backup
```

- Use a custom configuration file and write the result to `stdout`:

```
uncrustify -c {{path/to/uncrustify.cfg}} -f {{path/to/
file.cpp}}
```

- Explicitly set a configuration variable's value:

```
uncrustify --set {{option}}={{value}}
```

- Generate a new configuration file:

```
uncrustify --update-config -o {{path/to/new.cfg}}
```

unexpand

Convert spaces to tabs.

More information: https://www.gnu.org/software/coreutils/manual/html_node/unexpand-invocation.html.

- Convert blanks in each file to tabs, writing to `stdout`:

```
unexpand {{path/to/file}}
```

- Convert blanks to tabs, reading from `stdout`:

```
unexpand
```

- Convert all blanks, instead of just initial blanks:

```
unexpand {{[-a|--all]}} {{path/to/file}}
```

- Convert only leading sequences of blanks (overrides -a):

```
unexpand --first-only {{path/to/file}}
```

- Have tabs a certain number of characters apart, not 8 (enables -a):

```
unexpand {{[-t|--tabs]}} {{number}} {{path/to/file}}
```

unflatten

Adjust directed graphs to improve the layout aspect ratio.

Graphviz filters: **`acyclic`**, **`bcomps`**, **`comps`**, **`edgepaint`**, **`gvcolor`**, **`gvpack`**, **`mingle`**, **`nop`**, **`sccmap`**, **`tred`**, & **`unflatten`**.

More information: https://www.graphviz.org/pdf/unflatten.1.pdf.

- Adjust one or more directed graphs to improve the layout aspect ratio:

```
unflatten {{path/to/input1.gv path/to/input2.gv ...}} > {{path/to/output.gv}}
```

- Use `unflatten` as a preprocessor for `dot` layout to improve aspect ratio:

```
unflatten {{path/to/input.gv}} | dot -T {{png}} {{path/to/output.png}}
```

- Display help:

```
unflatten -?
```

unimatrix

Simulate the Matrix look with Unicode characters.

See also: **`cmatrix`**.

More information: https://github.com/will8211/unimatrix.

- Mimic the default output of `cmatrix` (no unicode, works in a TTY):

```
unimatrix --no-bold --speed {{96}} --character-list {{o}}
```

- No bold characters, slowly, with emojis, numbers, and a few symbols:

```
unimatrix --no-bold --speed {{50}} --character-list {{ens}}
```

- Change the color of characters:

```
unimatrix --color {{red|green|blue|white|...}}
```

- Select character set(s) using letter codes (see `unimatrix --help` for available character sets):

```
unimatrix --character-list {{character_sets}}
```

- Change the scrolling speed:

```
unimatrix --speed {{number}}
```

uniq

Output the unique lines from a input or file.

Since it does not detect repeated lines unless they are adjacent, we need to sort them first.

More information: https://www.gnu.org/software/coreutils/manual/html_node/uniq-invocation.html.

- Display each line once:

```
sort {{path/to/file}} | uniq
```

- Display only unique lines:

```
sort {{path/to/file}} | uniq {{[-u|--unique]}}
```

- Display only duplicate lines:

```
sort {{path/to/file}} | uniq {{[-d|--repeated]}}
```

- Display number of occurrences of each line along with that line:

```
sort {{path/to/file}} | uniq {{[-c|--count]}}
```

- Display number of occurrences of each line, sorted by the most frequent:

```
sort {{path/to/file}} | uniq {{[-c|--count]}} | sort {{[-nr|--numeric-sort --reverse]}}
```

unison

Bidirectional file synchronisation tool.

More information: https://github.com/bcpierce00/unison.

- Sync two directories (creates log first time these two directories are synchronized):

```
unison {{path/to/directory_1}} {{path/to/directory_2}}
```

- Automatically accept the (non-conflicting) defaults:

```
unison {{path/to/directory_1}} {{path/to/directory_2}} -auto
```

- Ignore some files using a pattern:

```
unison {{path/to/directory_1}} {{path/to/directory_2}} -
ignore {{pattern}}
```

- View documentation:

```
unison -doc {{topics}}
```

units

Convert between two units of measure.

More information: https://www.gnu.org/software/units/manual/units.html.

- Run in interactive mode:

```
units
```

- List all units containing a specific string in interactive mode:

```
search {{string}}
```

- Show the conversion between two simple units:

```
units {{quarts}} {{tablespoons}}
```

- Convert between units with quantities:

```
units "{{15 pounds}}" {{kilograms}}
```

- Show the conversion between two compound units:

```
units "{{meters / second}}" "{{inches / hour}}"
```

- Show the conversion between units with different dimensions:

```
units "{{acres}}" "{{ft^2}}"
```

- Show the conversion of byte multipliers:

```
units "{{15 megabytes}}" {{bytes}}
```

unlink

Remove a link to a file from the filesystem.

The file contents is lost if the link is the last one to the file.

More information: https://www.gnu.org/software/coreutils/manual/html_node/unlink-invocation.html.

- Remove the specified file if it is the last link:

```
unlink {{path/to/file}}
```

unlzma

This command is an alias of **`xz --format lzma --decompress`**.

- View documentation for the original command:

```
tldr xz
```

unp

Extract any archive.

Relevant extractors need to be installed, e.g. **`unrar`** for RAR.

More information: https://manned.org/unp.

- Extract an archive:

```
unp {{path/to/archive.zip}}
```

- Extract multiple archives:

```
unp {{path/to/archive1.tar.gz}} {{path/to/archive2.rar}}
```

unrar

Extract RAR archives.

More information: https://manned.org/unrar.

- Extract files with original directory structure:

```
unrar x {{compressed.rar}}
```

- Extract files to a specified path with the original directory structure:

```
unrar x {{compressed.rar}} {{path/to/extract}}
```

- Extract files into current directory, losing directory structure in the archive:

```
unrar e {{compressed.rar}}
```

- Test integrity of each file inside the archive file:

```
unrar t {{compressed.rar}}
```

- List files inside the archive file without decompressing it:

```
unrar l {{compressed.rar}}
```

unset

Remove shell variables or functions.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-unset.

- Remove the variable `foo`, or if the variable doesn't exist, remove the function `foo`:

```
unset {{foo}}
```

- Remove the variables foo and bar:

```
unset -v {{foo}} {{bar}}
```

- Remove the function my_func:

```
unset -f {{my_func}}
```

until

Simple shell loop that repeats until it receives zero as return value.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-until.

- Execute a command until it succeeds:

```
until {{command}}; do :; done
```

- Wait for a systemd service to be active:

```
until systemctl is-active {{[-q|--quiet]}} {{nginx}}; do
{{echo "Waiting..."}}; sleep 1; done; {{echo "Launched!"}}
```

unxz

This command is an alias of **`xz --decompress`**.

- View documentation for the original command:

```
tldr xz
```

unzip

Extract files/directories from Zip archives.

See also: **`zip`**.

More information: https://manned.org/unzip.

- Extract all files/directories from specific archives into the current directory:

```
unzip {{path/to/archive1.zip path/to/archive2.zip ...}}
```

- Extract files/directories from archives to a specific path:

```
unzip {{path/to/archive1.zip path/to/archive2.zip ...}} -d
{{path/to/output}}
```

- Extract files/directories from archives to `stdout` alongside the extracted file names:

```
unzip -c {{path/to/archive1.zip path/to/archive2.zip ...}}
```

- Extract an archive created on Windows, containing files with non-ASCII (e.g. Chinese or Japanese characters) filenames:

```
unzip -O {{gbk}} {{path/to/archive1.zip path/to/
archive2.zip ...}}
```

- List the contents of a specific archive without extracting them:

```
unzip -l {{path/to/archive.zip}}
```

- Extract a specific file from an archive:

```
unzip -j {{path/to/archive.zip}} {{path/to/file1_in_archive
path/to/file2_in_archive ...}}
```

unzstd

This command is an alias of **`zstd --decompress`**.

- View documentation for the original command:

  `tldr zstd`

updog

A replacement for Python's SimpleHTTPServer.

It allows uploading and downloading via HTTP/S, can set ad hoc SSL certificates and use HTTP basic auth.

More information: https://github.com/sc0tfree/updog.

- Start a HTTP server for the current directory:

```
updog
```

- Start a HTTP server for a specified directory:

```
updog --directory /{{path/to/directory}}
```

- Start a HTTP server on a specified port:

```
updog --port {{port}}
```

- Start a HTTP server with a password (To log in, leave the username blank and enter the password in the password field):

```
updog --password {{password}}
```

- Enable transport encryption via SSL:

```
updog --ssl
```

upt

Unified interface for managing packages across various operating systems, like Windows, many Linux distributions, macOS, FreeBSD and even Haiku.

It requires the native OS package manager to be installed.

See also: **`flatpak`**, **`brew`**, **`scoop`**, **`apt`**, **`dnf`**.

More information: https://github.com/sigoden/upt.

- Update the list of available packages:

```
upt update
```

- Search for a given package:

```
upt search {{search_term}}
```

- Show information for a package:

```
upt info {{package}}
```

- Install a given package:

```
upt install {{package}}
```

- Remove a given package:

```
upt {{remove|uninstall}} {{package}}
```

- Upgrade all installed packages:

```
upt upgrade
```

- Upgrade a given package:

```
upt upgrade {{package}}
```

- List installed packages:

```
upt list
```

uptime

Tell how long the system has been running and other information.

More information: https://www.gnu.org/software/coreutils/manual/html_node/uptime-invocation.html.

- Print current time, uptime, number of logged-in users and other information:

```
uptime
```

- Show only the amount of time the system has been booted for:

```
uptime {{[-p|--pretty]}}
```

- Print the date and time the system booted up at:

```
uptime {{[-s|--since]}}
```

- Display version:

```
uptime {{[-V|--version]}}
```

upx

Compress or decompress executables.

More information: https://upx.github.io.

- Compress executable:

```
upx {{path/to/file}}
```

- Decompress executable:

```
upx -d {{path/to/file}}
```

- Display help:

```
upx --help
```

users

Display a list of logged in users.

See also: **`useradd`**, **`userdel`**, **`usermod`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/users-invocation.html.

- Print logged in usernames:

```
users
```

- Print logged in usernames according to a given file:

```
users {{/var/log/wmtp}}
```

usleep

Delay execution for a specific interval in microseconds.

Largely deprecated in favor of **`nanosleep`**.

See also: **`sleep`**, **`nanosleep`**.

More information: https://manned.org/usleep.1.

- Delay in microseconds:

  ```
  usleep {{microseconds}}
  ```

- Execute a specific command after a 500,000 microseconds delay:

  ```
  usleep 500000 && {{command}}
  ```

usql

Universal CLI interface for SQL databases.

More information: https://github.com/xo/usql.

- Connect to a specific database:

```
usql {{sqlserver|mysql|postgres|sqlite3|...}}://{{username}}:{{password}}@{{host}}:{{port}}/{{database_name}}
```

- Execute commands from a file:

```
usql --file={{path/to/query.sql}}
```

- Execute a specific SQL command:

```
usql --command="{{sql_command}}"
```

- Run an SQL command in the `usql` prompt:

```
{{prompt}}=> {{command}}
```

- Display the database schema:

```
{{prompt}}=> \d
```

- Export query results to a specific file:

```
{{prompt}}=> \g {{path/to/file_with_results}}
```

- Import data from a CSV file into a specific table:

```
{{prompt}}=> \copy {{path/to/data.csv}} {{table_name}}
```

uudecode

Decode files encoded by **uuencode**.

More information: https://manned.org/uudecode.

- Decode a file that was encoded with `uuencode` and print the result to `stdout`:

  `uudecode {{path/to/encoded_file}}`

- Decode a file that was encoded with `uuencode` and write the result to a file:

  `uudecode -o {{path/to/decoded_file}} {{path/to/encoded_file}}`

uuencode

Encode binary files into ASCII for transport via mediums that only support simple ASCII encoding.

More information: https://manned.org/uuencode.

- Encode a file and print the result to `stdout`:

```
uuencode {{path/to/input_file}}
{{output_file_name_after_decoding}}
```

- Encode a file and write the result to a file:

```
uuencode -o {{path/to/output_file}} {{path/to/input_file}}
{{output_file_name_after_decoding}}
```

- Encode a file using Base64 instead of the default uuencode encoding and write the result to a file:

```
uuencode {{[-m|--base64]}} -o {{path/to/output_file}} {{path/
to/input_file}} {{output_file_name_after_decoding}}
```

uv add

Add package dependencies to the **`pyproject.toml`** file.

Packages are specified according to https://peps.python.org/pep-0508/.

More information: https://docs.astral.sh/uv/reference/cli/#uv-add.

- Add the latest version of a package:

  ```
  uv add {{package}}
  ```

- Add multiple packages:

  ```
  uv add {{package1 package2 ...}}
  ```

- Add a package with a version requirement:

  ```
  uv add {{package>=1.2.3}}
  ```

- Add packages to an optional dependency group, which will be included when published:

  ```
  uv add --optional {{optional}} {{package1 package2 ...}}
  ```

- Add packages to a local group, which will not be included when published:

  ```
  uv add --group {{group}} {{package1 package2 ...}}
  ```

- Add packages to the dev group, shorthand for `--group dev`:

  ```
  uv add --dev {{package1 package2 ...}}
  ```

- Add package as editable:

  ```
  uv add --editable {{path/to/package}}/
  ```

- Enable an extra when installing package, may be provided multiple times:

  ```
  uv add {{package}} --extra {{extra_feature}}
  ```

uv build

Build Python packages into source distributions and wheels.

More information: https://docs.astral.sh/uv/reference/cli/#uv-build.

- Build a package in the current directory:

```
uv build
```

- Build a package from a specific directory:

```
uv build {{path/to/directory}}
```

- Build only a wheel (skip source distribution):

```
uv build --wheel
```

- Build only a source distribution (skip wheel):

```
uv build --sdist
```

- Build and output to a specific directory:

```
uv build {{[-o|--out-dir]}} {{path/to/output}}
```

- Build a specific package in a workspace:

```
uv build --package {{package_name}}
```

- Build all packages in the workspace:

```
uv build {{[--all|--all-packages]}}
```

- Build with a specific Python interpreter:

```
uv build {{[-p|--python]}} {{python3.11}}
```

uv cache

Manage **uv**'s global cache directory.

More information: https://docs.astral.sh/uv/reference/cli/#uv-cache.

- Show the cache directory path:

```
uv cache dir
```

- Clean the entire cache (removes all cached packages and environments):

```
uv cache clean
```

- Clean the cache for specific packages:

```
uv cache clean {{package1 package2 ...}}
```

- Prune all unreachable objects from the cache:

```
uv cache prune
```

- Prune cache optimized for CI environments like GitHub Actions:

```
uv cache prune --ci
```

- Use a specific cache directory:

```
uv cache clean --cache-dir {{path/to/cache}}
```

- Clean cache with verbose output:

```
uv cache clean {{[-v|--verbose]}}
```

uv export

Export the project's lockfile to an alternate format.

More information: https://docs.astral.sh/uv/reference/cli/#uv-export.

- Export dependencies to a `requirements.txt` file:

```
uv export --format requirements-txt {{[-o|--output-file]}} {{requirements.txt}}
```

- Export dependencies to `pylock.toml` format:

```
uv export --format pylock.toml
```

- Export only production dependencies (exclude dev dependencies):

```
uv export --no-dev
```

- Export including a specific optional dependency group:

```
uv export --extra {{group_name}}
```

- Export including all optional dependencies:

```
uv export --all-extras
```

- Export including a specific dependency group:

```
uv export --group {{group_name}}
```

- Export without hashes:

```
uv export --no-hashes
```

- Export dependencies for a specific package in the workspace:

```
uv export --package {{package_name}}
```

uv help

Display detailed documentation for **uv** commands.

More information: https://docs.astral.sh/uv/reference/cli/#uv-help.

- Display general help for uv:

```
uv help
```

- Display help for a specific command:

```
uv help {{command}}
```

- Display help for a subcommand:

```
uv help {{command}} {{subcommand}}
```

- Display help without using a pager:

```
uv help --no-pager {{command}}
```

uv init

Create a new Python project.

More information: https://docs.astral.sh/uv/reference/cli/#uv-init.

- Initialize a project in the current directory:

```
uv init
```

- Initialize a project with a certain name:

```
uv init {{project_name}}
```

- Create a project in a given directory:

```
uv init --directory {{path/to/directory}} {{project_name}}
```

- Create a project for a Python library:

```
uv init {{[--lib|--library]}} {{project_name}}
```

- Specify the build system:

```
uv init --build-backend {{build_backend}} {{project_name}}
```

- Only create a `pyproject.toml`:

```
uv init --bare {{project_name}}
```

- Set the project description:

```
uv init --description "{{description}}" {{project_name}}
```

uv lock

Update the project's lockfile.

More information: https://docs.astral.sh/uv/reference/cli/#uv-lock.

- Create or update the project's lockfile:

```
uv lock
```

- Check if the lockfile is up-to-date without updating it:

```
uv lock --check
```

- Assert that a lockfile exists without checking if it's current:

```
uv lock --check-exists
```

- Preview what would be locked without writing the lockfile:

```
uv lock --dry-run
```

- Lock a specific Python script instead of the current project:

```
uv lock --script {{path/to/script.py}}
```

- Upgrade all packages to their latest compatible versions:

```
uv lock --upgrade
```

- Upgrade only specific packages:

```
uv lock --upgrade-package {{package1}} --upgrade-package
{{package2}}
```

uv pip

Provides pip-like commands for installing, uninstalling, and managing packages.

More information: https://docs.astral.sh/uv/reference/cli/#uv-pip.

- Install a package:

```
uv pip install {{package}}
```

- Install packages from a requirements file:

```
uv pip install {{[-r|--requirements]}} {{requirements.txt}}
```

- Install a package with a specific version:

```
uv pip install {{package==1.2.3}}
```

- Uninstall a package:

```
uv pip uninstall {{package}}
```

- Save installed packages to file:

```
uv pip freeze > {{requirements.txt}}
```

- List installed packages:

```
uv pip list
```

- Show information about an installed package:

```
uv pip show {{package}}
```

- Sync environment with a requirements file (install/uninstall to match exactly):

```
uv pip sync {{requirements.txt}}
```

uv publish

Upload distributions to an index.

More information: https://docs.astral.sh/uv/reference/cli/#uv-publish.

- Publish packages from `dist/` directory (default behavior):

```
uv publish
```

- Publish to a specific repository URL:

```
uv publish --publish-url {{https://upload.pypi.org/legacy/}}
```

- Publish using a specific username and password:

```
uv publish {{[-u|--username]}} {{username}} {{[-p|--
password]}} {{password}}
```

- Publish using an API token:

```
uv publish {{[-t|--token]}} {{your_api_token}}
```

- Publish specific distribution files:

```
uv publish {{path/to/dist/*.whl}} {{path/to/dist/*.tar.gz}}
```

- Publish to TestPyPI for testing:

```
uv publish --publish-url https://test.pypi.org/legacy/
```

uv python

Manage Python versions and installations.

More information: https://docs.astral.sh/uv/reference/cli/#uv-python.

- List all available Python installations:

```
uv python list
```

- Install a Python version:

```
uv python install {{version}}
```

- Uninstall a Python version:

```
uv python uninstall {{version}}
```

- Search for a Python installation:

```
uv python find {{version}}
```

- Pin the current project to use a specific Python version:

```
uv python pin {{version}}
```

- Show the uv Python installation directory:

```
uv python dir
```

uv remove

Remove dependencies from the project's **`pyproject.toml`** file.

More information: https://docs.astral.sh/uv/reference/cli/#uv-remove.

- Remove a dependency from the project:

  ```
  uv remove {{package}}
  ```

- Remove multiple dependencies:

  ```
  uv remove {{package1 package2 ...}}
  ```

- Remove a development dependency:

  ```
  uv remove --dev {{package}}
  ```

- Remove a dependency from an optional dependency group:

  ```
  uv remove --optional {{extra_name}} {{package}}
  ```

- Remove a dependency from a specific dependency group:

  ```
  uv remove --group {{group_name}} {{package}}
  ```

- Remove without syncing the virtual environment:

  ```
  uv remove --no-sync {{package}}
  ```

uv run

Run a command or script in the project environment.

More information: https://docs.astral.sh/uv/reference/cli/#uv-run.

- Run a Python script:

```
uv run {{path/to/script.py}}
```

- Run a Python module:

```
uv run {{[-m|--module]}} {{module_name}}
```

- Run a command with additional packages installed temporarily:

```
uv run {{[-w|--with]}} {{package}} {{command}}
```

- Run a script with packages from a requirements file:

```
uv run --with-requirements {{path/to/requirements.txt}}
{{path/to/script.py}}
```

- Run in an isolated environment (no project dependencies):

```
uv run --isolated {{path/to/script.py}}
```

- Run without syncing the environment first:

```
uv run --no-sync {{command}}
```

uv self

Manage the **uv** executable itself.

More information: https://docs.astral.sh/uv/reference/cli/#uv-self.

- Update uv to the latest version:

```
uv self update
```

- Update uv to a specific version:

```
uv self update {{0.4.0}}
```

- Check for available uv updates without installing:

```
uv self update --dry-run
```

- Update uv with verbose output:

```
uv self update {{[-v|--verbose]}}
```

- Display the current uv version:

```
uv self version
```

- Display only the version number:

```
uv self version --short
```

- Display version information in JSON format:

```
uv self version --output-format json
```

uv sync

Update the project's environment to match the lockfile.

More information: https://docs.astral.sh/uv/reference/cli/#uv-sync.

- Sync the project environment with the lockfile:

  ```
  uv sync
  ```

- Sync and include all optional dependencies:

  ```
  uv sync --all-extras
  ```

- Sync with specific optional dependencies:

  ```
  uv sync --extra {{extra_name}}
  ```

- Sync only development dependencies:

  ```
  uv sync --only-dev
  ```

- Sync excluding development dependencies:

  ```
  uv sync --no-dev
  ```

- Sync specific dependency groups:

  ```
  uv sync --group {{group_name}}
  ```

- Check if environment is already synchronized (no changes):

  ```
  uv sync --check
  ```

- Preview what would be synced without making changes:

  ```
  uv sync --dry-run
  ```

uv tool

Install and run commands provided by Python packages.

More information: https://docs.astral.sh/uv/reference/cli/#uv-tool.

- Run a command from a package, without installing it:

```
uv tool run {{command}}
```

- Install a Python package system-wide:

```
uv tool install {{package}}
```

- Upgrade an installed Python package:

```
uv tool upgrade {{package}}
```

- Uninstall a Python package:

```
uv tool uninstall {{package}}
```

- List Python packages installed system-wide:

```
uv tool list
```

uv tree

Display project dependencies in a tree format.

More information: https://docs.astral.sh/uv/reference/cli/#uv-tree.

- Show dependency tree for current environment:

```
uv tree
```

- Show dependency tree for all environments:

```
uv tree --universal
```

- Show dependency tree up to a certain depth:

```
uv tree {{[-d|--depth]}} {{n}}
```

- Show the latest available version for all outdated packages:

```
uv tree --outdated
```

- Exclude dependencies from the dev group:

```
uv tree --no-dev
```

- Show the inverted tree, so children are dependents instead of dependencies:

```
uv tree --invert
```

uv venv

Create an isolated Python environment for installing packages.

More information: https://docs.astral.sh/uv/reference/cli/#uv-venv.

- Create a virtual environment in the default location (`.venv`):

  ```
  uv venv
  ```

- Create a virtual environment at a specific path:

  ```
  uv venv {{path/to/venv}}
  ```

- Create using a specific Python version:

  ```
  uv venv --python {{3.12}}
  ```

- Create with a custom prompt prefix:

  ```
  uv venv --prompt {{my_project}}
  ```

- Create and allow overwriting existing environment:

  ```
  uv venv --allow-existing {{venv_name}}
  ```

uv version

Read or update a project's version.

More information: https://docs.astral.sh/uv/reference/cli/#uv-version.

- Display the current project version:

```
uv version
```

- Set the project version to a specific value:

```
uv version {{1.2.3}}
```

- Bump the project version using semantic versioning:

```
uv version --bump {{major|minor|patch}}
```

- Preview version changes without writing to `pyproject.toml`:

```
uv version --bump {{patch}} --dry-run
```

- Update version for a specific package in a workspace:

```
uv version --package {{package_name}} {{1.2.3}}
```

- Display version in JSON format:

```
uv version --output-format json
```

uv

A fast Python package and project manager.

Some subcommands such as **`tool`** and **`python`** have their own usage documentation.

More information: https://docs.astral.sh/uv/reference/cli.

- Create a new Python project in the current directory:

  ```
  uv init
  ```

- Create a new Python project at the specified path:

  ```
  uv init {{path/to/directory}}
  ```

- Add a new dependency to the project:

  ```
  uv add {{package}}
  ```

- Remove a dependency from the project:

  ```
  uv remove {{package}}
  ```

- Run a script in the project's environment:

  ```
  uv run {{path/to/script.py}}
  ```

- Run a command in the project's environment:

  ```
  uv run {{command}}
  ```

- Update a project's environment from `pyproject.toml`:

  ```
  uv sync
  ```

- Create a lock file for the project's dependencies:

  ```
  uv lock
  ```

uvicorn

Python ASGI HTTP Server, for asynchronous projects.

More information: https://www.uvicorn.org/settings/.

- Run Python web app:

```
uvicorn {{import.path:app_object}}
```

- Listen on port 8080 on localhost:

```
uvicorn --host {{localhost}} --port {{8080}}
{{import.path:app_object}}
```

- Turn on live reload:

```
uvicorn --reload {{import.path:app_object}}
```

- Use 4 worker processes for handling requests:

```
uvicorn --workers {{4}} {{import.path:app_object}}
```

- Run app over HTTPS:

```
uvicorn --ssl-certfile {{cert.pem}} --ssl-keyfile {{key.pem}}
{{import.path:app_object}}
```

V

A tool for managing V source code.

More information: https://docs.vlang.io/getting-started.html.

- Compile a file and output its executable:

```
v {{path/to/file.v}}
```

- Compile the current project or directory:

```
v .
```

- Compile a file and run it:

```
v run {{path/to/file.v}}
```

- Compile a file and run it, output its executable:

```
v crun {{path/to/file.v}}
```

- Re-compile on every modification to a file:

```
v watch {{path/to/file.v}}
```

- Re-run on every modification to a file:

```
v watch run {{path/to/file.v}}
```

- Open the v repl:

```
v repl
```

- Format a file and [w]rite to it:

```
v fmt -w {{path/to/file.v}}
```

vacuumdb

Garbage-collect and analyze a PostgreSQL database.

More information: https://www.postgresql.org/docs/current/app-vacuumdb.html.

- Vacuum a specific database:

```
vacuumdb {{database_name}}
```

- Vacuum all databases:

```
vacuumdb {{[-a|--all]}}
```

- Vacuum a specific table in a database:

```
vacuumdb {{[-t|--table]}} {{table_name}} {{database_name}}
```

- Vacuum and update statistics for the query planner:

```
vacuumdb {{[-z|--analyze]}} {{database_name}}
```

- Perform a full vacuum (more aggressive, locks tables, rewrites the whole table):

```
vacuumdb {{[-f|--full]}} {{database_name}}
```

- Vacuum with verbose output:

```
vacuumdb {{[-v|--verbose]}} {{database_name}}
```

- Vacuum a database using multiple parallel jobs:

```
vacuumdb --jobs {{number_of_jobs}} {{database_name}}
```

vagrant box

Manage Vagrant boxes (virtual machine images).

See also: **`vagrant`**.

More information: https://developer.hashicorp.com/vagrant/docs/cli/box.

- List all installed boxes:

```
vagrant box list
```

- Add a new box:

```
vagrant box add {{hashicorp/bionic64}}
```

- Add a box from a custom URL:

```
vagrant box add {{my-box}} {{https://example.com/my-box.box}}
```

- Remove an installed box:

```
vagrant box remove {{hashicorp/bionic64}}
```

- Update all boxes that are in use in the current Vagrant environment:

```
vagrant box update
```

- Update a specific box:

```
vagrant box update --box {{bento/debian-12}}
```

- Check if there is a new version available for the box that you are using:

```
vagrant box outdated
```

- Clean up old versions of installed boxes:

```
vagrant box prune
```

vagrant destroy

Stop a guest machine and destroy all its resources.

Any boxes installed are kept intact.

See also: **`vagrant`**.

More information: https://developer.hashicorp.com/vagrant/docs/cli/destroy.

- Destroy the currently running machine:

```
vagrant destroy
```

- Target a machine by name or ID:

```
vagrant destroy {{name|id}}
```

- Do not ask for confirmation before destroying:

```
vagrant destroy {{[-f|--force]}}
```

- Shut down the machine gracefully:

```
vagrant destroy {{[-g|--graceful]}}
```

vagrant global-status

Display the state of all Vagrant machines on the system.

The info is based on a cache, which may become stale and require pruning.

See also: **`vagrant`**, **`vagrant status`**.

More information: https://developer.hashicorp.com/vagrant/docs/cli/global-status.

- Display the state of all machines:

```
vagrant global-status
```

- Prune any stale entries from the output:

```
vagrant global-status --prune
```

vagrant halt

Shuts down the running machine Vagrant is managing.

See also: **`vagrant`**, **`vagrant box`**, **`vagrant plugin`**, **`vagrant validate`**.

More information: https://developer.hashicorp.com/vagrant/docs/cli/halt.

- Halt the currently running Vagrant machine gracefully:

  ```
  vagrant halt
  ```

- Halt a specific machine by its ID or name gracefully:

  ```
  vagrant halt {{id_or_name}}
  ```

- Forcefully halt the current running machine(s) (This can affect multiple running machines if they are part of the same Vagrant environment):

  ```
  vagrant halt {{[-f|--force]}}
  ```

- Forcefully halt a specific machine by its ID or name:

  ```
  vagrant halt {{[-f|--force]}} {{id_or_name}}
  ```

vagrant plugin

Manage Vagrant plugins.

See also: **`vagrant`**.

More information: https://developer.hashicorp.com/vagrant/docs/cli/plugin.

- List all the plugins currently installed:

```
vagrant plugin list
```

- Install a plugin from remote repositories, usually RubyGems:

```
vagrant plugin install {{vagrant_vbguest}}
```

- Install a plugin from a local file source:

```
vagrant plugin install {{path/to/my_plugin.gem}}
```

- Update all installed plugins to their latest version:

```
vagrant plugin update
```

- Update a plugin to the latest version:

```
vagrant plugin update {{vagrant_vbguest}}
```

- Uninstall a specific plugin:

```
vagrant plugin uninstall {{vagrant_vbguest}}
```

vagrant port

List mappings between guest and host ports.

See also: **`vagrant`**.

More information: https://developer.hashicorp.com/vagrant/docs/cli/port.

- List all port mappings of machine(s) running in the current directory:

```
vagrant port
```

- List mappings for a specific machine (if `Vagrantfile` is multi-machine):

```
vagrant port {{machine_name}}
```

- Display info for a specific guest port:

```
vagrant port --guest {{port_number}}
```

- Display machine-readable output:

```
vagrant port --machine-readable
```

vagrant reload

Equivalent of running **`halt`** followed by **`up`**.

A reload is usually required for changes in a Vagrantfile to take effect.

See also: **`vagrant`**.

More information: https://developer.hashicorp.com/vagrant/docs/cli/reload.

- Reload the currently running machine:

  ```
  vagrant reload
  ```

- Target a machine by name or ID:

  ```
  vagrant reload {{name|id}}
  ```

- Force the provisioners to run:

  ```
  vagrant reload --provision
  ```

vagrant resume

Resume a Vagrant managed machine that was previously suspended.

More information: https://developer.hashicorp.com/vagrant/docs/cli/resume.

- Resume machine specified by name or id:

```
vagrant resume {{name|id}}
```

- Resume and run all configured provisioners:

```
vagrant resume {{name|id}} --provision
```

- Resume and specify which provisioners to re-run:

```
vagrant resume {{name|id}} --provision-with {{provisioner}}
```

vagrant snapshot

Manage snapshots of Vagrant machines.

See also: **`vagrant`**.

More information: https://developer.hashicorp.com/vagrant/docs/cli/snapshot.

- Take a snapshot of the machine (running or stopped):

```
vagrant snapshot save {{snapshot_name}}
```

- Restore a snapshot and start the machine:

```
vagrant snapshot restore {{snapshot_name}}
```

- Restore a snapshot without starting the machine:

```
vagrant snapshot restore --no-start {{snapshot_name}}
```

- Delete a snapshot:

```
vagrant snapshot delete {{snapshot_name}}
```

- List available snapshots of the machine:

```
vagrant snapshot list
```

vagrant ssh

SSH into a running Vagrant machine.

See also: **`vagrant`**.

More information: https://developer.hashicorp.com/vagrant/docs/cli/ssh.

- SSH into the machine running in the current directory:

```
vagrant ssh
```

- Target a running machine by name or ID:

```
vagrant ssh {{name|id}}
```

- Execute an SSH command and exit:

```
vagrant ssh {{[-c|--command]}} {{ssh_command}}
```

- SSH without authentication, leaving authentication up to the user:

```
vagrant ssh {{[-p|--plain]}}
```

vagrant status

Display the state of machines in the current Vagrant environment.

See also: **`vagrant`**, **`vagrant global-status`**.

More information: https://developer.hashicorp.com/vagrant/docs/cli/status.

- View status of the machines in the current directory:

```
vagrant status
```

- Target a machine by name or ID:

```
vagrant status {{name|id}}
```

vagrant suspend

Save the state of the guest machine rather than fully shutting it down.

See also: **`vagrant`**.

More information: https://developer.hashicorp.com/vagrant/docs/cli/suspend.

- Suspend the currently running machine:

```
vagrant suspend
```

- Target a machine by name or ID:

```
vagrant suspend {{name|id}}
```

vagrant upload

Upload files and directories from the host to the guest machine.

More information: https://developer.hashicorp.com/vagrant/docs/cli/upload.

- Upload file or directory from the host to the guest machine:

```
vagrant upload {{path/to/source_file_or_directory}} {{path/
to/destination_file_or_directory}} {{name|id}}
```

- Compress the file or directory before uploading to guest machine:

```
vagrant upload --compress {{path/to/
source_file_or_directory}} {{path/to/
destination_file_or_directory}} {{name|id}}
```

- Specify which type of compression to use. Default type is `zip`:

```
vagrant upload --compression-type {{tgz|zip}} {{path/to/
source_file_or_directory}} {{path/to/
destination_file_or_directory}} {{name|id}}
```

- Create a temporary location on the guest machine and upload files to that location:

```
vagrant upload --temporary {{path/to/
source_file_or_directory}} {{path/to/
destination_file_or_directory}} {{name|id}}
```

vagrant validate

Check the validity of a Vagrantfile.

See also: **`vagrant`**, **`vagrant box`**, **`vagrant plugin`**.

More information: https://developer.hashicorp.com/vagrant/docs/cli/validate.

- Validate the syntax of the Vagrantfile to ensure it is correctly structured and free of errors:

```
vagrant validate
```

- Ensure that the Vagrantfile is correctly structured while ignoring provider-specific configuration options:

```
vagrant validate {{[-p|--ignore-provider]}} {{docker|hypervlibvirt|parallels|qemu|virtualbox|vmware_desktop}}
```

vagrant version

Display Vagrant version information.

Note: Internet access is required for vagrant version to check the latest available version.

More information: https://developer.hashicorp.com/vagrant/docs/cli/version.

- Display the currently installed version and latest version of Vagrant that is currently available:

  ```
  vagrant version
  ```

- Display version:

  ```
  vagrant --version
  ```

vagrant

Manage lightweight, reproducible, and portable development environments.

Some subcommands such as **`box`**, **`snapshot`**, **`halt`**, etc. have their own usage documentation.

More information: https://developer.hashicorp.com/vagrant/docs/cli.

- Create a `Vagrantfile` in the current directory with the base Vagrant box:

  ```
  vagrant init
  ```

- Create a `Vagrantfile` with a box from the Vagrant Public Registry:

  ```
  vagrant init {{ubuntu/focal64}}
  ```

- Start and provision the Vagrant environment:

  ```
  vagrant up
  ```

- Suspend the machine:

  ```
  vagrant suspend
  ```

- Halt the machine:

  ```
  vagrant halt
  ```

- Connect to the machine via SSH:

  ```
  vagrant ssh
  ```

- Output the SSH configuration file of the running Vagrant machine:

  ```
  vagrant ssh-config
  ```

- List all local boxes:

  ```
  vagrant box list
  ```

vala

Vala code runner.

Tutorial: https://wiki.gnome.org/Projects/Vala/Tutorial.

More information: https://docs.vala.dev/index.html.

- Run a vala file, with gtk+:

```
vala {{path/to/file.vala}} --pkg {{gtk+-3.0}}
```

- Display help:

```
vala --help
```

- Display version:

```
vala --version
```

valac

Vala code compiler.

Tutorial: https://wiki.gnome.org/Projects/Vala/Tutorial.

More information: https://docs.vala.dev/tutorials/programming-language/main/07-00-tools/07-01-valac.html.

- Compile a vala file, with gtk+:

```
valac {{path/to/file.vala}} --pkg {{gtk+-3.0}}
```

- Display help:

```
valac --help
```

- Display version:

```
valac --version
```

vale

Extensible style checker that supports multiple markup formats, such as Markdown and AsciiDoc.

More information: https://vale.sh.

- Check the style of a file:

```
vale {{path/to/file}}
```

- Check the style of a file with a specified configuration:

```
vale --config='{{path/to/.vale.ini}}' {{path/to/file}}
```

- Output the results in JSON format:

```
vale --output=JSON {{path/to/file}}
```

- Check style issues at the specific severity and higher:

```
vale --minAlertLevel={{suggestion|warning|error}} {{path/to/file}}
```

- Check the style from `stdin`, specifying markup format:

```
cat {{file.md}} | vale --ext=.md
```

- List the current configuration:

```
vale ls-config
```

valgrind

Wrapper for a set of expert tools for profiling, optimizing and debugging programs.

Commonly used tools include **`memcheck`**, **`cachegrind`**, **`callgrind`**, **`massif`**, **`helgrind`**, and **`drd`**.

More information: https://valgrind.org/docs/manual/manual-core.html#manual-core.options.

- Use the (default) Memcheck tool to show a diagnostic of memory usage by `program`:

  ```
  valgrind {{program}}
  ```

- Use Memcheck to report all possible memory leaks of `program` in full detail:

  ```
  valgrind --leak-check=full --show-leak-kinds=all {{program}}
  ```

- Use the Cachegrind tool to profile and log CPU cache operations of `program`:

  ```
  valgrind --tool=cachegrind {{program}}
  ```

- Use the Massif tool to profile and log heap memory and stack usage of `program`:

  ```
  valgrind --tool=massif --stacks=yes {{program}}
  ```

var-dump-server

Symfony dump server.

Collects data dumped by the Symfony VarDumper component.

More information: https://symfony.com/doc/current/components/var_dumper.html#the-dump-server.

- Start the server:

  ```
  var-dump-server
  ```

- Dump the data in an HTML file:

  ```
  var-dump-server --format=html > {{path/to/file.html}}
  ```

- Make the server listen on a specific address and port:

  ```
  var-dump-server --host {{127.0.0.1:9912}}
  ```

varnishlog

Display Varnish logs.

More information: https://varnish-cache.org/docs/trunk/reference/varnishlog.html.

- Display logs in real time:

```
varnishlog
```

- Only display requests to a specific domain:

```
varnishlog -q 'ReqHeader eq "Host: {{example.com}}"'
```

- Only display POST requests:

```
varnishlog -q 'ReqMethod eq "{{POST}}"'
```

- Only display requests to a specific path:

```
varnishlog -q 'ReqURL eq "/{{path}}"'
```

- Only display requests to paths matching a `regex`:

```
varnishlog -q 'ReqURL ~ "{{regex}}"'
```

vault

Interact with HashiCorp Vault.

More information: https://www.vaultproject.io/docs/commands.

- Connect to a Vault server and initialize a new encrypted data store:

  ```
  vault init
  ```

- Unseal (unlock) the vault, by providing one of the key shares needed to access the encrypted data store:

  ```
  vault unseal {{key-share-x}}
  ```

- Authenticate the CLI client against the Vault server, using an authentication token:

  ```
  vault auth {{authentication_token}}
  ```

- Store a new secret in the vault, using the generic back-end called "secret":

  ```
  vault write secret/{{hello}} value={{world}}
  ```

- Read a value from the vault, using the generic back-end called "secret":

  ```
  vault read secret/{{hello}}
  ```

- Read a specific field from the value:

  ```
  vault read -field={{field_name}} secret/{{hello}}
  ```

- Seal (lock) the Vault server, by removing the encryption key of the data store from memory:

  ```
  vault seal
  ```

VBoxManage clonevm

Create a clone of an existing virtual machine (VM).

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-clonevm.

- Clone the specified VM:

```
VBoxManage clonevm {{vm_name}}
```

- Specify a new name for the new VM:

```
VBoxManage clonevm {{vm_name}} --name {{new_vm_name}}
```

- Indicate the folder where the new VM configuration is saved:

```
VBoxManage clonevm {{vm_name}} --basefolder {{path/to/directory}}
```

- Register the cloned VM in VirtualBox:

```
VBoxManage clonevm {{vm_name}} --register
```

VBoxManage cloud

VirtualBox command-line interface for managing cloud instances and images.

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-cloud.

- List the instances in the specified state belonging to the specified compartment:

```
VBoxManage cloud --provider={{provider_name}} --
profile={{profile_name}} list instances --state={{running|
terminated|paused}} --compartment-id={{compartment_id}}
```

- Create a new instance:

```
VBoxManage cloud --provider={{provider_name}} --
profile={{profile_name}} instance create --domain-
name={{domain_name}} --image-id={{image_id}}
```

- Gather information about a particular instance:

```
VBoxManage cloud --provider={{provider_name}} --
profile={{profile_name}} instance info --id={{unique_id}}
```

- Terminate an instance:

```
VBoxManage cloud --provider={{provider_name}} --
profile={{profile_name}} instance terminate --
id={{unique_id}}
```

- List images within a specific compartment and state:

```
VBoxManage cloud --provider={{provider_name}} --
profile={{profile_name}} list images --compartment-
id={{compartment_id}} --state={{state_name}}
```

- Create a new image:

```
VBoxManage cloud --provider={{provider_name}} --
profile={{profile_name}} image create --instance-
id={{instance_id}} --display-name={{display_name}} --
compartment-id={{compartment_id}}
```

- Retrieve information about a particular image:

```
VBoxManage cloud --provider={{provider_name}} --
profile={{profile_name}} image info --id={{unique_id}}
```

- Delete an image:

```
VBoxManage cloud --provider={{provider_name}} --
profile={{profile_name}} image delete --id={{unique_id}}
```

VBoxManage controlvm

Change the state and the settings of a currently running virtual machine.

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-controlvm.

- Temporarily stop the execution of a virtual machine:

```
VBoxManage controlvm {{uuid|vm_name}} pause
```

- Resume the execution of a paused virtual machine:

```
VBoxManage controlvm {{uuid|vm_name}} resume
```

- Perform a cold reset on the virtual machine:

```
VBoxManage controlvm {{uuid|vm_name}} reset
```

- Poweroff a virtual machine with the same effect as pulling the power cable of a computer:

```
VBoxManage controlvm {{uuid|vm_name}} poweroff
```

- Shutdown the virtual machine and save its current state:

```
VBoxManage controlvm {{uuid|vm_name}} savestate
```

- Send an ACPI (Advanced Configuration and Power Interface) shutdown signal to the virtual machine:

```
VBoxManage controlvm {{uuid|vm_name}} acpipowerbutton
```

- Send command to reboot itself to the guest OS:

```
VBoxManage controlvm {{uuid|vm_name}} reboot
```

- Shutdown down the virtual machine without saving its state:

```
VBoxManage controlvm {{uuid|vm_name}} shutdown
```

VBoxManage createvm

Create a new virtual machine.

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-createvm.

- Create a new VM with default settings:

```
VBoxManage createvm --name {{vm_name}}
```

- Set the base folder where the VM configuration will be stored:

```
VBoxManage createvm --name {{vm_name}} --basefolder {{path/to/directory}}
```

- Set the guest OS type (one of `VBoxManage list ostypes`) for the imported VM:

```
VBoxManage createvm --name {{vm_name}} --ostype {{ostype}}
```

- Register the created VM in VirtualBox:

```
VBoxManage createvm --name {{vm_name}} --register
```

- Set the VM to the specified groups:

```
VBoxManage createvm --name {{vm_name}} --group {{group1,group2,...}}
```

- Set the Universally Unique Identifier (UUID) of the VM:

```
VBoxManage createvm --name {{vm_name}} --uuid {{uuid}}
```

- Set the cipher to use for encryption:

```
VBoxManage createvm --name {{vm_name}} --cipher {{AES-128|AES-256}}
```

VBoxManage export

Export virtual machines to a virtual appliance (ISO) or a cloud service.

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-export.

- Specify the target OVA file:

```
VBoxManage export --output {{path/to/file.ova}}
```

- Export in OVF 0.9 legacy mode:

```
VBoxManage export --legacy09
```

- Export in OVF (0.9|1.0|2.0) format:

```
VBoxManage export --{{ovf09|ovf10|ovf20}}
```

- Create manifest of the exported files:

```
VBoxManage export --manifest
```

- Specify a description of the VM:

```
VBoxManage export --description "{{vm_description}}"
```

VBoxManage extpack

Manage extension packs for Oracle VirtualBox.

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-extpack.

- Install extension packs to VirtualBox (Note: You need to remove the existing version of the extension pack before installing a new version.):

```
VBoxManage extpack install {{path/to/file.vbox-extpack}}
```

- Remove the existing version of the VirtualBox extension pack:

```
VBoxManage extpack install --replace
```

- Uninstall extension packs from VirtualBox:

```
VBoxManage extpack uninstall {{extension_pack_name}}
```

- Uninstall extension packs and skip most uninstallation refusals:

```
VBoxManage extpack uninstall --force {{extension_pack_name}}
```

- Clean up temporary files and directories left by extension packs:

```
VBoxManage extpack cleanup
```

VBoxManage import

Import a previously exported virtual machine (VM).

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-import.

- Import a VM from an OVF or OVA file:

```
VBoxManage import {{path/to/file.ovf}}
```

- Set the name of the imported VM:

```
VBoxManage import {{path/to/file.ovf}} --name {{vm_name}}
```

- Indicate the folder where the configuration of the imported VM will be stored:

```
VBoxManage import {{path/to/file.ovf}} --basefolder {{path/to/directory}}
```

- Register the imported VM in VirtualBox:

```
VBoxManage import {{path/to/file.ovf}} --register
```

- Perform a dry run to check the import without actually importing:

```
VBoxManage import {{path/to/file.ovf}} --dry-run
```

- Set the guest OS type (one of `VBoxManage list ostypes`) for the imported VM:

```
VBoxManage import {{path/to/file.ovf}} --ostype {{ostype}}
```

- Set the memory (in megabytes) for the imported VM:

```
VBoxManage import {{path/to/file.ovf}} --memory {{1}}
```

- Set the number of CPUs for the imported VM:

```
VBoxManage import {{path/to/file.ovf}} --cpus {{1}}
```

VBoxManage list

List information about the Oracle VM VirtualBox software and associated service.

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-list.

- List all VirtualBox virtual machines:

```
VBoxManage list vms
```

- Show DHCP servers available on the host system:

```
VBoxManage list dhcpservers
```

- Show Oracle VM VirtualBox extension packs currently installed:

```
VBoxManage list extpacks
```

- Show all virtual machine groups:

```
VBoxManage list groups
```

- Show virtual disk settings that are currently in use by VirtualBox:

```
VBoxManage list hdds
```

- Show host-only network interfaces available on host system:

```
VBoxManage list hostonlyifs
```

- Show the list of currently running virtual machines:

```
VBoxManage list runningvms
```

- Show host system information:

```
VBoxManage list hostinfo
```

VBoxManage modifyvm

Change settings for a virtual machine that is stopped.

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-modifyvm.

- Rename the VM:

```
VBoxManage modifyvm {{uuid|vm_name}} --name {{new_name}}
```

- Adjust memory and CPU:

```
VBoxManage modifyvm {{uuid|vm_name}} --memory {{2048}} --cpus {{2}}
```

- Enable Remote Display (VRDE):

```
VBoxManage modifyvm {{uuid|vm_name}} --vrde on
```

- Enable session recording:

```
VBoxManage modifyvm {{uuid|vm_name}} --recording on
```

VBoxManage movevm

Move a virtual machine (VM) to a new location on the host system.

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-movevm.

- Move the specified virtual machine to the current location:

```
VBoxManage movevm {{vm_name}}
```

- Specify the new location (full or relative pathname) of the virtual machine:

```
VBoxManage movevm {{vm_name}} --folder {{path/to/new_location}}
```

VBoxManage registervm

Register a virtual machine (VM).

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-registervm.

- Register an existing VM:

```
VBoxManage registervm {{path/to/file.vbox}}
```

- Supply the encryption password file of the VM:

```
VBoxManage registervm {{path/to/file.vbox}} --password
{{path/to/password_file}}
```

- Prompt for the encryption password on the command-line:

```
VBoxManage registervm {{path/to/file.vbox}} --password -
```

VBoxManage showvminfo

Show information about registered virtual machine.

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-showvminfo.

- Show information about a particular virtual machine:

```
VBoxManage showvminfo {{vm_name|uuid}}
```

- Show more detailed information about a particular virtual machine:

```
VBoxManage showvminfo --details {{vm_name|uuid}}
```

- Show information in a machine readable format:

```
VBoxManage showvminfo --machinereadable {{vm_name|uuid}}
```

- Specify password ID if the virtual machine is encrypted:

```
VBoxManage showvminfo --password-id {{password_id}}
{{vm_name|uuid}}
```

- Specify the password file if the virtual machine is encrypted:

```
VBoxManage showvminfo --password {{path/to/password_file}}
{{vm_name|uuid}}
```

- Show the logs of a specific virtual machine:

```
VBoxManage showvminfo --log {{vm_name|uuid}}
```

VBoxManage startvm

Start a virtual machine.

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-startvm.

- Start a virtual machine:

```
VBoxManage startvm {{vm_name|uuid}}
```

- Start a virtual machine with the specified UI mode:

```
VBoxManage startvm {{vm_name|uuid}} --type {{headless|gui|sdl|separate}}
```

- Specify a password file to start an encrypted virtual machine:

```
VBoxManage startvm {{vm_name|uuid}} --password {{path/to/password_file}}
```

- Specify a password ID to start an encrypted virtual machine:

```
VBoxManage startvm {{vm_name|uuid}} --password-id {{password_id}}
```

- Start a virtual machine with an environment variable pair name value:

```
VBoxManage startvm {{vm_name|uuid}} {{[-E|--putenv]}} {{name}}={{value}}
```

VBoxManage unregistervm

Unregister a virtual machine (VM).

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-unregistervm.

- Unregister an existing VM:

```
VBoxManage unregistervm {{uuid|vm_name}}
```

- Delete hard disk image files, all saved state files, VM logs, and XML VM machine files:

```
VBoxManage unregistervm {{uuid|vm_name}} --delete
```

- Delete all files from the VM:

```
VBoxManage unregistervm {{uuid|vm_name}} --delete-all
```

VBoxManage

Command-line interface to VirtualBox.

Includes all the functionality of the GUI and more.

Some subcommands such as **`startvm`** have their own usage documentation.

More information: https://www.virtualbox.org/manual/ch08.html#vboxmanage-intro.

- List all VirtualBox virtual machines:

```
VBoxManage list vms
```

- Show information about a particular virtual machine:

```
VBoxManage showvminfo {{name|uuid}}
```

- Start a virtual machine:

```
VBoxManage startvm {{name|uuid}}
```

- Start a virtual machine in headless mode:

```
VBoxManage startvm {{name|uuid}} --type headless
```

- Shutdown the virtual machine and save its current state:

```
VBoxManage controlvm {{name|uuid}} savestate
```

- Shutdown down the virtual machine without saving its state:

```
VBoxManage controlvm {{name|uuid}} poweroff
```

- Update VBox extension packs:

```
VBoxManage extpack install --replace
{{VboxExtensionPackFileName}}
```

- Display help:

```
VBoxManage --help
```

vc

This command is an alias of **`vercel`**.

- View documentation for the original command:

  ```
  tldr vercel
  ```

vcluster

Create and manage lightweight virtual Kubernetes clusters in namespaces.

More information: https://www.vcluster.com/docs/vcluster.

- Create a virtual cluster in a specific namespace:

```
vcluster create {{vcluster_name}} {{[-n|--namespace]}}
{{namespace}}
```

- Connect to a virtual cluster with a local port and insecure mode:

```
vcluster connect {{vcluster_name}} {{[-n|--namespace]}}
{{namespace}} --local-port {{port}} --insecure
```

- List all virtual clusters:

```
vcluster list
```

- Delete a virtual cluster:

```
vcluster delete {{vcluster_name}}
```

- List platform-managed virtual clusters:

```
vcluster platform list
```

- Create a platform-managed virtual cluster:

```
vcluster platform create {{vcluster_name}} {{[-n|--
namespace]}} {{namespace}}
```

- Connect to a platform-managed virtual cluster:

```
vcluster platform connect {{vcluster_name}} {{[-n|--
namespace]}} {{namespace}}
```

- Delete a platform-managed virtual cluster:

```
vcluster platform delete {{vcluster_name}} {{[-n|--
namespace]}} {{namespace}}
```

vcpkg

Package manager for C/C++ libraries.

Note: Packages are not installed in the system. To use them, you need to tell your build system (e.g. CMake) to use **`vckg`**.

More information: https://learn.microsoft.com/en-us/vcpkg/.

- Build and add package `libcurl` to the `vcpkg` environment:

  ```
  vcpkg install curl
  ```

- Build and add `zlib` using the `emscripten` toolchain:

  ```
  vcpkg install --triplet=wasm32-emscripten zlib
  ```

- Search for a package:

  ```
  vcpkg search {{pkg_name}}
  ```

- Configure a CMake project to use `vcpkg` packages:

  ```
  cmake -B build -DCMAKE_TOOLCHAIN_FILE={{path/to/
  vcpkg_install_directory}}/scripts/buildsystems/vcpkg.cmake
  ```

vcsh

Version Control System for the home directory using Git repositories.

See also: **`chezmoi`**, **`stow`**, **`tuckr`**, **`homeshick`**.

More information: https://github.com/RichiH/vcsh.

- Initialize an (empty) repository:

```
vcsh init {{repository_name}}
```

- Clone a repository into a custom directory name:

```
vcsh clone {{git_url}} {{repository_name}}
```

- List all managed repositories:

```
vcsh list
```

- Execute a Git command on a managed repository:

```
vcsh {{repository_name}} {{git_command}}
```

- Push/pull all managed repositories to/from remotes:

```
vcsh {{push|pull}}
```

- Write a custom `.gitignore` file for a managed repository:

```
vcsh write-gitignore {{repository_name}}
```

vdir

Verbosely list directory contents.

Drop-in replacement for **`ls -l -b`**.

More information: https://manned.org/vdir.

- List files and directories in the current directory, one per line, with details:

```
vdir
```

- List with sizes displayed in human-readable units (KB, MB, GB):

```
vdir {{[-h|--human-readable]}}
```

- List including hidden files (starting with a dot):

```
vdir {{[-a|--all]}}
```

- List files and directories sorting entries by size (largest first):

```
vdir -S
```

- List files and directories sorting entries by modification time (newest first):

```
vdir -t
```

- List grouping directories first:

```
vdir --group-directories-first
```

- Recursively list all files and directories in a specific directory:

```
vdir {{[-R|--recursive]}} {{path/to/directory}}
```

vectorize-pixelart

Convert PNG pixel art graphics to SVG/EPS vector images.

More information: https://github.com/und3f/vectorize-pixelart.

- Convert a PNG to a vector image format:

```
vectorize-pixelart {{path/to/input.png}} {{path/to/
output.svg|.eps}}
```

vegeta

A utility and a library for HTTP load testing.

See also: **ab**.

More information: https://github.com/tsenart/vegeta.

- Launch an attack lasting 30 seconds:

```
echo "{{GET https://example.com}}" | vegeta attack -
duration={{30s}}
```

- Launch an attack on a server with a self-signed HTTPS certificate:

```
echo "{{GET https://example.com}}" | vegeta attack -insecure
-duration={{30s}}
```

- Launch an attack with a rate of 10 requests per second:

```
echo "{{GET https://example.com}}" | vegeta attack -
duration={{30s}} -rate={{10}}
```

- Launch an attack and display a report:

```
echo "{{GET https://example.com}}" | vegeta attack -
duration={{30s}} | vegeta report
```

- Launch an attack and plot the results on a graph (latency over time):

```
echo "{{GET https://example.com}}" | vegeta attack -
duration={{30s}} | vegeta plot > {{path/to/results.html}}
```

- Launch an attack against multiple URLs from a file:

```
vegeta attack -duration={{30s}} -targets={{requests.txt}} |
vegeta report
```

vela

Tools for the Vela pipeline.

More information: https://go-vela.github.io/docs/reference/cli.

- Trigger a pipeline to run from a Git branch, commit or tag:

```
vela add deployment --org {{organization}} --repo
{{repository_name}} --target {{environment}} --ref {{branch|
commit|refs/tags/git_tag}} --description
"{{deploy_description}}"
```

- List deployments for a repository:

```
vela get deployment --org {{organization}} --repo
{{repository_name}}
```

- Inspect a specific deployment:

```
vela view deployment --org {{organization}} --repo
{{repository_name}} --deployment {{deployment_number}}
```

velero

Backup and migrate Kubernetes applications and their persistent volumes.

More information: https://velero.io/docs/main/.

- Create a backup containing all resources:

  ```
  velero backup create {{backup_name}}
  ```

- List all backups:

  ```
  velero backup get
  ```

- Delete a backup:

  ```
  velero backup delete {{backup_name}}
  ```

- Create a weekly backup, each living for 90 days (2160 hours):

  ```
  velero schedule create {{schedule_name}} --
  schedules="{{@every 7d}}" --ttl {{2160h0m0s}}
  ```

- Create a restore from the latest successful backup triggered by specific schedule:

  ```
  velero restore create --from-schedule {{schedule_name}}
  ```

venv

Create lightweight virtual environments in Python.

More information: https://docs.python.org/library/venv.html.

- Create a Python virtual environment:

```
python -m venv {{path/to/virtual_environment}}
```

- Activate the virtual environment (Linux and macOS):

```
{{[.|source]}} {{path/to/virtual_environment}}/bin/activate
```

- Activate the virtual environment (Windows):

```
{{path\to\virtual_environment}}\Scripts\activate.bat
```

- Deactivate the virtual environment:

```
deactivate
```

- Create an alias that generates a `venv` folder and automatically activates it:

```
alias venv='python -m venv .venv && source {{.venv/bin/
activate|.venv\Scripts\activate.bat}}'
```

vercel

Deploy and manage your Vercel deployments.

More information: https://vercel.com/docs/cli.

- Deploy the current directory:

```
vercel
```

- Deploy the current directory to production:

```
vercel --prod
```

- Deploy a directory:

```
vercel {{path/to/project}}
```

- Initialize an example project:

```
vercel init
```

- Deploy with environment variables:

```
vercel {{[-e|--env]}} {{ENV}}={{var}}
```

- Build with environment variables:

```
vercel {{[-b|--build-env]}} {{ENV}}={{var}}
```

- Set default regions to enable the deployment on:

```
vercel --regions {{region_id}}
```

- Remove a deployment:

```
vercel remove {{project_name}}
```

verilator

Convert Verilog and SystemVerilog hardware description language (HDL) design into a C++ or SystemC model to be executed after compiling.

More information: https://veripool.org/guide/latest/.

- Build a specific C project in the current directory:

```
verilator --binary --build-jobs 0 -Wall {{path/to/source.v}}
```

- Create a C++ executable in a specific folder:

```
verilator --cc --exe --build --build-jobs 0 -Wall {{path/to/source.cpp}} {{path/to/output.v}}
```

- Perform linting over a code in the current directory:

```
verilator --lint-only -Wall
```

- Create XML output about the design (files, modules, instance hierarchy, logic and data types) to feed into other tools:

```
verilator --xml-output -Wall {{path/to/output.xml}}
```

|

Pipe data between programs.

More information: https://gnu.org/software/bash/manual/bash.html#Pipelines.

- Pipe `stdout` to `stdin`:

```
{{command}} | {{command}}
```

- Pipe both `stdout` and `stderr` to `stdin`:

```
{{command}} |& {{command}}
```

vet

Scan package manifests, directories, container images, or GitHub repositories to identify vulnerabilities, malicious packages, and enforce security policies using CEL expressions.

More information: https://safedep.github.io/vet/vet.html.

- Scan the current directory:

```
vet scan {{[-D|--directory]}} .
```

- Scan the `package-lock.json` manifest file:

```
vet scan {{[-M|--manifests]}} {{path/to/manifest_file}}
```

- Scan with filter to fail on any critical vulnerability found in the codebase:

```
vet scan {{[-D|--directory]}} {{path/to/directory}} --filter
'vulns.critical.exists(p, true)' --filter-fail
```

- Scan any OSS package for malware:

```
vet inspect malware --purl {{package_url}}
```

- Start the MCP server for AI enabled security in code editors like Cursor:

```
vet server mcp
```

vf

VirtualFish is a fish shell tool for managing Python virtual environments.

More information: https://virtualfish.readthedocs.io/en/latest/.

- Create a virtual environment:

```
vf new {{virtualenv_name}}
```

- Create a virtual environment for a specific Python version:

```
vf new --python {{/usr/local/bin/python3.8}}
{{virtualenv_name}}
```

- Activate and use the specified virtual environment:

```
vf activate {{virtualenv_name}}
```

- Connect the current virtualenv to the current directory, so that it is activated automatically as soon as you enter it (and deactivated as soon as you leave):

```
vf connect
```

- Deactivate the current virtual environment:

```
vf deactivate
```

- List all virtual environments:

```
vf ls
```

- Remove a virtual environment:

```
vf rm {{virtualenv_name}}
```

- Display help:

```
vf help
```

vgmstream_cli

Play a wide variety of audio formats used in video games and convert them into **`wav`**.

More information: https://github.com/vgmstream/vgmstream/blob/master/doc/USAGE.md.

- Decode an `adc` file to `wav`. (Default output name is `input.wav`):

```
vgmstream_cli {{path/to/input.adc}} -o {{path/to/output.wav}}
```

- Print metadata without decoding the audio:

```
vgmstream_cli {{path/to/input.adc}} -m
```

- Decode an audio file without loops:

```
vgmstream_cli {{path/to/input.adc}} -o {{path/to/output.wav}} -i
```

- Decode with three loops, then add a 3s delay followed by a 5s fadeout:

```
vgmstream_cli {{path/to/input.adc}} -o {{path/to/output.wav}} -l {{3.0}} -f {{5.0}} -d {{3.0}}
```

- Convert multiple files to `bgm_(original name).wav` (Default `-o` pattern is `?f.wav`):

```
vgmstream_cli -o {{path/to/bgm_?f.wav}} {{path/to/file1.adc path/to/file2.adc ...}}
```

- Play the file looping endlessly (`channels` and `rate` must match metadata):

```
vgmstream_cli {{path/to/input.adc}} -pec | aplay --format cd --channels {{1}} --rate {{44100}}
```

vgrep

A user friendly pager for grep.

See also: **`ugrep`**, **`rg`**.

More information: https://github.com/vrothberg/vgrep/blob/main/docs/vgrep.1.md.

- Recursively search the current directory for a pattern and cache it:

```
vgrep {{search_pattern}}
```

- Display the contents of the cache:

```
vgrep
```

- Open the "4th" match from the cache in the default editor:

```
vgrep {{[-s|--show]}} {{4}}
```

- Display a context of "3" lines for each match in the cache:

```
vgrep {{[-s|--show]}} {{[c|context]}}{{3}}
```

- Display the number of matches for each directory in the tree:

```
vgrep {{[-s|--show]}} {{[t|tree]}}
```

- Display the number of matches for each file in the tree:

```
vgrep {{[-s|--show]}} {{[f|files]}}
```

- Start an interactive shell with cached matches:

```
vgrep {{[-i|--interactive]}}
```

vhs

Generate terminal GIFs from a tape file.

More information: https://github.com/charmbracelet/vhs.

- Create a tape file (add commands to the tape file using an editor):

```
vhs new {{path/to/file.tape}}
```

- Record inputs to a tape file:

```
vhs record > {{path/to/file.tape}}
```

- Record inputs to a tape file using a specific shell:

```
vhs record {{[-s|--shell]}} {{shell}} > {{path/to/file.tape}}
```

- Stop recording:

```
exit
```

- Validate the syntax of a tape file:

```
vhs validate {{path/to/file.tape}}
```

- Create a GIF from a tape file:

```
vhs {{path/to/file.tape}}
```

- Publish a GIF to https://vhs.charm.sh and get a shareable URL:

```
vhs publish {{path/to/file.gif}}
```

vi

This command is an alias of **vim**.

- View documentation for the original command:

```
tldr vim
```

vidir

Edit directories in a text editor.

More information: https://manned.org/vidir.

- Edit the contents of the specified directories:

```
vidir {{path/to/directory1 path/to/directory2 ...}}
```

- Display each action taken by the program:

```
vidir {{[-v|--verbose]}} {{path/to/directory1 path/to/
directory2 ...}}
```

- Edit the contents of current directory:

```
vidir
```

- Use the specified text editor:

```
EDITOR={{vim}} vidir {{path/to/directory1 path/to/
directory2 ...}}
```

- Read a list of files to edit from `stdin`:

```
{{command}} | vidir -
```

view

A read-only version of **`vim`**.

This is equivalent to **`vim -R`**.

More information: https://www.vim.org.

- Open a file:

```
view {{path/to/file}}
```

vifm

VI File Manager is a TUI file manager.

See also: **`clifm`**, **`vifm`**, **`mc`**, **`caja`**.

More information: https://github.com/vifm/vifm.

- Open the current directory:

```
vifm .
```

- Open specified directories on left or right plane:

```
vifm {{path/to/directory1 path/to/directory2 ...}}
```

vim

Vim (Vi IMproved), a command-line text editor, provides several modes for different kinds of text manipulation.

Pressing **<i>** in normal mode enters insert mode. Pressing **<Esc>** goes back to normal mode, which enables the use of Vim commands.

See also: **vimdiff**, **vimtutor**, **nvim**, **gvim**.

More information: https://www.vim.org.

- Open a file:

```
vim {{path/to/file}}
```

- Open a file at a specified line number:

```
vim +{{line_number}} {{path/to/file}}
```

- View Vim's help manual:

```
<:>help<Enter>
```

- Save and quit the current buffer:

```
{{<Esc><Z><Z>|<Esc><:>x<Enter>|<Esc><:>wq<Enter>}}
```

- Enter normal mode and undo the last operation:

```
<Esc><u>
```

- Search for a pattern in the file (press <n>/<N> to go to next/previous match):

```
</>{{search_pattern}}<Enter>
```

- Perform a regex substitution in the whole file:

```
<:>%s/{{regex}}/{{replacement}}/g<Enter>
```

- Display the line numbers:

```
<:>set nu<Enter>
```

vimdiff

Open up two or more files in vim and show the differences between them.

See also: **`vim`**, **`vimtutor`**, **`nvim`**.

More information: https://www.vim.org.

- Open two files and show the differences:

```
vimdiff {{path/to/file1}} {{path/to/file2}}
```

- Move the cursor to the window on the left|right:

```
<Ctrl w>{{<h>|<l>}}
```

- Jump to the previous difference:

```
<[><c>
```

- Jump to the next difference:

```
<]><c>
```

- Copy the highlighted difference from the other window to the current window:

```
<d><o>
```

- Copy the highlighted difference from the current window to the other window:

```
<d><p>
```

- Update all highlights and folds:

```
<:>diffupdate
```

- Toggle the highlighted code fold:

```
<z><a>
```

vimtutor

Vim tutor, teaching the basic vim commands.

See also: **`vim`**, **`vimdiff`**, **`nvim`**.

More information: https://manned.org/vimtutor.

- Launch the vim tutor using the given language:

```
vimtutor {{en|fr|de|...}}
```

- Exit the tutor:

```
<Esc><:>q<Enter>
```

vinmap

A multithreaded Nmap scanner that splits IP ranges into chunks, performs parallel scans, and merges XML or JSON results.

More information: https://pypi.org/project/vinmap.

- Perform a basic scan of a subnet:

```
vinmap {{[-ip|--ip_range]}} {{192.168.1.0/24}}
```

- Scan a domain with version and OS detection, saving results to a specific file:

```
vinmap {{[-ip|--ip_range]}} {{example.com}} {{[-s|--
scan_type]}} "-sV -O" -o {{path/to/scan_results.xml}}
```

- Scan an IP range using 10 chunks and 20 concurrent threads, uses half of the system's CPU cores if not specified:

```
vinmap {{[-ip|--ip_range]}} {{10.0.0.1-10.0.0.255}} {{[-n|--
num_chunks]}} 10 {{[-t|--threads]}} 20
```

- Output scan results in JSON format:

```
vinmap {{[-ip|--ip_range]}} {{192.168.1.1-192.168.1.100}}
{{[-f|--format]}} json
```

- Scan multiple IPs with default settings and save merged XML output:

```
vinmap {{[-ip|--ip_range]}} {{192.168.1.1,192.168.1.2,...}}
```

vipe

Run a text editor in the middle of a UNIX pipeline.

More information: https://manned.org/vipe.

- Edit the output of `command1` before piping it into `command2`:

```
{{command1}} | vipe | {{command2}}
```

- Buffer the output of `command1` in a temporary file with the specified file extension in order to aid syntax highlighting:

```
{{command1}} | vipe --suffix {{json}} | {{command2}}
```

- Use the specified text editor:

```
{{command1}} | EDITOR={{vim}} vipe | {{command2}}
```

virsh connect

Connect to a virtual machine hypervisor.

See also: **`virsh`**.

More information: https://manned.org/virsh.

- Connect to the default hypervisor:

```
virsh connect
```

- Connect as root to the local QEMU/KVM hypervisor:

```
virsh connect qemu:///system
```

- Launch a new instance of the hypervisor and connect to it as the local user:

```
virsh connect qemu:///session
```

- Connect as root to a remote hypervisor using SSH:

```
virsh connect qemu+ssh://{{user_name@host_name}}/system
```

virsh domblklist

List information about block devices associated with a virtual machine.

See also: **`virsh`**.

More information: https://manned.org/virsh.

- List the target name and source path of the block devices:

```
virsh domblklist --domain {{vm_name}}
```

- List the disk type and device value as well as the target name and source path:

```
virsh domblklist --domain {{vm_name}} --details
```

virsh help

Display information about **`virsh`** commands or command groups.

See also: **`virsh`**.

More information: https://manned.org/virsh.

- List the `virsh` commands grouped into related categories:

```
virsh help
```

- List the command categories:

```
virsh help | grep "keyword"
```

- List the commands in a category:

```
virsh help {{category_keyword}}
```

- Display help for a command:

```
virsh help {{command}}
```

virsh list

List the ID, name, and state of virtual machines.

See also: **`virsh`**.

More information: https://manned.org/virsh.

- List information about running virtual machines:

```
virsh list
```

- List information about virtual machines regardless of state:

```
virsh list --all
```

- List information about virtual machines with autostart either enabled or disabled:

```
virsh list --all --{{autostart|no-autostart}}
```

- List information about virtual machines either with or without snapshots:

```
virsh list --all --{{with-snapshot|without-snapshot}}
```

virsh pool-autostart

Enable or disable autostart for a virtual machine storage pool.

See also: **`virsh`**.

More information: https://manned.org/virsh.

- Enable autostart for the storage pool specified by name or UUID (determine using `virsh pool-list`):

```
virsh pool-autostart --pool {{name|uuid}}
```

- Disable autostart for the storage pool specified by name or UUID:

```
virsh pool-autostart --pool {{name|uuid}} --disable
```

virsh pool-build

Build the underlying storage system for a virtual machine storage pool as defined in it's configuration file in **`/etc/libvirt/storage`**.

See also: **`virsh`**, **`virsh-pool-define-as`**, **`virsh-pool-start`**.

More information: https://manned.org/virsh.

- Build the storage pool specified by name or UUID (determine using `virsh pool-list`):

```
virsh pool-build --pool {{name|uuid}}
```

virsh pool-define-as

Create a configuration file in **`/etc/libvirt/storage`** for a persistent virtual machine storage pool from the provided arguments.

See also: **`virsh`**, **`virsh-pool-build`**, **`virsh-pool-start`**.

More information: https://manned.org/virsh.

- Create the configuration file for a storage pool called pool_name using `/var/vms` as the underlying storage system:

  ```
  virsh pool-define-as --name {{pool_name}} --type {{dir}} --target {{/var/vms}}
  ```

virsh pool-delete

Delete the underlying storage system of an inactive virtual machine storage pool.

See also: **`virsh`**, **`virsh-pool-destroy`**, **`virsh-pool-undefine`**.

More information: https://manned.org/virsh.

- Delete the underlying storage system for the storage pool specified by name or UUID (determine using `virsh pool-list`):

```
virsh pool-delete --pool {{name|uuid}}
```

virsh pool-destroy

Stop an active virtual machine storage pool.

See also: **`virsh`**, **`virsh-pool-delete`**.

More information: https://manned.org/virsh.

- Stop a storage pool specified by name or UUID (determine using `virsh pool-list`):

```
virsh pool-destroy --pool {{name|uuid}}
```

virsh pool-info

List information about a virtual machine storage pool.

See also: **`virsh`**.

More information: https://manned.org/virsh.

- List the name, UUID, state, persistence type, autostart status, capacity, space allocated, and space available for the storage pool specified by name or UUID (determine using `virsh pool-list`):

  ```
  virsh pool-info --pool {{name|uuid}}
  ```

virsh pool-list

List information about virtual machine storage pools.

See also: **`virsh`**, **`virsh-pool-autostart`**, **`virsh-pool-define-as`**.

More information: https://manned.org/virsh.

- List the name, state, and whether autostart is enabled or disabled for active storage pools:

```
virsh pool-list
```

- List information for active and inactive or just inactive storage pools:

```
virsh pool-list --{{all|inactive}}
```

- List extended information about persistence, capacity, allocation, and available space for active storage pools:

```
virsh pool-list --details
```

- List information for active storage pools with either autostart enabled or disabled:

```
virsh pool-list --{{autostart|no-autostart}}
```

- List information for active storage pools that are either persistent or transient:

```
virsh pool-list --{{persistent|transient}}
```

- List the name and UUID of active storage pools:

```
virsh pool-list --name --uuid
```

virsh pool-start

Start a previously configured but inactive virtual machine storage pool.

See also: **`virsh`**, **`virsh-pool-define-as`**, **`virsh-pool-destroy`**.

More information: https://manned.org/virsh.

- Start the storage pool specified by name or UUID (determine using `virsh pool-list`) and create the underlying storage system if it doesn't exist:

```
virsh pool-start --pool {{name|uuid}} --build
```

virsh pool-undefine

Delete the configuration file in **`/etc/libvirt/storage`** for a stopped virtual machine storage pool.

See also: **`virsh`**, **`virsh-pool-destroy`**.

More information: https://manned.org/virsh.

- Delete the configuration for the storage pool specified name or UUID (determine using `virsh pool-list`):

  ```
  virsh pool-undefine --pool {{name|uuid}}
  ```

virsh undefine

Delete a virtual machine.

More information: https://manned.org/virsh.

- Delete only the virtual machine configuration file:

```
virsh undefine --domain {{vm_name}}
```

- Delete the configuration file and all associated storage volumes:

```
virsh undefine --domain {{vm_name}} --remove-all-storage
```

- Delete the configuration file and the specified storage volumes using the target name or the source name (as obtained from the `virsh domblklist` command):

```
virsh undefine --domain {{vm_name}} --storage {{sda,sdb,path/to/source,...}}
```

virsh

Manage **`virsh`** guest domains. (Note: **`guest_id`** can be the ID, name or UUID of the guest).

Some subcommands such as **`list`** have their own usage documentation.

More information: https://libvirt.org/manpages/virsh.html.

- Connect to a hypervisor session:

```
virsh connect {{qemu:///system}}
```

- Activate a network named `default`:

```
sudo virsh net-start {{default}}
```

- List all domains:

```
virsh list --all
```

- Create a guest from a configuration file:

```
virsh create {{path/to/config_file.xml}}
```

- Edit a guest's configuration file (editor can be changed with $EDITOR):

```
virsh edit {{guest_id}}
```

- Start/reboot/shutdown/suspend/resume a guest:

```
virsh {{command}} {{guest_id}}
```

- Save the current state of a guest to a file:

```
virsh save {{guest_id}} {{filename}}
```

- Delete a running guest:

```
virsh destroy {{guest_id}} && virsh undefine {{guest_id}}
```

virt-clone

Clone a libvirt virtual machine.

More information: https://manned.org/virt-clone.

- Clone a virtual machine and automatically generate a new name, storage path, and MAC address:

```
virt-clone {{[-o|--original]}} {{vm_name}} --auto-clone
```

- Clone a virtual machine and specify the new name, storage path, and MAC address:

```
virt-clone {{[-o|--original]}} {{vm_name}} {{[-n|--name]}} {{new_vm_name}} {{[-f|--file]}} {{path/to/new_storage}} {{[-m|--mac]}} {{ff:ff:ff:ff:ff:ff|RANDOM}}
```

virt-install

Create virtual machines with libvirt and begin OS installation.

More information: https://manned.org/virt-install.

- Create a virtual machine with 1 GB RAM and 12 GB storage and start a Debian installation:

```
virt-install {{[-n|--name]}} {{vm_name}} --memory {{1024}} --disk path={{path/to/image.qcow2}},size={{12}} {{[-c|--cdrom]}} {{path/to/debian.iso}}
```

- Create a x86-64, KVM-accelerated, UEFI-based virtual machine with the Q35 chipset, 4 GiB RAM, 16 GiB RAW storage, and start a Fedora installation:

```
virt-install {{[-n|--name]}} {{vm_name}} --arch {{x86_64}} --virt-type {{kvm}} --machine {{q35}} --boot {{uefi}} --memory {{4096}} --disk path={{path/to/image.raw}},size={{16}} {{[-c|--cdrom]}} {{path/to/fedora.iso}}
```

- Create a diskless live virtual machine without an emulated sound device or a USB controller. Don't start an installation and don't autoconnect to console but attach a cdrom to it (might be useful for when using a live CD like tails):

```
virt-install {{[-n|--name]}} {{vm_name}} --memory {{512}} --disk {{none}} --controller {{type=usb,model=none}} --sound {{none}} --autoconsole {{none}} --install {{no_install=yes}} {{[-c|--cdrom]}} {{path/to/tails.iso}}
```

- Create a virtual machine with 16 GiB RAM, 250 GiB storage, 8 cores with hyperthreading, a specific CPU topology, and a CPU model that shares most features with the host CPU:

```
virt-install {{[-n|--name]}} {{vm_name}} --cpu {{host-model}},topology.sockets={{1}},topology.cores={{4}},topology.threads={
--memory {{16384}} --disk path={{path/to/image.qcow2}},size={{250}} {{[-c|--cdrom]}} {{path/to/debian.iso}}
```

- Create a virtual machine and kickstart an automated deployment based on Fedora 35 using only remote resources (no ISO required):

```
virt-install {{[-n|--name]}} {{vm_name}} --memory {{2048}} --disk path={{path/to/image.qcow2}},size={{20}} {{[-l|--location]}} {{https://download.fedoraproject.org/pub/fedora/linux/releases/35/Everything/x86_64/os/}} {{[-x|--extra-args]}} "{{inst.ks=https://path/to/valid_kickstart.org}}"
```

virt-qemu-run

Experimental tool to run a QEMU Guest VM independent of **`libvirtd`**.

More information: https://libvirt.org/manpages/virt-qemu-run.html.

- Run a QEMU virtual machine:

```
virt-qemu-run {{path/to/guest.xml}}
```

- Run a QEMU virtual machine and store the state in a specific directory:

```
virt-qemu-run {{[-r|--root]}} {{path/to/directory}} {{path/to/guest.xml}}
```

- Run a QEMU virtual machine and display verbose information about the startup:

```
virt-qemu-run {{[-v|--verbose]}} {{path/to/guest.xml}}
```

- Display help:

```
virt-qemu-run {{[-h|--help]}}
```

virt-sparsify

Make virtual machine drive images thin-provisioned.

Note: Use only for offline machines to avoid data corruption.

More information: https://manned.org/virt-sparsify.

- Create a sparsified compressed image without snapshots from an unsparsified one:

```
virt-sparsify --compress {{path/to/image.qcow2}} {{path/to/image_new.qcow2}}
```

- Sparsify an image in-place:

```
virt-sparsify --in-place {{path/to/image.img}}
```

virt-sysprep

Reset, unconfigure, or customize a virtual machine image.

More information: https://manned.org/virt-sysprep.

- List all supported operations (enabled operations are indicated with asterisks):

```
virt-sysprep --list-operations
```

- Remove sensitive system data from a virtual machine image:

```
sudo virt-sysprep {{[-a|--add]}} {{path/to/image.qcow2}}
```

- Specify a virtual machine by its name and run all enabled operations but don't actually apply the changes:

```
sudo virt-sysprep {{[-d|--domain]}} {{vm_name}} {{[-n|--dry-run]}}
```

- Run only the specified operations:

```
sudo virt-sysprep {{[-d|--domain]}} {{vm_name}} --operations {{operation1,operation2,...}}
```

- Generate a new `/etc/machine-id` file and enable customizations to be able to change the host name to avoid network conflicts:

```
sudo virt-sysprep {{[-d|--domain]}} {{vm_name}} --enable {{customizations}} --hostname {{host_name}} --operation {{machine-id}}
```

- Display help:

```
virt-sysprep {{[-he|--help]}}
```

virtualenv

Create virtual isolated Python environments.

More information: https://virtualenv.pypa.io/en/latest/cli_interface.html.

- Create a new environment:

```
virtualenv {{path/to/venv}}
```

- Customize the prompt prefix:

```
virtualenv --prompt {{prompt_prefix}} {{path/to/venv}}
```

- Use a different version of Python with virtualenv:

```
virtualenv {{[-p|--python]}} {{path/to/pythonbin}} {{path/to/venv}}
```

- Start (select) the environment:

```
source {{path/to/venv}}/bin/activate
```

- Stop the environment:

```
deactivate
```

virtualenvwrapper

Group of simple wrapper commands for Python's **`virtualenv`** tool.

More information: https://virtualenvwrapper.readthedocs.org.

- Create a new Python `virtualenv` in `$WORKON_HOME`:

```
mkvirtualenv {{virtualenv_name}}
```

- Create a `virtualenv` for a specific Python version:

```
mkvirtualenv --python {{/usr/local/bin/python3.8}}
{{virtualenv_name}}
```

- Activate or use a different `virtualenv`:

```
workon {{virtualenv_name}}
```

- Stop the `virtualenv`:

```
deactivate
```

- List all virtual environments:

```
lsvirtualenv
```

- Remove a `virtualenv`:

```
rmvirtualenv {{virtualenv_name}}
```

- Get summary of all virtualenvwrapper commands:

```
virtualenvwrapper
```

visudo

Safely edit the sudoers file.

More information: https://www.sudo.ws/docs/man/visudo.man.

- Edit the sudoers file:

```
sudo visudo
```

- Check the sudoers file for errors:

```
sudo visudo {{[-c|--check]}}
```

- Edit the sudoers file using a specific editor:

```
sudo EDITOR={{editor}} visudo
```

- Display version information:

```
visudo {{[-V|--version]}}
```

Vite

Create a Vite project.

Used to build JavaScript projects.

Available templates: vanilla, vanilla-ts, vue, vue-ts, react, react-ts, react-swc, react-swc-ts, preact, preact-ts, lit, lit-ts, svelte, svelte-ts.

More information: https://vitejs.dev/guide.

- Setup using `npm` 6.x:

```
npm create vite@latest my-react-app --template react-ts
```

- Setup using `npm` 7+, extra double-dash is needed:

```
npm create vite@latest my-react-app -- --template react-ts
```

- Setup using `yarn`:

```
yarn create vite my-react-app --template react-ts
```

- Setup using `pnpm`:

```
pnpm create vite my-react-app --template react-ts
```

vitest

Fast, modern testing framework built for Vite, offering seamless integration, TypeScript support, and a Jest-compatible API for unit, integration, and snapshot testing.

More information: https://vitest.dev/guide.

- Run all available tests:

  ```
  vitest run
  ```

- Run the test suites from the given files:

  ```
  vitest run {{path/to/file1 path/to/file2 ...}}
  ```

- Run the test suites from files within the current and subdirectories, whose paths match the given `regex`:

  ```
  vitest run {{regex1}} {{regex2}}
  ```

- Run the tests whose names match the given `regex`:

  ```
  vitest run --testNamePattern {{regex}}
  ```

- Watch files for changes and automatically re-run related tests:

  ```
  vitest
  ```

- Run tests with coverage:

  ```
  vitest run --coverage
  ```

- Run all tests but stops immediately after the first test failure:

  ```
  vitest run --bail=1
  ```

- Display help:

  ```
  vitest --help
  ```

viu

View images on the terminal.

More information: https://github.com/atanunq/viu.

- Render an image or animated GIF:

```
viu {{path/to/file}}
```

- Render an image or GIF from the internet using `curl`:

```
curl {{[-s|--silent]}} {{https://example.com/image.png}} |
viu -
```

- Render an image with a transparent background:

```
viu {{[-t|--transparent]}} {{path/to/file}}
```

- Render an image with a specific width and height in pixels:

```
viu {{[-w|--width]}} {{width}} {{[-h|--height]}} {{height}}
{{path/to/file}}
```

- Render an image or GIF and display its file name:

```
viu {{[-n|--name]}} {{path/to/file}}
```

vivaldi

This command is an alias of **`chromium`**.

More information: https://vivaldi.com.

- View documentation for the original command:

```
tldr chromium
```

vladimyr

Dario Vladović's personal CLI.

More information: https://github.com/vladimyr/vladimyr-cli.

- Start Dario's interactive CLI:

```
vladimyr
```

vlc

Cross-platform multimedia player.

See also: **`mpv`**, **`mplayer`**, **`ytfzf`**.

More information: https://wiki.videolan.org/Documentation:Command_line/.

- Play a file:

```
vlc {{path/to/file}}
```

- Play in fullscreen:

```
vlc --fullscreen {{path/to/file}}
```

- Play muted:

```
vlc --no-audio {{path/to/file}}
```

- Play repeatedly:

```
vlc --loop {{path/to/file}}
```

- Play video from a URL:

```
vlc {{https://www.youtube.com/watch?v=oHg5SJYRHA0}}
```

vmtouch

Manage the filesystem cache.

More information: https://manned.org/vmtouch.

- Print the cache status of a file:

```
vmtouch {{path/to/file}}
```

- Load a file into cache:

```
vmtouch -t {{path/to/file}}
```

- Evict a file from cache:

```
vmtouch -e {{path/to/file}}
```

- Lock a file in cache to prevent eviction from memory:

```
vmtouch -l {{path/to/file}}
```

- Lock a file and daemonize the program:

```
vmtouch -ld {{path/to/file}}
```

volta

A JavaScript Tool Manager that installs Node.js runtimes, npm and Yarn package managers, or any binaries from npm.

More information: https://volta.sh.

- List all installed tools:

```
volta list
```

- Install the latest version of a tool:

```
volta install {{node|npm|yarn|package_name}}
```

- Install a specific version of a tool:

```
volta install {{node|npm|yarn}}@version
```

- Choose a tool version for a project (will store it in `package.json`):

```
volta pin {{node|npm|yarn}}@version
```

- Display help:

```
volta help
```

- Display help for a subcommand:

```
volta help {{fetch|install|uninstall|pin|list|completions|
which|setup|run|help}}
```

vsce

Extension manager for Visual Studio Code.

More information: https://github.com/microsoft/vscode-vsce.

- List all the extensions created by a publisher:

  `vsce list {{publisher}}`

- Publish an extension as major, minor or patch version:

  `vsce publish {{major|minor|patch}}`

- Unpublish an extension:

  `vsce unpublish {{extension_id}}`

- Package the current working directory as a `.vsix` file:

  `vsce package`

- Show the metadata associated with an extension:

  `vsce show {{extension_id}}`

vt

Interface for VirusTotal.

API key from a VirusTotal account is required for this command.

More information: https://github.com/VirusTotal/vt-cli.

- Scan a specific file for viruses:

```
vt scan file {{path/to/file}}
```

- Scan a URL for viruses:

```
vt scan url {{url}}
```

- Display information from a specific analysis:

```
vt analysis {{file_id|analysis_id}}
```

- Download files in encrypted Zip format (requires premium account):

```
vt download {{file_id}} --output {{path/to/directory}} --zip
--zip-password {{password}}
```

- Initialize or re-initialize `vt` to enter API key interactively:

```
vt init
```

- Display information about a domain:

```
vt domain {{url}}
```

- Display information for a specific URL:

```
vt url {{url}}
```

- Display information for a specific IP address:

```
vt domain {{ip_address}}
```

vue build

A subcommand provided by **`@vue/cli`** and **`@vue/cli-service-global`** that enables quick prototyping.

More information: https://cli.vuejs.org/guide/cli-service.html#vue-cli-service-serve.

- Build a `.js` or `.vue` file in production mode with zero config:

```
vue build {{filename}}
```

vue init

Legacy project initialization subcommand of the Vue.js framework.

More information: https://cli.vuejs.org/guide/creating-a-project.html#pulling-2-x-templates-legacy.

- Create a new project using one of the default templates:

```
vue init {{webpack|webpack-simple|browserify|browserify-simple|simple}} {{project_name}}
```

- Create a new project using a local template:

```
vue init {{path/to/template_directory}} {{project_name}}
```

- Create a new project using a template from GitHub:

```
vue init {{username}}/{{repo}} {{project_name}}
```

vue serve

A subcommand provided by **`@vue/cli`** and **`@vue/cli-service-global`** that enables quick prototyping.

More information: https://cli.vuejs.org/guide/cli-service.html#vue-cli-service-serve.

- Serve a `.js` or `.vue` file in development mode with zero config:

```
vue serve {{filename}}
```

vue

Multi-purpose CLI for Vue.js.

Some subcommands such as **`build`** have their own usage documentation.

More information: https://cli.vuejs.org/guide/.

- Create a new Vue project interactively:

  ```
  vue create {{project_name}}
  ```

- Create a new project with web UI:

  ```
  vue ui
  ```

vulkaninfo

Print system Vulkan information.

More information: https://vulkan.lunarg.com/doc/view/latest/linux/vulkaninfo.html.

- Print full Vulkan information:

```
vulkaninfo
```

- Print a summary:

```
vulkaninfo --summary
```

- Make a HTML document of the full Vulkan information:

```
vulkaninfo --html
```

w

Display who is logged in and their processes.

More information: https://manned.org/w.

- Display information about all users who are currently logged in:

```
w
```

- Display information about a specific user:

```
w {{username}}
```

- Display logged-in user information without a header:

```
w {{[-h|--no-header]}}
```

- Display information without including the login, JCPU and PCPU columns:

```
w {{[-s|--short]}}
```

w3m

A text-based web browser.

Supports SSL and mouse input, even over SSH.

More information: https://w3m.sourceforge.net/MANUAL.

- Open a URL:

```
w3m {{https://example.com}}
```

- Open a URL in monochrome mode:

```
w3m {{https://example.com}} -M
```

- Open a URL without mouse support:

```
w3m {{https://example.com}} -no-mouse
```

- Open a new browser tab:

```
<Shift t>
```

- Display your browser history:

```
<Ctrl h>
```

- Quit w3m:

```
<q><y>
```

wafw00f

Identify and fingerprint Web Application Firewall (WAF) products protecting a website.

More information: https://github.com/EnableSecurity/wafw00f/wiki/Usage#arguments-list.

- Check if a website is using any WAF:

```
wafw00f {{https://www.example.com}}
```

- Test for all detectable WAFs without stopping at the first match:

```
wafw00f {{[-a|--findall]}} {{https://www.example.com}}
```

- Pass requests through a proxy (such as BurpSuite):

```
wafw00f {{[-p|--proxy]}} {{http://localhost:8080}} {{https://www.example.com}}
```

- Test for a specific WAF product (run `wafw00f --list` to get list of all supported WAFs):

```
wafw00f {{[-t|--test]}} {{Cloudflare|Cloudfront|Fastly|ZScaler|...}} {{https://www.example.com}}
```

- Pass custom headers from a file:

```
wafw00f {{[-H|--headers]}} {{path/to/headers.txt}} {{https://www.example.com}}
```

- Read target inputs from a file and show verbose output (multiple `v` for more verbosity):

```
wafw00f {{[-i|--input]}} {{path/to/urls.txt}} -{{vv}}
```

- List all WAFs that can be detected:

```
wafw00f {{[-l|--list]}}
```

waifu2x-ncnn-vulkan

Image upscaler for manga/anime-style images using NCNN neural network framework.

More information: https://github.com/nihui/waifu2x-ncnn-vulkan.

- Upscale an image:

```
waifu2x-ncnn-vulkan -i {{path/to/input_file}} -o {{path/to/output_file}}
```

- Upscale an image by a custom scale factor and denoise it:

```
waifu2x-ncnn-vulkan -i {{path/to/input_file}} -o {{path/to/output_file}} -s {{1|2|4|8|16|32}} -n {{-1|0|1|2|3}}
```

- Save the upscaled image in a specific format:

```
waifu2x-ncnn-vulkan -i {{path/to/input_file}} -o {{path/to/output_file}} -f {{jpg|png|webp}}
```

wait

Wait for a process to complete before proceeding.

See also: **`ps`** to view information about running processes and **`waitpid`**.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-wait.

- Wait for a process to finish given its process ID (PID) and return its exit status:

```
wait {{pid}}
```

- Wait for all processes known to the invoking shell to finish:

```
wait
```

- Wait for a job to finish (run `jobs` to find the job number):

```
wait %{{job_number}}
```

- Display help:

```
wait --help
```

waitress-serve

Pure Python WSGI HTTP Server.

More information: https://docs.pylonsproject.org/projects/waitress/en/latest/runner.html.

- Run a Python web app:

```
waitress-serve {{import.path:wsgi_func}}
```

- Listen on port 8080 on localhost:

```
waitress-serve --listen={{localhost}}:{{8080}}
{{import.path:wsgi_func}}
```

- Start waitress on a Unix socket:

```
waitress-serve --unix-socket={{path/to/socket}}
{{import.path:wsgi_func}}
```

- Use 4 threads to process requests:

```
waitress-serve --threads={{4}} {{import.path:wsgifunc}}
```

- Call a factory method that returns a WSGI object:

```
waitress-serve --call {{import.path.wsgi_factory}}
```

- Use the HTTPS URL scheme:

```
waitress-serve --url-scheme={{https}}
{{import.path:wsgi_func}}
```

wakeonlan

Send packets to wake-on-LAN (WOL) enabled PCs.

More information: https://github.com/jpoliv/wakeonlan.

- Send packets to all devices on the local network (255.255.255.255) by specifying a MAC address:

```
wakeonlan {{01:02:03:04:05:06}}
```

- Send packet to a specific device via IP address:

```
wakeonlan {{01:02:03:04:05:06}} {{[-i|--ip]}}
{{192.168.178.2}}
```

- Print the commands, but don't execute them (dry-run):

```
wakeonlan {{[-n|--dry-run]}} {{01:02:03:04:05:06}}
```

- Run in quiet mode:

```
wakeonlan {{[-q|--quiet]}} {{01:02:03:04:05:06}}
```

wal

Create color schemes based on the dominant colors of a wallpaper.

More information: https://github.com/dylanaraps/pywal/wiki/Getting-Started.

- Preview color scheme:

```
wal --preview {{image.png}}
```

- Create color scheme:

```
wal -i {{image.png}}
```

- Create a light color scheme:

```
wal -i {{image.png}} -l
```

- Skip setting the desktop wallpaper:

```
wal -i {{image.png}} -n
```

- Skip setting the terminal colors:

```
wal -i {{image.png}} -s
```

- Restore the previously generated color scheme and wallpaper:

```
wal -R
```

wapm

The WebAssembly package manager.

More information: https://wapm.io/help/reference.

- Interactively create a new `wapm.toml` file:

```
wapm init
```

- Download all the packages listed as dependencies in `wapm.toml`:

```
wapm install
```

- Download a specific version of a package and add it to the list of dependencies in `wapm.toml`:

```
wapm install {{package}}@{{version}}
```

- Download a package and install it globally:

```
wapm install --global {{package}}
```

- Uninstall a package and remove it from the list of dependencies in `wapm.toml`:

```
wapm uninstall {{package}}
```

- Print a tree of locally installed dependencies:

```
wapm list
```

- List top-level globally installed packages:

```
wapm list --global
```

- Execute a package command using the Wasmer runtime:

```
wapm run {{command_name}} {{arguments}}
```

warp-cli

Connect, disconnect and switch modes of a connection to Cloudflare's WARP service.

WARP is a VPN that encrypts traffic for privacy, security, and speed.

See also: **`fastd`**, **`ivpn`**, **`mozillavpn`**, **`mullvad`**.

More information: https://developers.cloudflare.com/warp-client/.

- Register the current device to WARP (must be run before first connection):

```
warp-cli registration new
```

- Connect to WARP:

```
warp-cli connect
```

- Disconnect from WARP:

```
warp-cli disconnect
```

- Display the WARP connection status:

```
warp-cli status
```

- Switch to a specific mode:

```
warp-cli set-mode {{mode}}
```

- Display help:

```
warp-cli help
```

- Display help for a subcommand:

```
warp-cli help {{subcommand}}
```

warp-diag

Diagnostic and feedback tool for Cloudflare's WARP service.

See also: **`warp-cli`**.

More information: https://developers.cloudflare.com/warp-client/.

- Generate a Zip file with information about the system configuration and the WARP connection:

```
warp-diag
```

- Generate a Zip file with debug information including a timestamp to the output filename:

```
warp-diag --add-ts
```

- Save the output file under a specific directory:

```
warp-diag --output {{path/to/directory}}
```

- Submit a new feedback to Cloudflare's WARP interactively:

```
warp-diag feedback
```

wasm-objdump

Display information from WebAssembly binaries.

More information: https://github.com/WebAssembly/wabt.

- Display the section headers of a given binary:

```
wasm-objdump {{[-h|--headers]}} {{file.wasm}}
```

- Display the entire disassembled output of a given binary:

```
wasm-objdump {{[-d|--disassemble]}} {{file.wasm}}
```

- Display the details of each section:

```
wasm-objdump {{[-x|--details]}} {{file.wasm}}
```

- Display the details of a given section:

```
wasm-objdump {{[-j|--section]}} '{{import}}' {{[-x|--
details]}} {{file.wasm}}
```

wasm-opt

Optimize WebAssembly binary files.

More information: https://github.com/webassembly/binaryen.

- Apply default optimizations and write to a given file:

```
wasm-opt -O {{input.wasm}} {{[-o|--output]}} {{output.wasm}}
```

- Apply all optimizations and write to a given file (takes more time, but generates optimal code):

```
wasm-opt -O4 {{input.wasm}} {{[-o|--output]}} {{output.wasm}}
```

- Optimize a file for size:

```
wasm-opt -Oz {{input.wasm}} {{[-o|--output]}} {{output.wasm}}
```

- Print the textual representation of the binary to console:

```
wasm-opt {{input.wasm}} --print
```

wasm2c

Convert a file from the WebAssembly binary format to a C source file and header.

More information: https://github.com/WebAssembly/wabt.

- Convert a file to a C source file and header and display it to the console:

```
wasm2c {{file.wasm}}
```

- Write the output to a given file (`file.h` gets additionally generated):

```
wasm2c {{file.wasm}} {{[-o|--output]}} {{file.c}}
```

wasm2wat

Convert a file from the WebAssembly binary format to the text format.

More information: https://github.com/WebAssembly/wabt.

- Convert a file to the text format and display it to the console:

```
wasm2wat {{file.wasm}}
```

- Write the output to a given file:

```
wasm2wat {{file.wasm}} {{[-o|--output]}} {{file.wat}}
```

wat2wasm

Convert a file from the WebAssembly text format to the binary format.

More information: https://github.com/WebAssembly/wabt.

- Parse and check a file for errors:

```
wat2wasm {{file.wat}}
```

- Write the output binary to a given file:

```
wat2wasm {{file.wat}} {{[-o|--output]}} {{file.wasm}}
```

- Display simplified representation of every byte:

```
wat2wasm {{[-v|--verbose]}} {{file.wat}}
```

watch

Execute a program periodically and monitor the output in full-screen mode.

More information: https://manned.org/watch.

- Repeatedly run a command and show the result:

```
watch {{command}}
```

- Re-run a command every 60 seconds:

```
watch {{[-n|--interval]}} 60 {{command}}
```

- Monitor disk space, highlighting differences as they appear:

```
watch {{[-d|--differences]}} df
```

- Repeatedly run a pipeline and show the result:

```
watch "{{command1}} | {{command2}} | {{command3}}"
```

- Exit `watch` if the visible output changes:

```
watch {{[-g|--chgexit]}} {{lsblk}}
```

- Interpret terminal control characters:

```
watch {{[-c|--color]}} {{ls --color=always}}
```

watchexec

Run arbitrary commands when files change.

More information: https://manned.org/watchexec.

- Call `ls -la` when any file in the current directory changes:

```
watchexec {{ls -la}}
```

- Run `make` when any JavaScript, CSS and HTML file in the current directory changes:

```
watchexec {{[-e|--exts]}} {{js,css,html}} make
```

- Run `make` when any file in the `lib` or `src` directory changes:

```
watchexec {{[-w|--watch]}} {{lib}} {{[-w|--watch]}} {{src}} {{make}}
```

- Call/restart `my_server` when any file in the current directory changes, sending `SIGKILL` to stop the child process:

```
watchexec {{[-r|--restart]}} --stop-signal {{SIGKILL}} {{my_server}}
```

- Restart the execution of a command when any Java source file in the current directory changes, sending `SIGKILL` and only checking for updates every `n`ms:

```
watchexec {{[-r|--restart]}} --stop-signal {{SIGKILL}} --poll {{10000}} {{[-e|--exts]}} {{java}} {{command}}
```

watchman

A service that watches files, and triggers actions when changes occur.

More information: https://facebook.github.io/watchman/docs/cli-options.

- Infer the root directory of the project containing the specified directory, and watch its files and sub-folders for changes:

```
watchman watch-project {{path/to/directory}}
```

- Add a trigger to run a command when files with a specified filename pattern in a watched directory change:

```
watchman -- trigger {{path/to/watched_directory}} {{trigger_name}} '{{pattern}}' -- {{command}}
```

- List all watched directories:

```
watchman watch-list
```

- Delete a watch on a directory and its associated triggers:

```
watchman watch-del {{path/to/watched_directory}}
```

- Delete all watched directories and triggers:

```
watchman watch-del-all
```

- List all triggers on a watched directory:

```
watchman trigger-list {{path/to/watched_directory}}
```

- Delete a trigger from a watched directory:

```
watchman trigger-del {{path/to/watched_directory}} {{trigger_name}}
```

- Temporarily stop `watchman`, until the next time you call a `watchman` command:

```
watchman shutdown-server
```

watson

A wonderful CLI to track your time.

More information: https://github.com/TailorDev/Watson.

- Start monitoring time in project:

```
watson start {{project}}
```

- Start monitoring time in project with tags:

```
watson start {{project}} +{{tag}}
```

- Stop monitoring time for the current project:

```
watson stop
```

- Display the latest working sessions:

```
watson log
```

- Edit most recent frame:

```
watson edit
```

- Remove most recent frame:

```
watson remove
```

waybar

Highly customizable Wayland bar for Sway and Wlroots based compositors.

More information: https://github.com/Alexays/Waybar.

- Start `waybar` with the default configuration and stylesheet:

```
waybar
```

- Use a different configuration file:

```
waybar {{[-c|--config]}} {{path/to/config.jsonc}}
```

- Use a different stylesheet file:

```
waybar {{[-s|--style]}} {{path/to/stylesheet.css}}
```

- Set the logging level:

```
waybar {{[-l|--log-level]}} {{trace|debug|info|warning|error|critical|off}}
```

waymore

Fetch URLs of a domain from Wayback Machine, Common Crawl, Alien Vault OTX, URLScan, and VirusTotal.

Note: Unless specified, output is dumped into the **`results/`** directory where waymore's **`config.yml`** resides (by default in **`~/.config/waymore/`**).

More information: https://github.com/xnl-h4ck3r/waymore.

- Search for URLs of a domain (output will typically be in `~/.config/waymore/results/`):

```
waymore {{[-i|--input]}} {{example.com}}
```

- Limit search results to only include a list of URLs for a domain and store outputs to the specified file:

```
waymore -mode U {{[-oU|--output-urls]}} {{path/to/example.com-urls.txt}} {{[-i|--input]}} {{example.com}}
```

- Only output the content bodies of URLs and store outputs to the specified directory:

```
waymore -mode R {{[-oR|--output-responses]}} {{path/to/example.com-url-responses}} {{[-i|--input]}} {{example.com}}
```

- Filter the results by specifying date ranges:

```
waymore -from {{YYYYMMDD|YYYYMM|YYYY}} {{[-to|--to-date]}} {{YYYYMMDD|YYYYMM|YYYY}} {{[-i|--input]}} {{example.com}}
```

wbmptopbm

Convert a wireless bitmap file to a PBM image.

More information: https://netpbm.sourceforge.net/doc/wbmptopbm.html.

- Convert a WBMP file to a PBM image:

```
wbmptopbm {{path/to/input_file.wbpm}} > {{path/to/
output_file.pbm}}
```

wc

Count lines, words, and bytes.

More information: https://www.gnu.org/software/coreutils/manual/html_node/wc-invocation.html.

- Count all lines in a file:

```
wc {{[-l|--lines]}} {{path/to/file}}
```

- Count all words in a file:

```
wc {{[-w|--words]}} {{path/to/file}}
```

- Count all bytes in a file:

```
wc {{[-c|--bytes]}} {{path/to/file}}
```

- Count all characters in a file (taking multi-byte characters into account):

```
wc {{[-m|--chars]}} {{path/to/file}}
```

- Count all lines, words and bytes from `stdin`:

```
{{find .}} | wc
```

- Count the length of the longest line in number of characters:

```
wc {{[-L|--max-line-length]}} {{path/to/file}}
```

wcurl

A simple wrapper around **`curl`** to easily download files.

See also: **`curl`**.

More information: https://curl.se/wcurl/manual.html.

- Download the contents of a URL to a file indicated by the URL ("foo" in this case):

```
wcurl {{https://example.com/foo}}
```

- Download the contents of a URL to a file with a specified name:

```
wcurl {{[-o|--output]}} {{bar}} {{https://example.com/foo}}
```

- Download the contents of a URL, enabling progress bar and defaulting to HTTP/2:

```
wcurl --curl-options "--progress-bar --http2" {{https://example.com/foo}}
```

- Resume from an interrupted download:

```
wcurl --curl-options "--clobber --continue-at -" {{https://example.com/foo}}
```

wdiff

Display word differences between text files.

More information: https://www.gnu.org/software/wdiff/manual/wdiff.html#wdiff-invocation.

- Compare two files:

```
wdiff {{path/to/file1}} {{path/to/file2}}
```

- Ignore case when comparing:

```
wdiff {{[-i|--ignore-case]}} {{path/to/file1}} {{path/to/file2}}
```

- Display how many words are deleted, inserted or replaced:

```
wdiff {{[-s|--statistics]}} {{path/to/file1}} {{path/to/file2}}
```

weasyprint

Render HTML to PDF or PNG.

More information: https://doc.courtbouillon.org/weasyprint/stable/api_reference.html#command-line-api.

- Render an HTML file to PDF:

```
weasyprint {{path/to/input.html}} {{path/to/output.pdf}}
```

- Render an HTML file to PNG, including an additional user stylesheet:

```
weasyprint {{path/to/input.html}} {{path/to/output.png}} --stylesheet {{path/to/stylesheet.css}}
```

- Output additional debugging information when rendering:

```
weasyprint {{path/to/input.html}} {{path/to/output.pdf}} --verbose
```

- Specify a custom resolution when outputting to PNG:

```
weasyprint {{path/to/input.html}} {{path/to/output.png}} --resolution {{300}}
```

- Specify a base URL for relative URLs in the input HTML file:

```
weasyprint {{path/to/input.html}} {{path/to/output.png}} --base-url {{url_or_filename}}
```

web-ext

Manage web extension development.

More information: https://github.com/mozilla/web-ext.

- Run the web extension in the current directory in Firefox:

```
web-ext run
```

- Run a web extension from a specific directory in Firefox:

```
web-ext run --source-dir {{path/to/directory}}
```

- Display verbose execution output:

```
web-ext run --verbose
```

- Run a web extension in Firefox Android:

```
web-ext run --target firefox-android
```

- Lint the manifest and source files for errors:

```
web-ext lint
```

- Build and package the extension:

```
web-ext build
```

- Display verbose build output:

```
web-ext build --verbose
```

- Sign a package for self-hosting:

```
web-ext sign --api-key {{api_key}} --api-secret
{{api_secret}}
```

webpack

Bundle a web project's js files and other assets into a single output file.

More information: https://webpack.js.org.

- Create a single output file from an entry point file:

```
webpack {{app.js}} {{bundle.js}}
```

- Load CSS files too from the JavaScript file (this uses the CSS loader for CSS files):

```
webpack {{app.js}} {{bundle.js}} --module-bind '{{css=css}}'
```

- Pass a configuration file (with e.g. the entry script and the output filename) and show compilation progress:

```
webpack --config {{webpack.config.js}} --progress
```

- Automatically recompile on changes to project files:

```
webpack --watch {{app.js}} {{bundle.js}}
```

webpmux

Create WebP animations.

More information: https://developers.google.com/speed/webp/docs/webpmux.

- Create a two-frame animation:

```
webpmux -frame {{path/to/frame1.webp}} +{{500}} -frame
{{path/to/frame2.webp}} +{{500}} -loop {{0}} -o {{path/to/
output.webp}}
```

webstorm

The JetBrains JavaScript IDE.

More information: https://www.jetbrains.com/help/webstorm/working-with-the-ide-features-from-command-line.html.

- Open the current directory in WebStorm:

```
webstorm
```

- Open a specific directory in WebStorm:

```
webstorm {{path/to/directory}}
```

- Open specific files in the LightEdit mode:

```
webstorm -e {{path/to/file1 path/to/file2 ...}}
```

- Open and wait until done editing a specific file in the LightEdit mode:

```
webstorm --wait -e {{path/to/file}}
```

- Open a file with the cursor at the specific line:

```
webstorm --line {{line_number}} {{path/to/file}}
```

- Open and compare files (supports up to 3 files):

```
webstorm diff {{path/to/file1 path/to/file2 path/to/
optional_file3}}
```

- Open and perform a three-way merge:

```
webstorm merge {{path/to/left_file}} {{path/to/right_file}}
{{path/to/target_file}}
```

webtorrent

The interface for WebTorrent.

Supports magnets, URLs, info hashes and **`.torrent`** files.

More information: https://github.com/webtorrent/webtorrent-cli.

- Download a torrent:

```
webtorrent download "{{torrent_id}}"
```

- Stream a torrent to VLC media player:

```
webtorrent download "{{torrent_id}}" --vlc
```

- Stream a torrent to a Digital Living Network Alliance (DLNA) device:

```
webtorrent download "{{torrent_id}}" --dlna
```

- Display a list of files for a specific torrent:

```
webtorrent download "{{torrent_id}}" --select
```

- Specify a file index from the torrent to download:

```
webtorrent download "{{torrent_id}}" --select {{index}}
```

- Seed a specific file or directory:

```
webtorrent seed {{path/to/file_or_directory}}
```

- Create a new torrent file for the specified file path:

```
webtorrent create {{path/to/file}}
```

- Display information for a magnet URI or `.torrent` file:

```
webtorrent info {{path/to/file_or_magnet}}
```

weechat

Internet relay chat client with various features.

More information: https://weechat.org/doc.

- Start WeeChat:

```
weechat
```

- Do not load any plugin(s) on startup:

```
weechat --no-plugin
```

- Do not load any script(s) on startup:

```
weechat --no-script
```

- Do not connect to servers automatically:

```
weechat --no-connect
```

- Write default terminal colors to `stdout`:

```
weechat --colors
```

wezterm cli

Interact with a running Wezterm GUI or multiplexer.

More information: https://wezterm.org/cli/cli/index.html.

- List windows, tabs, and panes:

```
wezterm cli list
```

- Split the current pane and print the new pane's ID to `stdout`:

```
wezterm cli split-pane --{{left|right|top|bottom}} --{{cells|percent}} {{n}}
```

- Activate (focus) a pane:

```
wezterm cli activate-pane --pane-id {{id}}
```

- Kill a pane:

```
wezterm cli kill-pane --pane-id {{id}}
```

wezterm

Wez's Terminal Emulator - a powerful cross-platform terminal emulator and multiplexer.

Some subcommands such as **`cli`** have their own usage documentation.

More information: https://wezterm.org/cli/general.

- Start a new Wezterm process and create a window:

```
wezterm
```

- Establish an `ssh` session in a new Wezterm window:

```
wezterm ssh {{user}}@{{host}}:{{port}}
```

- Connect to the multiplexer (`wezterm-mux-server`):

```
wezterm connect {{domain_name}}
```

- Output an image to the terminal:

```
wezterm imgcat {{path/to/image}}
```

- Record a terminal session as an asciicast (by default recordings are saved to `/tmp`):

```
wezterm record
```

- Replay an asciicast terminal session:

```
wezterm replay {{path/to/cast_file}}
```

- Specify the configuration file to use (overrides the normal configuration file resolution):

```
wezterm --config-file {{path/to/config_file}}
```

- Display help:

```
wezterm help
```

wfuzz

A web application bruteforcer.

More information: https://wfuzz.readthedocs.io/en/latest/user/basicusage.html.

- Directory and file bruteforce using the specified [w]ordlist and also [p]roxying the traffic:

```
wfuzz -w {{path/to/file}} -p {{127.0.0.1:8080:HTTP}} {{http://example.com/FUZZ}}
```

- Save the results to a [f]ile:

```
wfuzz -w {{path/to/file}} -f {{filename}} {{http://example.com/FUZZ}}
```

- Show [c]olorized output while only showing the declared response codes in the output:

```
wfuzz -c -w {{path/to/file}} --sc {{200,301,302}} {{http://example.com/FUZZ}}
```

- Use a custom [H]eader to fuzz subdomains while [h]iding specific response [c]odes and word counts. Increase the [t]hreads to 100 and include the target ip/domain:

```
wfuzz -w {{path/to/file}} -H "{{Host: FUZZ.example.com}}" --hc {{301}} --hw {{222}} -t {{100}} {{example.com}}
```

- Brute force Basic Authentication using a list of usernames and passwords from files for each FUZ[z] keyword, [h]iding response [c]odes of unsuccessful attempts:

```
wfuzz -c --hc {{401}} -s {{delay_between_requests_in_seconds}} -z file,{{path/to/usernames}} -z file,{{path/to/passwords}} --basic 'FUZZ:FUZ2Z' {{https://example.com}}
```

- Provide wordlist directly from the command-line and use POST request for fuzzing:

```
wfuzz -z list,{{word1-word2-...}} {{https://api.example.com}} -d "{{id=FUZZ&showwallet=true}}"
```

- Provide wordlists from a file applying base64 and md5 encoding on them (`wfuzz -e encoders` lists all available encoders):

```
wfuzz -z file,{{path/to/file}},none-base64-md5 {{https://
example.com/FUZZ}}
```

- List available encoders/payloads/iterators/printers/scripts:

```
wfuzz -e {{encoders|payloads|iterators|printers|scripts}}
```

wget

Download files from the Web.

Supports HTTP, HTTPS, and FTP.

See also: **`curl`**.

More information: https://www.gnu.org/software/wget/manual/wget.html.

- Download the contents of a URL to a file (named "foo" in this case):

```
wget {{https://example.com/foo}}
```

- Download the contents of a URL to a file (named "bar" in this case):

```
wget {{[-O|--output-document]}} {{bar}} {{https://
example.com/foo}}
```

- Download a single web page and all its resources with 3-second intervals between requests (scripts, stylesheets, images, etc.):

```
wget {{[-pkw|--page-requisites --convert-links --wait]}} 3
{{https://example.com/some_page.html}}
```

- Download all listed files within a directory and its sub-directories (does not download embedded page elements):

```
wget {{[-mnp|--mirror --no-parent]}} {{https://example.com/
some_path/}}
```

- Limit the download speed and the number of connection retries:

```
wget --limit-rate {{300k}} {{[-t|--tries]}} {{100}}
{{https://example.com/some_path/}}
```

- Download a file from an HTTP server using Basic Auth (also works for FTP):

```
wget --user {{username}} --password {{password}} {{https://
example.com}}
```

- Continue an incomplete download:

```
wget {{[-c|--continue]}} {{https://example.com}}
```

- Download all URLs stored in a text file to a specific directory:

```
wget {{[-P|--directory-prefix]}} {{path/to/directory}} {{[-
i|--input-file]}} {{path/to/URLs.txt}}
```

wget2

An improved version of **`wget`** for downloading files from the web.

Supports HTTP, HTTPS, and HTTP/2 protocols with enhanced performance.

By default, **`wget2`** uses multiple threads for faster downloads.

More information: https://manned.org/wget2.

- Download the contents of a URL to a file using multiple threads (default behavior differs from `wget`):

```
wget2 {{https://example.com/resource}}
```

- Limit the number of threads used for downloading (default is 5 threads):

```
wget2 --max-threads {{10}} {{https://example.com/resource}}
```

- Download a single web page and all its resources (scripts, stylesheets, images, etc.):

```
wget2 {{[-p|--page-requisites]}} {{[-k|--convert-links]}} {{https://example.com/somepage.html}}
```

- Mirror a website, but do not ascend to the parent directory (does not download embedded page elements):

```
wget2 {{[-m|--mirror]}} {{[-np|--no-parent]}} {{https://example.com/somepath/}}
```

- Limit the download speed and the number of connection retries:

```
wget2 --limit-rate {{300k}} {{[-t|--tries]}} {{100}} {{https://example.com/somepath/}}
```

- Continue an incomplete download (behavior is consistent with `wget`):

```
wget2 {{[-c|--continue]}} {{https://example.com}}
```

- Download all URLs stored in a text file to a specific directory:

```
wget2 {{[-P|--directory-prefix]}} {{path/to/directory}} {{[-i|--input-file]}} {{URLs.txt}}
```

- Download a file from an HTTP server using Basic Auth (also works for HTTPS):

```
wget2 --user {{username}} --password {{password}} {{https://example.com}}
```

whatwaf

Detect and bypass web application firewalls and protection systems.

More information: https://github.com/Ekultek/WhatWaf.

- Detect protection on a single URL, optionally use verbose output:

```
whatwaf {{[-u|--url]}} {{https://example.com}} --verbose
```

- Detect protection on a list of URLs in parallel from a file (one URL per line):

```
whatwaf {{[-t|--threads]}} {{number}} {{[-l|--list]}} {{path/to/file}}
```

- Send requests through a proxy and use custom payload list from a file (one payload per line):

```
whatwaf --proxy {{http://127.0.0.1:8080}} --pl {{path/to/file}} {{[-u|--url]}} {{https://example.com}}
```

- Send requests through Tor (Tor must be installed) using custom payloads (comma-separated):

```
whatwaf --tor {{[-p|--payloads]}} '{{payload1,payload2,...}}' {{[-u|--url]}} {{https://example.com}}
```

- Use a random user-agent, set throttling and timeout, send a POST request, and force HTTPS connection:

```
whatwaf --ra --throttle {{seconds}} --timeout {{seconds}} {{[-P|--post]}} --force-ssl {{[-u|--url]}} {{http://example.com}}
```

- List all WAFs that can be detected:

```
whatwaf --wafs
```

- List all available tamper scripts:

```
whatwaf --tampers
```

whatweb

Next-generation web scanner.

More information: https://github.com/urbanadventurer/WhatWeb/.

- Scan websites/targets for web technologies:

```
whatweb {{website1 website2 ...}}
```

- Read targets/websites from a file:

```
whatweb {{[-i|--input-file]}} {{targets_file}}
```

- Scan a website/target in verbose mode:

```
whatweb {{[-v|--verbose]}} {{example.com}}
```

- Run an aggressive scan on a website:

```
whatweb {{[-a|--aggression]}} 3 {{example.com}}
```

- Scan a network and suppress errors:

```
whatweb --no-errors {{192.168.0.0/24}}
```

- List plugins:

```
whatweb {{[-l|--list-plugins]}}
```

- List plugin details:

```
whatweb {{[-I|--info-plugins]}} {{plugin_name}}
```

whence

A Zsh builtin to indicate how a command would be interpreted.

More information: https://zsh.sourceforge.io/Doc/Release/Shell-Builtin-Commands.html#index-whence.

- Interpret `command`, with expansion if defined as an `alias` (similar to the `command -v` builtin):

```
whence "{{command}}"
```

- Display type of `command`, with location if defined as a function, or binary (equivalent to the `type` and `command -V` builtins):

```
whence -v "{{command}}"
```

- Same as above, except display content of shell functions instead of location (equivalent to `which` builtin):

```
whence -c "{{command}}"
```

- Same as above, but show all occurrences on command path (equivalent to the `where` builtin):

```
whence -ca "{{command}}"
```

- Search only the `PATH` for `command`, ignoring builtins, aliases or shell functions (equivalent to the `where` command):

```
whence -p "{{command}}"
```

where

Reports all known instances of a command.

It could be an executable in the PATH environment variable, an alias, or a shell builtin.

More information: https://zsh.sourceforge.io/Doc/Release/Shell-Builtin-Commands.html.

- Find all instances of a command:

```
where {{command}}
```

which

Locate a program in the user's path.

See also: **`whereis`**, **`whatis`**, **`type`**.

More information: https://manned.org/which.

- Search the PATH environment variable and display the location of any matching executables:

```
which {{executable}}
```

- If there are multiple executables which match, display all:

```
which {{[-a|--all]}} {{executable}}
```

while

Simple shell loop that repeats while the return value remains zero.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-while.

- Read `stdin` and perform an action on every line:

```
while read line; do {{echo "$line"}}; done
```

- Execute a command forever once every second:

```
while :; do {{command}}; sleep 1; done
```

- Execute a command until it fails:

```
while {{command}}; do :; done
```

whisper

Convert audio files to **`txt`**, **`vtt`**, **`srt`**, **`tsv`** and **`json`**.

More information: https://github.com/openai/whisper.

- Convert a specific audio file to all of the given file formats:

```
whisper {{path/to/audio.mp3}}
```

- Convert an audio file specifying the output format of the converted file:

```
whisper {{path/to/audio.mp3}} --output_format {{txt}}
```

- Convert an audio file using a specific model for conversion:

```
whisper {{path/to/audio.mp3}} --model
{{tiny.en,tiny,base.en,base,small.en,small,medium.en,medium,large-
v1,large-v2,large}}
```

- Convert an audio file specifying which language the audio file is in to reduce conversion time:

```
whisper {{path/to/audio.mp3}} --language {{english}}
```

- Convert an audio file and save it to a specific location:

```
whisper {{path/to/audio.mp3}} --output_dir "{{path/to/
output}}"
```

- Convert an audio file in quiet mode:

```
whisper {{path/to/audio.mp3}} --verbose {{False}}
```

who

Display who is logged in and related data (processes, boot time).

See also: **`whoami`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/who-invocation.html.

- Display the username, line, and time of all currently logged-in sessions:

```
who
```

- Display all available information:

```
who {{[-a|--all]}}
```

- Display all available information with table headers:

```
who {{[-aH|--all --heading]}}
```

whoami

Print the username associated with the current effective user ID.

More information: https://www.gnu.org/software/coreutils/manual/html_node/whoami-invocation.html.

- Display currently logged username:

```
whoami
```

- Display the username after a change in the user ID:

```
sudo whoami
```

whois

Client for the WHOIS (RFC 3912) protocol.

More information: https://manned.org/whois.

- Get information about a domain name:

```
whois {{example.com}}
```

- Get information about an IP address:

```
whois {{8.8.8.8}}
```

- Get abuse contact for an IP address:

```
whois -b {{8.8.8.8}}
```

wiggle

A patch application tool resolving conflicts in patches that **`patch`** cannot handle.

Note: Wiggle forcefully applies all changes, merging when conflicts arise, and reporting unresolvable issues.

More information: https://manned.org/wiggle.

- Apply changes from the patch file to the original file:

```
wiggle {{path/to/my_patch.patch}}
```

- Apply changes to the output file:

```
wiggle {{path/to/my_patch.patch}} {{[-o|--output]}} {{path/to/output_file.txt}}
```

- Take any changes in `file.rej` that could not have been applied and merge them into a file:

```
wiggle {{[-r|--replace]}} {{path/to/file}} {{path/to/file.rej}}
```

- Extract one branch of a patch or merge file:

```
wiggle {{[-x|--extract]}} {{path/to/my_patch.patch}}
```

- Apply a patch and save the compared words to the output file:

```
wiggle {{[-w|--words]}} {{path/to/my_word_patch.patch}} {{[-o|--output]}} {{path/to/word_patched_code.c}}
```

- Display help about the merge function:

```
wiggle {{[-m|--merge]}} {{[-h|--help]}}
```

wikiman

Offline search engine for documentation.

Supports manual pages, Arch Wiki, Gentoo Wiki, FreeBSD documentation, and tldr-pages.

More information: https://github.com/filiparag/wikiman.

- Search for a specific topic in all installed sources:

```
wikiman {{search_term}}
```

- Search for a topic in a specific [s]ource:

```
wikiman -s {{source}} {{search_term}}
```

- Search for a topic in two or more specific [s]ources:

```
wikiman -s {{source1,source2,...}} {{search_term}}
```

- List existing [S]ources:

```
wikiman -S
```

- Display [h]elp:

```
wikiman -h
```

wikit

Easily get Wikipedia summaries.

More information: https://github.com/KorySchneider/wikit.

- Show a short summary of a specific topic on Wikipedia:

```
wikit {{topic}}
```

- Specify a language (ISO 639-1 language code):

```
wikit {{topic}} {{[-l|--lang]}} {{language_code}}
```

- Open the full Wikipedia article in the default browser:

```
wikit {{topic}} -b
```

- Open a disambiguation menu:

```
wikit {{topic}} -d
```

winicontopam

Convert a Windows ICO file to a PAM file.

More information: https://netpbm.sourceforge.net/doc/winicontopam.html.

- Read an ICO file and convert the best quality image contained therein to the PAM format:

```
winicontopam {{path/to/input_file.ico}} > {{path/to/
output.pam}}
```

- Convert all images in the input file to PAM:

```
winicontopam {{[-al|-allimages]}} {{path/to/input_file.ico}}
> {{path/to/output.pam}}
```

- Convert the n'th image in the input file to PAM:

```
winicontopam {{[-i|-image]}} {{n}} {{path/to/input_file.ico}}
> {{path/to/output.pam}}
```

- If the image(s) to be extracted contain graded transparency data and an AND mask, write the AND mask into the fifth channel of the output PAM file:

```
winicontopam {{[-an|-andmasks]}} {{path/to/input_file.ico}} >
{{path/to/output.pam}}
```

winicontoppm

This command has been superseded by **`winicontopam`**.

More information: https://netpbm.sourceforge.net/doc/winicontoppm.html.

- View documentation for the current command:

```
tldr winicontopam
```

wipeclean

Clear the terminal screen using an animated wiper.

More information: https://github.com/JeanJouliaCode/wipeClean.

- Clear the terminal screen:

```
wipeclean
```

- Set the animation speed in frames per second (defaults to 150):

```
wipeclean --speed {{speed}}
```

wishlist

SSH directory and multiplexer.

Acts as a single entry point for connecting to SSH servers or Wish applications.

More information: https://github.com/charmbracelet/wishlist.

- Explore SSH servers listed in your `~/.ssh/config` file (local mode):

```
wishlist
```

- Start Wishlist in server mode to provide remote access:

```
wishlist {{[s|serve]}}
```

- Use a custom configuration file:

```
wishlist {{[-c|--config]}} {{path/to/config.yaml}}
```

- Discover SSH endpoints using Zeroconf (mDNS/Bonjour):

```
wishlist --zeroconf.enabled
```

- Discover SSH nodes from DNS SRV records:

```
wishlist --srv.domain {{example.com}}
```

- Discover SSH nodes from a Tailscale tailnet:

```
wishlist --tailscale.net={{tailnet_name}} --
tailscale.key={{tskey-api-abc123}}
```

wkhtmltopdf

Convert HTML documents or web pages into PDF files.

More information: https://wkhtmltopdf.org/usage/wkhtmltopdf.txt.

- Convert a HTML document into PDF:

```
wkhtmltopdf {{input.html}} {{output.pdf}}
```

- Specify the PDF page size (please see `PaperSize` of `QPrinter` for supported sizes):

```
wkhtmltopdf --page-size {{A4}} {{input.html}} {{output.pdf}}
```

- Set the PDF page margins:

```
wkhtmltopdf --margin-{{top|bottom|left|right}} {{10mm}}
{{input.html}} {{output.pdf}}
```

- Set the PDF page orientation:

```
wkhtmltopdf --orientation {{Landscape|Portrait}}
{{input.html}} {{output.pdf}}
```

- Generate a greyscale version of the PDF document:

```
wkhtmltopdf --grayscale {{input.html}} {{output.pdf}}
```

wlc

Manage localization projects on a Weblate instance.

More information: https://docs.weblate.org/en/latest/wlc.html#commands.

- List projects using a configuration file:

```
wlc {{[-c|--config]}} {{path/to/file}} list-projects
```

- List components in a project, and override API URL and API key:

```
wlc {{[-u|--url]}} {{URL}} {{[-k|--key]}} {{key}} ls
{{project}}
```

- List translations from a component in a specific format:

```
wlc {{[-f|--format]}} {{text|csv|json|html}} ls {{project}}/
{{component}}
```

- Print statistics for a project:

```
wlc stats {{project}}
```

- Display help:

```
wlc {{[-h|--help]}}
```

wlsunset

Adjust the color temperature in Wayland compositors depending on the sunset/sunrise times.

More information: https://manned.org/wlsunset.

- Automatically compute sunset/sunrise times based on the specified location:

```
wlsunset -l {{latitude}} -L {{longitude}}
```

- Manually set the sunset/sunrise times (time format: `HH:MM`):

```
wlsunset -s {{sunset_time}} -S {{sunrise_time}}
```

- Set the high and low temperatures (default: high - 6500, low - 4000):

```
wlsunset -T {{high_temp}} -t {{low_temp}}
```

- Set the animation time to `n` seconds (only applicable when using manual sunset/sunrise times):

```
wlsunset -d {{n}} -s {{sunset_time}} -S {{sunrise_time}}
```

- Set the gamma (default: 1.0):

```
wlsunset -g {{gamma_value}}
```

wondershaper

Allows the user to limit the bandwidth of network adapters.

More information: https://github.com/magnific0/wondershaper#usage.

- Show the current [s]tatus of a specific [a]dapter:

```
wondershaper -s -a {{adapter_name}}
```

- Clear limits from a specific [a]dapter:

```
wondershaper -c -a {{adapter_name}}
```

- Set a specific maximum [d]ownload rate (in Kbps):

```
wondershaper -a {{adapter_name}} -d {{1024}}
```

- Set a specific maximum [u]pload rate (in Kbps):

```
wondershaper -a {{adapter_name}} -u {{512}}
```

- Set a specific maximum [d]ownload rate and [u]pload rate (in Kpbs):

```
wondershaper -a {{adapter_name}} -d {{1024}} -u {{512}}
```

- Display [h]elp:

```
wondershaper -h
```

wordgrinder

Word processor.

More information: https://cowlark.com/wordgrinder.

- Start WordGrinder (loads a blank document by default):

```
wordgrinder
```

- Open a given file:

```
wordgrinder {{path/to/file}}
```

- Show the menu:

```
<Alt m>
```

wormhole

Get things from one computer to another, safely.

More information: https://magic-wormhole.readthedocs.io.

- Send a file:

```
wormhole send {{path/to/file}}
```

- Receive a file:

```
wormhole receive {{wormhole_code}}
```

- Send raw text:

```
wormhole send
```

- Send a file with a custom code phrase:

```
wormhole send --code {{custom_code}} {{path/to/file}}
```

wp

The official interface to manage WordPress instances.

More information: https://developer.wordpress.org/cli/commands/.

- Print information about the operating system, shell, PHP, and WP-CLI (`wp`) installation:

```
wp --info
```

- Update WP-CLI:

```
wp cli update
```

- Download a fresh WordPress installation to current directory, optionally specifying the locale:

```
wp core download --locale={{locale}}
```

- Create basic `wpconfig` file (assuming database on `localhost`):

```
wp config create --dbname={{dbname}} --dbuser={{dbuser}} --dbpass={{dbpass}}
```

- Install and activate a WordPress plugin:

```
wp plugin install {{plugin}} --activate
```

- Replace all instances of a string in the database:

```
wp search-replace {{old_string}} {{new_string}}
```

- Import the contents of a WordPress Extended RSS (WXR) file:

```
wp import {{path/to/file.xml}}
```

wpa_supplicant

Manage protected wireless networks.

More information: https://manned.org/wpa_supplicant.1.

- Join a protected wireless network:

```
wpa_supplicant -i {{interface}} -c {{path/to/
wpa_supplicant_conf.conf}}
```

- Join a protected wireless network and run it in a daemon:

```
wpa_supplicant -B -i {{interface}} -c {{path/to/
wpa_supplicant_conf.conf}}
```

wpaclean

Clean capture files to get only the 4-way handshake and a beacon.

Part of Aircrack-ng network software suite.

More information: https://manned.org/wpaclean.1.

- Clean capture and save only the 4-way handshake and a beacon in the result:

```
wpaclean {{path/to/result.cap}} {{path/to/capture.cap}}
```

- Clean multiple captures and save 4-way handshakes and beacons in the result:

```
wpaclean {{path/to/result.cap}} {{path/to/capture1.cap path/to/capture2.cap ...}}
```

wpm

Typeracer-like console app for measuring your WPM.

More information: https://github.com/cslarsen/wpm.

- Start `wpm`:

  ```
  wpm
  ```

- Start `wpm` with short texts:

  ```
  wpm --short
  ```

- Start `wpm` using a specific text file:

  ```
  wpm --load {{path/to/file.txt}}
  ```

- Tag your race scores:

  ```
  wpm --tag {{tag_name}}
  ```

- Show score statistics grouped by tags:

  ```
  wpm --stats
  ```

- Start `wpm` with monochrome colors:

  ```
  wpm --monochrome
  ```

wpscan

WordPress vulnerability scanner.

More information: https://github.com/wpscanteam/wpscan.

- Update the vulnerability database:

```
wpscan --update
```

- Scan a WordPress website:

```
wpscan --url {{url}}
```

- Scan a WordPress website, using random user agents and passive detection:

```
wpscan --url {{url}} --stealthy
```

- Scan a WordPress website, checking for vulnerable plugins and specifying the path to the `wp-content` directory:

```
wpscan --url {{url}} --enumerate {{vp}} --wp-content-dir {{remote/path/to/wp-content}}
```

- Scan a WordPress website through a proxy:

```
wpscan --url {{url}} --proxy {{protocol://ip:port}} --proxy-auth {{username:password}}
```

- Perform user identifiers enumeration on a WordPress website:

```
wpscan --url {{url}} --enumerate {{u}}
```

- Execute a password guessing attack on a WordPress website:

```
wpscan --url {{url}} --usernames {{username|path/to/usernames.txt}} --passwords {{path/to/passwords.txt}} threads {{20}}
```

- Scan a WordPress website, collecting vulnerability data from the WPVulnDB (https://wpvulndb.com/):

```
wpscan --url {{url}} --api-token {{token}}
```

wrangler

Cloudflare Workers tool.

More information: https://developers.cloudflare.com/workers/wrangler/commands/.

- Initialize a project with a skeleton configuration:

```
wrangler init {{project_name}}
```

- Authenticate with Cloudflare:

```
wrangler login
```

- Start a local development server:

```
wrangler dev --host {{hostname}}
```

- Publish the worker script:

```
wrangler publish
```

- Aggregate logs from the production worker:

```
wrangler tail
```

write

Write a message on the terminal of a specified logged in user (**`<Ctrl c>`** to stop writing messages).

Use the **`who`** command to find out all terminal_ids of all active users active on the system.

See also: **`mesg`**.

More information: https://manned.org/write.1p.

- Send a message to a given user on a given terminal ID:

```
write {{username}} {{terminal_id}}
```

- Send message to "testuser" on terminal `/dev/tty/5`:

```
write {{testuser}} {{tty/5}}
```

- Send message to "johndoe" on pseudo terminal `/dev/pts/5`:

```
write {{johndoe}} {{pts/5}}
```

wrk

HTTP benchmarking tool.

More information: https://github.com/wg/wrk.

- Run a benchmark for 30 seconds, using 12 threads, and keeping 400 HTTP connections open:

```
wrk {{[-t|--threads]}} {{12}} {{[-c|--connections]}} {{400}} {{[-d|--duration]}} {{30s}} "{{http://127.0.0.1:8080/index.html}}"
```

- Run a benchmark with a custom header:

```
wrk {{[-t|--threads]}} {{2}} {{[-c|--connections]}} {{5}} {{[-d|--duration]}} {{5s}} {{[-H|--header]}} "{{Host: example.com}}" "{{http://example.com/index.html}}"
```

- Run a benchmark with a request timeout of 2 seconds:

```
wrk {{[-t|--threads]}} {{2}} {{[-c|--connections]}} {{5}} {{[-d|--duration]}} {{5s}} --timeout {{2s}} "{{http://example.com/index.html}}"
```

wuzz

Interactively inspect HTTP requests and responses.

More information: https://github.com/asciimoo/wuzz.

- Start `wuzz`:

```
wuzz
```

- Send an HTTP request:

```
<Ctrl r>
```

- Switch to the next view:

```
<Ctrl j><Tab>
```

- Switch to the previous view:

```
<Ctrl k><Shift Tab>
```

- Display help:

```
<F1>
```

x11docker

Securely run GUI applications and desktop UIs in Docker containers.

See also: **xephyr**.

More information: https://github.com/mviereck/x11docker#terminal-syntax.

- Launch VLC in a container:

```
x11docker {{[-p|--pulseaudio]}} --share {{$HOME/Videos}}
{{jess/vlc}}
```

- Launch Xfce in a window:

```
x11docker {{[-d|--desktop]}} {{x11docker/xfce}}
```

- Launch GNOME in a window:

```
x11docker {{[-d|--desktop]}} {{[-g|--gpu]}} --
init={{systemd}} {{x11docker/gnome}}
```

- Launch KDE Plasma in a window:

```
x11docker {{[-d|--desktop]}} {{[-g|--gpu]}} --
init={{systemd}} {{x11docker/kde-plasma}}
```

- Display help:

```
x11docker --help
```

x8

A hidden parameters discovery suite for identifying vulnerable or interesting web parameters.

More information: https://github.com/Sh1Yo/x8/blob/main/docs.md.

- Check hidden parameters in a URL query:

```
x8 {{[-u|--url]}} {{https://example.com/}} {{[-w|--
wordlist]}} {{path/to/wordlist.txt}}
```

- Check parameters with a custom query injection point (`%s`):

```
x8 {{[-u|--url]}} {{https://example.com/?something=1%26%s}}
{{[-w|--wordlist]}} {{path/to/wordlist.txt}}
```

- Send parameters via POST body with JSON format:

```
x8 {{[-u|--url]}} {{https://example.com/}} {{[-X|--method]}}
{{POST}} {{[-b|--body]}} '{{{"x":{%s\}\}}}' {{[-w|--
wordlist]}} {{path/to/wordlist.txt}}
```

- Check parameters with a custom template (`%k` for key, `%v` for value):

```
x8 {{[-u|--url]}} {{https://example.com/}} {{[-P|--param-
template]}} {{user[%k]=%v}} {{[-w|--wordlist]}} {{path/to/
wordlist.txt}}
```

- Encode parameters for special characters in queries:

```
x8 {{[-u|--url]}} {{https://example.com/?path=..
%2faction.php%3f%s%23}} --encode {{[-w|--wordlist]}} {{path/
to/wordlist.txt}}
```

- Discover hidden headers for a URL:

```
x8 {{[-u|--url]}} {{https://example.com/}} --headers {{[-w|--
wordlist]}} {{path/to/headers.txt}}
```

- Check multiple URLs in parallel with high concurrency and verify found parameters:

```
x8 {{[-u|--url]}} {{https://example.com/}} {{https://
4rt.one/}} {{[-W|--workers]}} {{0}} -c {{3}} --verify
```

- Save request and response data for found parameters to a directory:

```
x8 {{[-u|--url]}} {{https://example.com/}} {{[-w|--
wordlist]}} {{path/to/wordlist.txt}} --save-responses {{path/
to/output_directory}}
```

x_x

View Excel and CSV files.

More information: https://github.com/kristianperkins/x_x.

- View an XLSX or CSV file:

```
x_x {{file.xlsx|file.csv}}
```

- View an XLSX or CSV file, using the first row as table headers:

```
x_x {{[-h|--heading]}} {{0}} {{file.xlsx|file.csv}}
```

- View a CSV file with unconventional delimiters:

```
x_x {{[-d|--delimiter]}} '{{;}}' {{[-q|--quotechar]}}
'{{|}}' {{file.csv}}
```

xar

Manage .xar archives.

More information: https://manned.org/xar.

- Create a xar archive of all files in a given directory:

```
xar -cf {{archive.xar}} {{path/to/directory}}
```

- List the contents of a given xar archive:

```
xar -tf {{archive.xar}}
```

- Extract the contents of a given xar archive to the current directory:

```
xar -xf {{archive.xar}}
```

xargs

Execute a command with piped arguments coming from another command, a file, etc.

The input is treated as a single block of text and split into separate pieces on spaces, tabs, newlines and end-of-file.

More information: https://www.gnu.org/software/findutils/manual/html_mono/find.html#Invoking-xargs.

- Run a command using the input data as arguments:

```
{{arguments_source}} | xargs {{command}}
```

- Run multiple chained commands on the input data:

```
{{arguments_source}} | xargs sh -c "{{command1}} &&
{{command2}} | {{command3}}"
```

- Gzip all files with `.log` extension taking advantage of multiple threads (`-print0` uses a null character to split file names, and `-0` uses it as delimiter):

```
find . -name '*.log' -print0 | xargs {{[-0|--null]}} {{[-P|--
max-procs]}} {{4}} {{[-n|--max-args]}} 1 gzip
```

- Execute the command once per argument:

```
{{arguments_source}} | xargs {{[-n|--max-args]}} 1
{{command}}
```

- Execute the command once for each input line, replacing any occurrences of the placeholder (here marked as _) with the input line:

```
{{arguments_source}} | xargs -I _ {{command}} _
{{optional_extra_arguments}}
```

- Parallel runs of up to `max-procs` processes at a time; the default is 1. If `max-procs` is 0, xargs will run as many processes as possible at a time:

```
{{arguments_source}} | xargs {{[-P|--max-procs]}} {{max-
procs}} {{command}}
```

- Prompt user for confirmation before executing command (confirm with `y` or `Y`):

```
{{arguments_source}} | xargs {{[-p|--interactive]}}
{{command}}
```

xbmtopbm

Convert an X11 or X10 bitmap to a PBM image.

More information: https://netpbm.sourceforge.net/doc/xbmtopbm.html.

- Convert an XBM image to a PPM image:

```
xbmtopbm {{path/to/input_file.xbm}} > {{path/to/
output_file.pbm}}
```

xcaddy

The custom build tool for the Caddy Web Server.

More information: https://github.com/caddyserver/xcaddy.

- Build Caddy server from source:

```
xcaddy build
```

- Build Caddy server with a specific version (defaults to latest):

```
xcaddy build {{version}}
```

- Build Caddy with a specific module:

```
xcaddy build --with {{module_name}}
```

- Build Caddy and output to a specific file:

```
xcaddy build --output {{path/to/file}}
```

- Build and run Caddy for a development plugin in the current directory:

```
xcaddy run
```

- Build and run Caddy for a development plugin using a specific Caddy config:

```
xcaddy run --config {{path/to/file}}
```

XCV

Cut, copy, and paste in the command-line.

More information: https://github.com/busterc/xcv.

- Cut a file:

```
xcv x {{input_file}}
```

- Copy a file:

```
xcv c {{input_file}}
```

- Paste a file:

```
xcv v {{output_file}}
```

- List files available for pasting:

```
xcv l
```

xdelta

Delta encoding utility.

Often used for applying patches to binary files.

More information: https://github.com/jmacd/xdelta.

- Apply a patch:

```
xdelta -d -s {{path/to/input_file}} {{path/to/
delta_file.xdelta}} {{path/to/output_file}}
```

- Create a patch:

```
xdelta -e -s {{path/to/old_file}} {{path/to/new_file}}
{{path/to/output_file.xdelta}}
```

xe

Execute a command once for each line piped from another command or file.

More information: https://github.com/leahneukirchen/xe/blob/master/README.

- Run a command once for each line of input data as arguments:

```
{{arguments_source}} | xe {{command}}
```

- Execute the commands, replacing any occurrence of the placeholder (marked as `{}`) with the input line:

```
{{arguments_source}} | xe {{command}} {}
{{optional_extra_arguments}}
```

- Execute a shellscript, joining every `N` lines into a single call:

```
echo -e 'a\nb' | xe -N{{2}} -s 'echo $2 $1'
```

- Delete all files with a `.backup` extension:

```
find . -name '{{*.backup}}' | xe rm -v
```

- Run up to `max-jobs` processes in parallel; the default is 1. If `max-jobs` is 0, xe will run as many processes as cpu cores:

```
{{arguments_source}} | xe -j {{max-jobs}} {{command}}
```

Xephyr

A nested X server that runs as an X application.

More information: https://manned.org/xserver-xephyr.

- Create a black window with display ID ":2":

```
Xephyr -br -ac -noreset -screen {{800x600}} {{:2}}
```

- Start an X application on the new screen:

```
DISPLAY=:2 {{command_name}}
```

xetex

Compile a PDF document from XeTeX source files.

More information: https://www.tug.org/xetex/.

- Compile a PDF document:

```
xetex {{source.tex}}
```

- Compile a PDF document, specifying an output directory:

```
xetex -output-directory={{path/to/directory}} {{source.tex}}
```

- Compile a PDF document, exiting if errors occur:

```
xetex -halt-on-error {{source.tex}}
```

xev

Print contents of X events.

More information: https://gitlab.freedesktop.org/xorg/app/xev.

- Monitor all occurring X events:

```
xev
```

- Monitor all X events of the root window instead of creating a new one:

```
xev -root
```

- Monitor all X events of a particular window:

```
xev -id {{window_id}}
```

- Monitor X events from a given category (can be specified multiple times):

```
xev -event {{event_category}}
```

xgettext

Extract gettext strings from code files.

More information: https://www.gnu.org/software/gettext/manual/gettext.html#xgettext-Invocation.

- Scan file and output strings to `messages.po`:

```
xgettext {{path/to/input_file}}
```

- Use a different output filename:

```
xgettext {{[-o|--output]}} {{path/to/output_file}} {{path/to/input_file}}
```

- Append new strings to an existing file:

```
xgettext {{[-j|--join-existing]}} {{[-o|--output]}} {{path/to/output_file}} {{path/to/input_file}}
```

- Don't add a header containing metadata to the output file:

```
xgettext --omit-header {{path/to/input_file}}
```

- Display help:

```
xgettext {{[-h|--help]}}
```

xh

Friendly and fast tool for sending HTTP requests.

Note: **`xh`**, written in rust, serves as an effective drop-in replacement for **`http`**.

See also: **`http`**, **`curl`**.

More information: https://github.com/ducaale/xh.

- Send a GET request (shows response headers and content):

```
xh {{https://postman-echo.com/get}}
```

- Send a POST request with a JSON body (key-value pairs are added to a top-level JSON object - e.g. `{"name": "john", "age": 25}`):

```
xh post {{https://postman-echo.com/post}} {{name=john}}
{{age=25}}
```

- Send a GET request with query parameters (e.g. `https://postman-echo.com/response-headers?foo1=bar1&foo2=bar2`):

```
xh get {{https://postman-echo.com/response-headers}}
{{foo1==bar1}} {{foo2==bar2}}
```

- Send a GET request with a custom header:

```
xh get {{https://postman-echo.com}} {{header-name:header-
value}}
```

- Make a GET request and save the response body to a file:

```
xh {{[-d|--download]}} {{https://example.com}} {{[-o|--
output]}} {{path/to/file}}
```

- Construct a request but do not send it (similar to a dry-run):

```
xh --offline {{get|delete|...}} {{https://example.com}}
```

- Show equivalent `curl` command (this will not send any request):

```
xh --{{curl|curl-long}} {{--follow --verbose get https://
example.com user-agent:curl}}
```

xidel

Download and extract data from HTML/XML pages as well as JSON APIs.

More information: https://www.videlibri.de/xidel.html.

- Print all URLs found by a Google search:

```
xidel {{https://www.google.com/search?q=test}} {{[-e|--
extract]}} "//a/extract(@href, 'url[?]q=([^&]+)&', 1)[. !=
'']"
```

- Print the title of all pages found by a Google search and download them:

```
xidel {{https://www.google.com/search?q=test}} {{[-f|--
follow]}} "{{//a/extract(@href, 'url[?]q=([^&]+)&', 1)[. !=
'']}}" {{[-e|--extract]}} {{//title}} --download
'{{{$host}/}}'
```

- Follow all links on a page and print the titles, with XPath:

```
xidel {{https://example.org}} {{[-f|--follow]}} {{//a}} {{[-
e|--extract]}} {{//title}}
```

- Follow all links on a page and print the titles, with CSS selectors:

```
xidel {{https://example.org}} {{[-f|--follow]}}
"{{css('a')}}" --css {{title}}
```

- Follow all links on a page and print the titles, with pattern matching:

```
xidel {{https://example.org}} {{[-f|--follow]}} "{{<a>{.}</
a>*}}" {{[-e|--extract]}} "{{<title>{.}</title>}}"
```

- Read the pattern from example.xml (which will also check if the element containing "ood" is there, and fail otherwise):

```
xidel {{path/to/example.xml}} {{[-e|--extract]}}
"{{<x><foo>ood</foo><bar>{.}</bar></x>}}"
```

- Print all newest Stack Overflow questions with title and URL using pattern matching on their RSS feed:

```
xidel {{http://stackoverflow.com/feeds}} {{[-e|--extract]}}
"{{<entry><title>{title:=.}</title><link>{uri:=@href}</
link></entry>+}}"
```

- Check for unread Reddit mail, Webscraping, combining CSS, XPath, JSONiq, and automatically form evaluation:

```
xidel {{https://reddit.com}} {{[-f|--follow]}}
"{{form(css('form.login-form')[1], {'user': '$your_username',
'passwd': '$your_password'})}}" {{[-e|--extract]}}
"{{css('#mail')/@title}}"
```

ximtoppm

Convert a XIM file to a PPM image.

More information: https://netpbm.sourceforge.net/doc/ximtoppm.html.

- Convert an XIM image to a PPM image:

```
ximtoppm {{path/to/input_file.xim}} > {{path/to/
output_file.ppm}}
```

- Store the transparency mask of the input image in the specified file:

```
ximtoppm {{[-a|-alphaout]}} {{path/to/alpha_file.pbm}}
{{path/to/input_file.xim}} > {{path/to/output_file.ppm}}
```

xkcdpass

A flexible and scriptable password generator which generates strong passphrases.

Inspired by XKCD 936.

More information: https://github.com/redacted/XKCD-password-generator.

- Generate one passphrase with the default options:

  ```
  xkcdpass
  ```

- Generate one passphrase whose first letters of each word match the provided argument:

  ```
  xkcdpass {{[-a|--acrostic]}} {{acrostic}}
  ```

- Generate passwords interactively:

  ```
  xkcdpass {{[-i|--interactive]}}
  ```

xkill

Kill a window interactively in a graphical session.

See also: **`kill`**, **`killall`**.

More information: https://www.x.org/releases/current/doc/man/man1/xkill.1.xhtml.

- Display a cursor to kill a window when pressing the left mouse button (press any other mouse button to cancel):

```
xkill
```

- Display a cursor to select a window to kill by pressing any mouse button:

```
xkill -button any
```

- Kill a window with a specific ID (use `xwininfo` to get info about windows):

```
xkill -id {{id}}
```

xlsclients

List client applications running on an X11 display.

More information: https://manned.org/xlsclients.

- List clients on the default display:

```
xlsclients
```

- List clients on all screens:

```
xlsclients -a
```

- List clients with detailed information:

```
xlsclients -l
```

- Limit the command output length per client to a specific number of characters:

```
xlsclients -m {{max_command_length}}
```

- Specify a particular display to inspect:

```
xlsclients -display :{{display_number}}
```

- List clients on remote host's display:

```
xlsclients -display {{remote_host}}:0
```

- Display version:

```
xlsclients -version
```

xmake

A cross-platform C & C++ build utility based on Lua.

More information: https://xmake.io/#/getting_started.

- Create an Xmake C project, consisting of a hello world and `xmake.lua`:

```
xmake create {{[-l|--language]}} {{[c|clean]}} {{[-P|--
project]}} {{project_name}}
```

- Build and run an Xmake project:

```
xmake {{[b|build]}} {{[r|run]}}
```

- Run a compiled Xmake target directly:

```
xmake {{[r|run]}} {{target_name}}
```

- Configure a project's build targets:

```
xmake {{[f|config]}} {{[-p |--plat=]}}{{macosx|linux|
iphoneos|...}} {{[-a |--arch=]}}{{x86_64|i386|arm64|...}}
{{[-m |--mode=]}}{{debug|release}}
```

- Install the compiled target to a directory:

```
xmake {{[i|install]}} {{[-o |--installdir=]}}{{path/to/
directory}}
```

xml c14n

This command is an alias of **`xml canonic`**.

- View documentation for the original command:

```
tldr xml canonic
```

xml canonic

Make XML documents canonical.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139560880.

- Make an XML document canonical, preserving comments:

```
xml {{[c14n|canonic]}} {{path/to/input.xml|URI}} > {{path/to/output.xml}}
```

- Make an XML document canonical, removing comments:

```
xml {{[c14n|canonic]}} --without-comments {{path/to/input.xml|URI}} > {{path/to/output.xml}}
```

- Make XML exclusively canonical, using an XPATH from a file, preserving comments:

```
xml {{[c14n|canonic]}} --exc-with-comments {{path/to/input.xml|URI}} {{path/to/c14n.xpath}}
```

- Display help:

```
xml {{[c14n|canonic]}} --help
```

xml depyx

Convert a PYX (ESIS - ISO 8879) document to XML format.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139550832.

- Convert a PYX (ESIS - ISO 8879) document to XML format:

```
xml {{[p2x|depyx]}} {{path/to/input.pyx|URI}} > {{path/to/output.xml}}
```

- Convert a PYX document from `stdin` to XML format:

```
cat {{path/to/input.pyx}} | xml {{[p2x|depyx]}} > {{path/to/output.xml}}
```

- Display help:

```
xml {{[p2x|depyx]}} --help
```

xml edit

Edit an XML document.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139594320.

- Delete elements matching an XPATH from an XML document:

```
xml {{[ed|edit]}} {{[-d|--delete]}} "{{XPATH1}}" {{path/to/input.xml|URI}}
```

- Move an element node of an XML document from XPATH1 to XPATH2:

```
xml {{[ed|edit]}} {{[-m|--move]}} "{{XPATH1}}" "{{XPATH2}}" {{path/to/input.xml|URI}}
```

- Rename all attributes named "id" to "ID":

```
xml {{[ed|edit]}} {{[-r|--rename]}} "{{//*/@id}}" -v "{{ID}}" {{path/to/input.xml|URI}}
```

- Rename sub-elements of the element "table" that are named "rec" to "record":

```
xml {{[ed|edit]}} {{[-r|--rename]}} "{{/xml/table/rec}}" -v "{{record}}" {{path/to/input.xml|URI}}
```

- Update the XML table record with "id=3" to the value "id=5":

```
xml {{[ed|edit]}} {{[-u|--update]}} "{{xml/table/rec[@id=3]/@id}}" {{[-v|--value]}} {{5}} {{path/to/input.xml|URI}}
```

- Display help:

```
xml {{[ed|edit]}} {{[-h|--help]}}
```

xml elements

Extract elements and display the structure of an XML document.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139665568.

- Extract elements from an XML document (producing XPATH expressions):

```
xml {{[el|elements]}} {{path/to/input.xml|URI}} > {{path/to/elements.xpath}}
```

- Extract elements and their attributes from an XML document:

```
xml {{[el|elements]}} -a {{path/to/input.xml|URI}} > {{path/to/elements.xpath}}
```

- Extract elements and their attributes and values from an XML document:

```
xml {{[el|elements]}} -v {{path/to/input.xml|URI}} > {{path/to/elements.xpath}}
```

- Print sorted unique elements from an XML document to see its structure:

```
xml {{[el|elements]}} -u {{path/to/input.xml|URI}}
```

- Print sorted unique elements from an XML document up to a depth of 3:

```
xml {{[el|elements]}} -d{{3}} {{path/to/input.xml|URI}}
```

- Display help:

```
xml {{[el|elements]}} --help
```

xml escape

Escape special XML characters, e.g. **`<a1>`** → **`<a1>`**.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139540960.

- Escape special XML characters in a string:

```
xml {{[esc|escape]}} "{{<a1>}}"
```

- Escape special XML characters from `stdin`:

```
echo "{{<a1>}}" | xml {{[esc|escape]}}
```

- Display help:

```
xml {{[esc|escape]}} --help
```

xml format

Format an XML document.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139569312.

- Format an XML document, indenting with tabs:

```
xml {{[fo|format]}} {{[-t|--indent-tab]}} {{path/to/input.xml|URI}} > {{path/to/output.xml}}
```

- Format an HTML document, indenting with 4 spaces:

```
xml {{[fo|format]}} {{[-H|--html]}} {{[-s|--indent-spaces]}} {{4}} {{path/to/input.html|URI}} > {{path/to/output.html}}
```

- Recover parsable parts of a malformed XML document, without indenting:

```
xml {{[fo|format]}} {{[-R|--recover]}} {{[-n|--noindent]}} {{path/to/malformed.xml|URI}} > {{path/to/recovered.xml}}
```

- Format an XML document from `stdin`, removing the `DOCTYPE` declaration:

```
cat {{path\to\input.xml}} | xml {{[fo|format]}} {{[-D|--dropdtd]}} > {{path/to/output.xml}}
```

- Format an XML document, omitting the XML declaration:

```
xml {{[fo|format]}} {{[-o|--omit-decl]}} {{path\to\input.xml|URI}} > {{path/to/output.xml}}
```

- Display help:

```
xml {{[fo|format]}} --help
```

xml list

List a directory's contents (like **`ls`**) in XML format.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139535968.

- Write the current directory's listing to an XML document:

```
xml {{[ls|list]}} > {{path/to/dir_list.xml}}
```

- Write the specified directory's listing to an XML document:

```
xml {{[ls|list]}} {{path/to/directory}} > {{path/to/dir_list.xml}}
```

- Display help:

```
xml {{[ls|list]}} --help
```

xml p2x

This command is an alias of **`xml depyx`**.

- View documentation for the original command:

```
tldr xml depyx
```

xml pyx

Convert an XML document to PYX (ESIS - ISO 8879) format.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139550832.

- Convert an XML document to PYX format:

```
xml pyx {{path/to/input.xml|URI}} > {{path/to/output.pyx}}
```

- Convert an XML document from `stdin` to PYX format:

```
cat {{path/to/input.xml}} | xml pyx > {{path/to/output.pyx}}
```

- Display help:

```
xml pyx --help
```

xml select

Select from XML documents using XPATHs.

Tip: use **`xml elements`** to display the XPATHs of an XML document.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139652416.

- Select all elements matching "XPATH1" and print the value of their sub-element "XPATH2":

```
xml {{[sel|select]}} {{[-t|--template]}} {{[-m|--match]}}
"{{XPATH1}}" {{[-v|--value-of]}} "{{XPATH2}}" {{path/to/
input.xml|URI}}
```

- Match "XPATH1" and print the value of "XPATH2" as text with new-lines:

```
xml {{[sel|select]}} {{[-T|--text]}} {{[-t|--template]}} {{[-
m|--match]}} "{{XPATH1}}" {{[-v|--value-of]}}
"{{XPATH2}}" {{[-n|--nl]}} {{path/to/input.xml|URI}}
```

- Count the elements of "XPATH1":

```
xml {{[sel|select]}} {{[-t|--template]}} {{[-v|--value-of]}}
"count({{XPATH1}})" {{path/to/input.xml|URI}}
```

- Count all nodes in one or more XML documents:

```
xml {{[sel|select]}} {{[-T|--text]}} {{[-t|--template]}} {{[-
f|--inp-name]}} {{[-o|--output]}} " " {{[-v|--value-of]}}
"count(node())" {{[-n|--nl]}} {{path/to/input1.xml|URI}}
{{path/to/input2.xml|URI}}
```

- Display help:

```
xml {{[sel|select]}} --help
```

xml transform

Transform XML documents using XSLT.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139602800.

- Transform an XML document using an XSL stylesheet, passing one XPATH parameter and one literal string parameter:

```
xml {{[tr|transform]}} {{path/to/stylesheet.xsl}} -p "{{Count='count(/xml/table/rec)'}}" -s {{Text="Count="}} {{path/to/input.xml|URI}}
```

- Display help:

```
xml {{[tr|transform]}} --help
```

xml unescape

Unescape special XML characters, e.g. **`<a1>`** → **`<a1>`**.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139540960.

- Unescape special XML characters from a string:

```
xml {{[unesc|unescape]}} "{{&lt;a1&gt;}}"
```

- Unescape special XML characters from `stdin`:

```
echo "{{&lt;a1&gt;}}" | xml {{[unesc|unescape]}}
```

- Display help:

```
xml {{[esc|escape]}} --help
```

xml validate

Validate XML documents.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139576400.

- Validate one or more XML documents for well-formedness only:

```
xml {{[val|validate]}} {{path/to/input1.xml|URI1 path/to/input2.xml|URI2 ...}}
```

- Validate one or more XML documents against a Document Type Definition (DTD):

```
xml {{[val|validate]}} {{[-d|--dtd]}} {{path/to/schema.dtd}} {{path/to/input1.xml|URI1 path/to/input2.xml|URI2 ...}}
```

- Validate one or more XML documents against an XML Schema Definition (XSD):

```
xml {{[val|validate]}} {{[-s|--xsd]}} {{path/to/schema.xsd}} {{path/to/input1.xml|URI1 path/to/input2.xml|URI2 ...}}
```

- Validate one or more XML documents against a Relax NG schema (RNG):

```
xml {{[val|validate]}} {{[-r|--relaxng]}} {{path/to/schema.rng}} {{path/to/input1.xml|URI1 path/to/input2.xml|URI2 ...}}
```

- Display help:

```
xml {{[val|validate]}} --help
```

xml xmln

This command is an alias of **`xml pyx`**.

- View documentation for the original command:

```
tldr xml pyx
```

xml

XMLStarlet Toolkit: query, edit, check, convert and transform XML documents.

Some subcommands such as **`xml validate`** have their own usage documentation.

More information: https://xmlstar.sourceforge.net/doc/UG/xmlstarlet-ug.html#idm47077139670224.

- Display general help, including the list of subcommands:

```
xml --help
```

- Execute a subcommand with input from a file or URI, printing to `stdout`:

```
xml {{subcommand}} {{options}} {{path/to/input.xml|URI}}
```

- Execute a subcommand using `stdin` and `stdout`:

```
xml {{subcommand}} {{options}}
```

- Execute a subcommand with input from a file or URI and output to a file:

```
xml {{subcommand}} {{options}} {{path/to/input.xml|URI}} > {{path/to/output}}
```

- Display help for a specific subcommand:

```
xml {{subcommand}} --help
```

- Display version:

```
xml --version
```

xmllint

XML parser and linter that supports XPath, a syntax for navigating XML trees.

More information: https://manned.org/xmllint.

- Return all nodes (tags) named "foo":

```
xmllint --xpath "//{{foo}}" {{source_file.xml}}
```

- Return the contents of the first node named "foo" as a string:

```
xmllint --xpath "string(//{{foo}})" {{source_file.xml}}
```

- Return the href attribute of the second anchor element in an HTML file:

```
xmllint --html --xpath "string(//a[2]/@href)" webpage.xhtml
```

- Return human-readable (indented) XML from file:

```
xmllint --format {{source_file.xml}}
```

- Check that an XML file meets the requirements of its DOCTYPE declaration:

```
xmllint --valid {{source_file.xml}}
```

- Validate XML against DTD schema hosted online:

```
xmllint --dtdvalid {{URL}} {{source_file.xml}}
```

xmlstarlet

A XML/XSLT toolkit.

Note: You will likely need to know XPath: https://developer.mozilla.org/en-US/docs/Web/XPath.

More information: https://xmlstar.sourceforge.net/docs.php.

- Format an XML document and print to `stdout`:

```
xmlstarlet format {{path/to/file.xml}}
```

- XML document can also be piped from `stdin`:

```
{{cat path/to/file.xml}} | xmlstarlet format
```

- Print all nodes that match a given XPath:

```
xmlstarlet select --template --copy-of {{xpath}} {{path/to/file.xml}}
```

- Insert an attribute to all matching nodes, and print to `stdout` (source file is unchanged):

```
xmlstarlet edit --insert {{xpath}} --type attr --name {{attribute_name}} --value {{attribute_value}} {{path/to/file.xml}}
```

- Update the value of all matching nodes in place (source file is changed):

```
xmlstarlet edit --inplace --update {{xpath}} --value {{new_value}} {{file.xml}}
```

- Delete all matching nodes in place (source file is changed):

```
xmlstarlet edit --inplace --delete {{xpath}} {{file.xml}}
```

- Escape or unescape special XML characters in a given string:

```
xmlstarlet [un]escape {{string}}
```

- List a given directory as XML (omit argument to list current directory):

```
xmlstarlet ls {{path/to/directory}}
```

xmlto

Apply an XSL stylesheet to an XML document.

More information: https://manned.org/xmlto.

- Convert a DocBook XML document to PDF format:

```
xmlto pdf {{document.xml}}
```

- Convert a DocBook XML document to HTML format and store the resulting files in a separate directory:

```
xmlto -o {{path/to/html_files}} html {{document.xml}}
```

- Convert a DocBook XML document to a single HTML file:

```
xmlto {{html-nochunks}} {{document.xml}}
```

- Specify a stylesheet to use while converting a DocBook XML document:

```
xmlto -x {{stylesheet.xsl}} {{output_format}}
{{document.xml}}
```

xo

A pluggable, zero-configuration linting utility for JavaScript.

More information: https://github.com/xojs/xo.

- Lint files in the "src" directory:

  ```
  xo
  ```

- Lint a given set of files:

  ```
  xo {{path/to/file1.js path/to/file2.js ...}}
  ```

- Automatically fix any lint issues found:

  ```
  xo --fix
  ```

- Lint using spaces as indentation instead of tabs:

  ```
  xo --space
  ```

- Lint using the "prettier" code style:

  ```
  xo --prettier
  ```

xonsh

Python-powered, cross-platform, Unix-gazing shell.

Write and mix sh/Python code in Xonsh (pronounced conch).

More information: https://xon.sh/contents.html.

- Start an interactive shell session:

```
xonsh
```

- Execute a single command and then exit:

```
xonsh -c "{{command}}"
```

- Run commands from a script file and then exit:

```
xonsh {{path/to/script_file.xonsh}}
```

- Define environment variables for the shell process:

```
xonsh -D{{name1}}={{value1}} -D{{name2}}={{value2}}
```

- Load the specified `.xonsh` or `.json` configuration files:

```
xonsh --rc {{path/to/file1.xonsh}} {{path/to/file2.json}}
```

- Skip loading the `.xonshrc` configuration file:

```
xonsh --no-rc
```

xpdf

Portable Document Format (PDF) file viewer.

More information: https://www.xpdfreader.com/xpdf-man.html.

- Open a PDF file:

```
xpdf {{path/to/file.pdf}}
```

- Open a specific page in a PDF file:

```
xpdf {{path/to/file.pdf}} :{{page_number}}
```

- Open a compressed PDF file:

```
xpdf {{path/to/file.pdf.tar}}
```

- Open a PDF file in fullscreen mode:

```
xpdf -fullscreen {{path/to/file.pdf}}
```

- Specify the initial zoom:

```
xpdf -z {{75}}% {{path/to/file.pdf}}
```

- Specify the initial zoom at page width or full page:

```
xpdf -z {{page|width}} {{path/to/file.pdf}}
```

xplr

Terminal-based file system explorer.

More information: https://xplr.dev.

- Open a directory:

```
xplr {{path/to/directory}}
```

- Focus on a file:

```
xplr {{path/to/file}}
```

- Focus on a directory:

```
xplr --force-focus {{path/to/directory}}
```

- Open a directory with specific files or directories selected:

```
xplr {{path/to/directory}} {{path/to/
selected_file_or_directory1}} {{path/to/
selected_file_or_directory2}}
```

xpmtoppm

Convert an X11 pixmap to a PPM image.

More information: https://netpbm.sourceforge.net/doc/xpmtoppm.html.

- Convert an XPM image to a PPM image:

```
xpmtoppm {{path/to/input_file.xpm}} > {{path/to/
output_file.ppm}}
```

- Store the transparency mask of the input image in the specified file:

```
xpmtoppm {{[-a|--alphaout]}} {{path/to/alpha_file.pbm}}
{{path/to/input_file.xpm}} > {{path/to/output_file.ppm}}
```

xprop

Display window and font properties in an X server.

More information: https://manned.org/xprop.

- Display the name of the root window:

```
xprop -root WM_NAME
```

- Display the window manager hints for a window:

```
xprop -name "{{window_name}}" WM_HINTS
```

- Display the point size of a font:

```
xprop -font "{{font_name}}" POINT_SIZE
```

- Display all the properties of the window with the ID 0x200007:

```
xprop -id {{0x200007}}
```

xsp

Mono ASP.NET Web Server.

More information: https://www.mono-project.com/docs/web/aspnet/.

- Listen on all interfaces (`0.0.0.0`) and port `8080`:

```
xsp
```

- Listen on a specific IP address and port:

```
xsp --address {{127.0.0.1}} --port {{8000}}
```

XSV

A CSV toolkit written in Rust.

More information: https://github.com/BurntSushi/xsv.

- Inspect the headers of a file:

```
xsv headers {{path/to/file.csv}}
```

- Count the number of entries:

```
xsv count {{path/to/file.csv}}
```

- Get an overview of the shape of entries:

```
xsv stats {{path/to/file.csv}} | xsv table
```

- Select a few columns:

```
xsv select {{column1,column2}} {{path/to/file.csv}}
```

- Show 10 random entries:

```
xsv sample {{10}} {{path/to/file.csv}}
```

- Join a column from one file to another:

```
xsv join --no-case {{column1}} {{path/to/file1.csv}}
{{column2}} {{path/to/file2.csv}} | xsv table
```

xteddy

A cuddly teddy bear for your X Windows desktop.

More information: https://manned.org/xteddy.

- Display a cuddly teddy bear on your X desktop:

```
xteddy
```

- Use the window manager to display the teddy bear and ignore the "quit" (<q>) command:

```
xteddy -wm -noquit
```

- Make the teddy bear stay on top of all other windows:

```
xteddy -float
```

- Display another image [F]ile instead of the cuddly teddy bear:

```
xteddy -F {{path/to/image}}
```

- Set the initial location of the teddy bear (`width` and `height` are ignored):

```
xteddy -geometry {{width}}x{{height}}+{{x}}+{{y}}
```

xvminitoppm

Convert an XV thumbnail picture to PPM.

More information: https://netpbm.sourceforge.net/doc/xvminitoppm.html.

- Convert an XV thumbnail image file to PPM:

```
xvminitoppm {{path/to/input_file}} > {{path/to/
output_file.ppm}}
```

xwdtopnm

Convert an X11 or X10 window dump file to PNM.

More information: https://netpbm.sourceforge.net/doc/xwdtopnm.html.

- Convert a XWD image file to PBM:

```
xwdtopnm {{path/to/input_file.xwd}} > {{path/to/
output_file.pnm}}
```

- Display information about the conversion process:

```
xwdtopnm {{[-verb|-verbose]}} {{path/to/input_file.xwd}} >
{{path/to/output_file.pnm}}
```

- Display the contents of the X11 header of the input file:

```
xwdtopnm {{[-h|-headerdump]}} {{path/to/input_file.xwd}} >
{{path/to/output_file.pnm}}
```

xwininfo

Display information about windows.

See also: **`xprop`**, **`xkill`**.

More information: https://www.x.org/releases/current/doc/man/man1/xwininfo.1.xhtml.

- Display a cursor to select a window to display its attributes (id, name, size, position, ...):

```
xwininfo
```

- Display the tree of all windows:

```
xwininfo -tree -root
```

- Display the attributes of a window with a specific ID:

```
xwininfo -id {{id}}
```

- Display the attributes of a window with a specific name:

```
xwininfo -name {{name}}
```

- Display the ID of a window searching by name:

```
xwininfo -tree -root | grep {{keyword}} | head -1 | perl -ne 'print $1 if /(0x[\da-f]+)/ig;'
```

xxd

Create a hexadecimal representation (hexdump) from a binary file, or vice-versa.

See also: **`hexyl`**, **`od`**, **`hexdump`**.

More information: https://manned.org/xxd.

- Generate a hexdump from a binary file and display the output:

```
xxd {{input_file}}
```

- Generate a hexdump from a binary file and save it as a text file:

```
xxd {{input_file}} {{output_file}}
```

- Display a more compact output, replacing consecutive zeros (if any) with a star:

```
xxd {{[-a|-autoskip]}} {{input_file}}
```

- Display the output with 10 columns of one octet (byte) each:

```
xxd {{[-c|-cols]}} {{10}} {{input_file}}
```

- Display output only up to a length of 32 bytes:

```
xxd {{[-l|-len]}} {{32}} {{input_file}}
```

- Display the output in plain mode, without any gaps between the columns:

```
xxd {{[-p|-postscript]}} {{input_file}}
```

- Revert a plaintext hexdump back into binary, and save it as a binary file:

```
xxd {{[-r|-revert]}} {{[-p|-postscript]}} {{input_file}} {{output_file}}
```

xxh

Bring your shell with all of your customizations through SSH sessions.

Note: **xxh** does not install anything into system directories on the target machine; removing **~/.xxh** will clear all traces of xxh on the target machine.

More information: https://github.com/xxh/xxh#usage.

- Connect to a host and run the current shell:

```
xxh "{{host}}"
```

- Install the current shell into the target machine without prompting:

```
xxh "{{host}}" ++install
```

- Run the specified shell on the target machine:

```
xxh "{{host}}" ++shell {{xonsh|zsh|fish|bash|osquery}}
```

- Use a specific xxh configuration directory on the target machine:

```
xxh "{{host}}" ++host-xxh-home {{~/.xxh}}
```

- Use the specified configuration file on the host machine:

```
xxh "{{host}}" ++xxh-config {{~/.config/xxh/config.xxhc}}
```

- Specify a password to use for the SSH connection:

```
xxh "{{host}}" ++password "{{password}}"
```

- Install an xxh package on the target machine:

```
xxh "{{host}}" ++install-xxh-packages {{package}}
```

- Set an environment variable for the shell process on the target machine:

```
xxh "{{host}}" ++env {{name}}={{value}}
```

xz

Compress or decompress XZ and LZMA files.

More information: https://manned.org/xz.

- Compress a file using xz:

```
xz {{path/to/file}}
```

- Decompress an XZ file:

```
xz {{[-d|--decompress]}} {{path/to/file.xz}}
```

- Compress a file using lzma:

```
xz {{[-F|--format]}} lzma {{path/to/file}}
```

- Decompress an LZMA file:

```
xz {{[-d|--decompress]}} {{[-F|--format]}} lzma {{path/to/file.lzma}}
```

- Decompress a file and write to `stdout` (implies `--keep`):

```
xz {{[-d|--decompress]}} {{[-c|--stdout]}} {{path/to/file.xz}}
```

- Compress a file, but don't delete the original:

```
xz {{[-k|--keep]}} {{path/to/file}}
```

- Compress a file using the fastest compression:

```
xz -0 {{path/to/file}}
```

- Compress a file using the best compression:

```
xz -9 {{path/to/file}}
```

xzcat

This command is an alias of **`xz --decompress --stdout`**.

- View documentation for the original command:

```
tldr xz
```

xzcmp

Invokes **cmp** on files compressed with **xz**, **lzma**, **gzip**, **bzip2**, **lzop**, or **zstd**.

All options specified are passed directly to **cmp**.

More information: https://manned.org/xzcmp.

- Compare two specific files:

```
xzcmp {{path/to/file1}} {{path/to/file2}}
```

xzdiff

Invokes **diff** on files compressed with **xz**, **lzma**, **gzip**, **bzip2**, **lzop**, or **zstd**.

All options specified are passed directly to **diff**.

More information: https://manned.org/xzdiff.

- Compare two files:

```
xzdiff {{path/to/file1}} {{path/to/file2}}
```

- Compare two files, showing the differences side by side:

```
xzdiff --side-by-side {{path/to/file1}} {{path/to/file2}}
```

- Compare two files and report only that they differ (no details on what is different):

```
xzdiff --brief {{path/to/file1}} {{path/to/file2}}
```

- Compare two files and report when the files are the same:

```
xzdiff --report-identical-files {{path/to/file1}} {{path/to/file2}}
```

- Compare two files using paginated results:

```
xzdiff --paginate {{path/to/file1}} {{path/to/file2}}
```

xzegrep

This command is an alias of **`xzgrep --extended-regexp`**.

See also: **`egrep`**.

- View documentation for the original command:

```
tldr xzgrep
```

xzfgrep

This command is an alias of **`xzgrep --fixed-strings`**.

See also: **`fgrep`**.

- View documentation for the original command:

```
tldr xzgrep
```

xzgrep

Search files possibly compressed with **`xz`**, **`lzma`**, **`gzip`**, **`bzip2`**, **`lzop`**, or **`zstd`** using **`regex`**.

See also: **`grep`**.

More information: https://manned.org/xzgrep.

- Search for a pattern within a file:

```
xzgrep "{{search_pattern}}" {{path/to/file}}
```

- Search for an exact string (disables `regex`):

```
xzgrep {{[-F|--fixed-strings]}} "{{exact_string}}" {{path/to/file}}
```

- Search for a pattern in all files showing line numbers of matches:

```
xzgrep {{[-n|--line-number]}} "{{search_pattern}}" {{path/to/file}}
```

- Use extended `regex` (supports `?`, `+`, `{}`, `()` and `|`), in case-insensitive mode:

```
xzgrep {{[-E|--extended-regexp]}} {{[-i|--ignore-case]}} "{{search_pattern}}" {{path/to/file}}
```

- Print 3 lines of [C]ontext around, [B]efore, or [A]fter each match:

```
xzgrep --{{context|before-context|after-context}} {{3}} "{{search_pattern}}" {{path/to/file}}
```

- Print file name and line number for each match with color output:

```
xzgrep {{[-H|--with-filename]}} {{[-n|--line-number]}} --color=always "{{search_pattern}}" {{path/to/file}}
```

- Search for lines matching a pattern, printing only the matched text:

```
xzgrep {{[-o|--only-matching]}} "{{search_pattern}}" {{path/to/file}}
```

xzless

Display text from **xz** and **lzma** compressed files.

See also: **less**.

More information: https://manned.org/xzless.

- View a compressed file:

```
xzless {{path/to/file}}
```

- View a compressed file and display line numbers:

```
xzless --LINE-NUMBERS {{path/to/file}}
```

- View a compressed file and quit if the entire file can be displayed on the first screen:

```
xzless --quit-if-one-screen {{path/to/file}}
```

xzmore

Display text from **`xz`** or **`lzma`** compressed files.

Almost equivalent to **`xzless`**, except it respects the **`PAGER`** environment variable, uses **`more`** by default and you cannot pass options to the pager.

More information: https://manned.org/xzmore.

- View a compressed file:

```
xzmore {{path/to/file}}
```

ya

Manage Yazi packages and plugins.

More information: https://github.com/sxyazi/yazi.

- Add a package:

```
ya pack {{[-a|--all]}} {{package}}
```

- Upgrade all packages:

```
ya pack {{[-u|--upgrade]}}
```

- Subscribe to messages from all remote instances:

```
ya sub {{kinds}}
```

- Publish a message to the current instance with string body:

```
ya pub --str {{string_message}}
```

- Publish a message to the current instance with JSON body:

```
ya pub --json {{json_message}}
```

- Publish a message to the specified instance with string body:

```
ya pub-to --str {{message}} {{receiver}} {{kind}}
```

yacas

Yet Another Computer Algebra System.

More information: https://www.yacas.org.

- Start an interactive `yacas` session:

```
yacas
```

- While in a `yacas` session, execute a statement:

```
{{Integrate(x)Cos(x)}};
```

- While in a `yacas` session, display an example:

```
Example();
```

- Quit from a `yacas` session:

```
quit
```

- Execute one or more `yacas` scripts (without terminal or prompts), then exit:

```
yacas -p -c {{path/to/script1 path/to/script2 ...}}
```

- Execute and print the result of one statement, then exit:

```
echo "{{Echo( Deriv(x)Cos(1/x) );}}" | yacas -p -c /dev/stdin
```

yacc

Generate an LALR parser (in C) with a formal grammar specification file.

See also: **`bison`**.

More information: https://manned.org/yacc.1p.

- Create a file `y.tab.c` containing the C parser code and compile the grammar file with all necessary constant declarations for values. (Constant declarations file `y.tab.h` is created only when the `-d` flag is used):

```
yacc -d {{path/to/grammar_file.y}}
```

- Compile a grammar file containing the description of the parser and a report of conflicts generated by ambiguities in the grammar:

```
yacc -d {{path/to/grammar_file.y}} -v
```

- Compile a grammar file, and prefix output filenames with `prefix` instead of `y`:

```
yacc -d {{path/to/grammar_file.y}} -v -b {{prefix}}
```

yadm alt

Create symbolic links and process templates for any managed files.

Learn more about templates: https://yadm.io/docs/templates.

More information: https://yadm.io/docs/alternates.

- Create symbolic links between alternate files manually:

```
yadm alt
```

yadm bootstrap

Execute Yadm's bootstrap file.

This file should be created in **`$HOME/.config/yadm/bootstrap`**.

More information: https://yadm.io/docs/bootstrap.

- Execute bootstrap executable:

```
yadm bootstrap
```

yadm clone

Works just like **`git clone`**. In addition you can pass extra flags to configure your repository.

If there is a bootstrap file in the repository, you will be prompted to execute it.

See also: **`git clone`**.

More information: https://yadm.io/docs/common_commands.

- Clone an existing repository:

```
yadm clone {{remote_repository_location}}
```

- Clone an existing repository, then execute the bootstrap file:

```
yadm clone {{remote_repository_location}} --bootstrap
```

- Clone an existing repository and after cloning, do not execute the bootstrap file:

```
yadm clone {{remote_repository_location}} --no-bootstrap
```

- Change the worktree that `yadm` will use during cloning:

```
yadm clone {{remote_repository_location}} --w
{{worktree_file}}
```

- Change the branch that `yadm` gets files from:

```
yadm clone {{remote_repository_location}} -b {{branch}}
```

- Override an existing repository local branch:

```
yadm clone {{remote_repository_location}} -f
```

yadm config

Pass options to **`yadm`**'s config file. Change the **`.config`** of the repository managed by **`yadm`**.

More information: https://github.com/TheLocehiliosan/yadm/blob/master/yadm.md#configuration.

- Set or update a `yadm`'s Git configuration:

```
yadm config {{key.inner-key}} {{value}}
```

- Get a value from `yadm`'s Git configuration:

```
yadm config --get {{key}}
```

- Unset a value in `yadm`'s Git configuration:

```
yadm config --unset {{key}}
```

- List all values in `yadm`'s Git configuration:

```
yadm config --list
```

yadm decrypt

Decrypt files that were encrypted by **`yadm`**.

When activating this command you will be prompted for a password.

More information: https://yadm.io/docs/encryption.

- Decrypt files:

```
yadm decrypt
```

yadm encrypt

Encrypt files listed in the designated encrypt file.

After the files are encrypted they will be save in the designated archive folder.

More information: https://yadm.io/docs/encryption.

- Encrypt files listed in the designated encrypt file:

```
yadm encrypt
```

- Create the necessary files and folders for encryption:

```
touch {{path/to/encrypt_file}} && mkdir {{path/to/
archive_folder}}
```

yadm enter

Run a sub-shell with all Git variables set. This sub-shell can be used to easily interact with the local **`yadm`** repository using Git commands.

This could be useful if you are using a tool which uses Git directly.

More information: https://github.com/TheLocehiliosan/yadm/blob/master/yadm.md#commands.

- Run a sub-shell with all Git variables set:

  ```
  yadm enter
  ```

- Exit the sub-shell:

  ```
  exit
  ```

yadm git-crypt

Git Crypt enables transparent encryption and decryption of files in a git repository.

See also: **`git-crypt`**.

More information: https://github.com/AGWA/git-crypt.

- Initialize repo to use Git Crypt:

```
yadm git-crypt init
```

- Share the repository using GPG:

```
yadm git-crypt add-gpg-user {{user_id}}
```

- After cloning a repository with encrypted files, unlock them:

```
yadm git-crypt unlock
```

- Export a symmetric secret key:

```
yadm git-crypt export-key {{path/to/key_file}}
```

yadm gitconfig

Pass options to **`git config`**. Change the **`.gitconfig`** of the repository managed by **`yadm`**.

See also: **`git config`**.

More information: https://github.com/TheLocehiliosan/yadm/blob/master/yadm.md#commands.

- Update or set a Git configuration value:

```
yadm gitconfig {{key.inner-key}} {{value}}
```

- Get a value from `yadm`'s Git configuration:

```
yadm gitconfig --get {{key}}
```

- Unset a value in `yadm`'s Git configuration:

```
yadm gitconfig --unset {{key}}
```

- List all values in `yadm`'s Git configuration:

```
yadm gitconfig --list
```

yadm init

Initialize a new, empty repository for tracking dotfiles.

The repository is stored in **`$HOME/.local/share/yadm/repo.git`**.

More information: https://yadm.io/docs/getting_started.

- Execute:

```
yadm init
```

- Override the worktree:

```
yadm init -w {{path/to/worktree_folder}}
```

- Overwrite an existing repository:

```
yadm init -f {{path/to/local_repository}}
```

yadm introspect

Look at data that is managed by **`yadm`**.

The purpose of introspection is to support command line completion.

More information: https://github.com/TheLocehiliosan/yadm/blob/master/yadm.md#commands.

- Output commands:

  `yadm introspect commands`

- Output configs:

  `yadm introspect configs`

- Output switches for the main `yadm` command:

  `yadm introspect switches`

- Output repo:

  `yadm introspect repo`

yadm list

Print a list of files managed by **`yadm`**.

More information: https://github.com/TheLocehiliosan/yadm/blob/master/yadm.md#commands.

- Print a list of files managed by `yadm` in the current directory:

```
yadm list
```

- List all files managed by `yadm` completely:

```
yadm list -a
```

yadm perms

Update permissions.

It is usually unnecessary to run this command, as **`yadm`** automatically processes permissions by default. This automatic behavior can be disabled by setting the configuration **`yadm.auto-perms`** to **`"false"`**.

More information: https://github.com/TheLocehiliosan/yadm/blob/master/yadm.md#permissions.

- Change file permissions:

```
yadm perms
```

yadm transcrypt

If **`transcrypt`** is installed, this command allows you to pass options directly to **`transcrypt`**.

With the environment configured to use the yadm repository.

Transcrypt enables transparent encryption and decryption of files in a Git repository.

More information: https://github.com/elasticdog/transcrypt.

- Set the symmetric cipher to utilize for encryption:

```
yadm transcrypt --cipher={{cipher}}
```

- Pass the password to derive the key from:

```
yadm transcrypt --password={{password}}
```

- Assume yes and accept defaults for non-specified options:

```
yadm transcrypt --yes
```

- Display the current repository's cipher and password:

```
yadm transcrypt --display
```

- Re -encrypt all encrypted files using new credentials:

```
yadm transcrypt --rekey
```

yadm upgrade

Upgrade **`yadm`** to the latest version.

Upgrading will attempt to de-initialize and re-initialize your submodules.

More information: https://github.com/TheLocehiliosan/yadm/blob/master/yadm.md#commands.

- Upgrade `yadm` to the latest version:

  ```
  yadm upgrade
  ```

- Force the upgrade regardless of changes:

  ```
  yadm upgrade -f
  ```

yadm

A dotfiles manager that works by using **`git`**.

Some subcommands such as **`init`**, **`clone`**, **`push`**, and **`pull`** have their own usage documentation.

More information: https://yadm.io/docs/overview.

- Override the `yadm` directory. `yadm` stores its configurations relative to this directory:

  ```
  yadm --yadm-dir
  ```

- Override the `yadm` data directory: `yadm` stores its data relative to this directory:

  ```
  yadm --yadm-data
  ```

- Override the location of the `yadm` repository:

  ```
  yadm --yadm-repo
  ```

- Override the location of the `yadm` configuration file:

  ```
  yadm --yadm-config
  ```

- Override the location of the `yadm` encryption configuration:

  ```
  yadm --yadm-encrypt
  ```

- Override the location of the `yadm` encrypted files archive:

  ```
  yadm --yadm-archive
  ```

- Override the location of the `yadm` bootstrap program:

  ```
  yadm --yadm-bootstrap
  ```

yank

Read input from **`stdin`** and display a selection interface that allows a field to be selected and copied to the clipboard.

More information: https://manned.org/yank.

- Yank using the default delimiters (\f, \n, \r, \s, \t):

```
{{sudo dmesg}} | yank
```

- Yank an entire line:

```
{{sudo dmesg}} | yank -l
```

- Yank using a specific delimiter:

```
{{echo hello=world}} | yank -d {{=}}
```

- Only yank fields matching a specific pattern:

```
{{ps ux}} | yank -g "{{[0-9]+}}"
```

yapf

Python style guide checker.

More information: https://github.com/google/yapf.

- Display a diff of the changes that would be made, without making them (dry-run):

```
yapf {{[-d|--diff]}} {{path/to/file}}
```

- Recursively format all Python files in a directory, concurrently:

```
yapf {{[-ri|--recursive --in-place]}} --style {{pep8}} {{[-p|--parallel]}} {{path/to/directory}}
```

yard

Documentation tool for Ruby.

More information: https://rubydoc.info/gems/yard/file/docs/GettingStarted.md#using-yard-to-generate-documentation.

- Create the documentation:

  ```
  yard
  ```

- Create the documentation and save it to one file:

  ```
  yard --one-file
  ```

- List all undocumented objects:

  ```
  yard stats --list-undoc
  ```

yarn-why

Identifies why a Yarn package has been installed.

More information: https://github.com/amio/yarn-why.

- Show why a Yarn package is installed:

```
yarn-why {{package}}
```

yarn

JavaScript and Node.js package manager alternative.

More information: https://yarnpkg.com/cli.

- Install a module globally:

```
yarn global add {{module_name}}
```

- Install all dependencies referenced in the `package.json` file (the `install` is optional):

```
yarn install
```

- Install a module and save it as a dependency to the `package.json` file (add `--dev` to save as a dev dependency):

```
yarn add {{module_name}}@{{version}}
```

- Uninstall a module and remove it from the `package.json` file:

```
yarn remove {{module_name}}
```

- Interactively create a `package.json` file:

```
yarn init
```

- Identify whether a module is a dependency and list other modules that depend upon it:

```
yarn why {{module_name}}
```

yazi

Blazing fast terminal file manager written in Rust.

Efficient, user-friendly, and customizable file management experience.

More information: https://github.com/sxyazi/yazi.

- Launch Yazi from the current directory:

```
yazi
```

- Print debug information:

```
yazi --debug
```

- Write the current working directory on exit to the file:

```
yazi --cwd-file {{path/to/cwd_file}}
```

- Clear the cache directory:

```
yazi --clear-cache
```

ybacklight

Manage screen backlight brightness.

See also: **`xbacklight`**.

More information: https://github.com/pixelcmtd/ybacklight.

- Print current brightness and maximal brightness, shortened and separated by a slash:

```
ybacklight Sc/Sm
```

- Specify the brightness:

```
ybacklight s{{420}}
```

- Increase the brightness by 42 big steps (4200 by default):

```
ybacklight Si42
```

- Decrease the brightness by 300:

```
ybacklight d300
```

ybmtopbm

Convert a Bennet Yee "face" file to PBM.

More information: https://netpbm.sourceforge.net/doc/ybmtopbm.html.

- Convert a YBM image file to PBM:

```
ybmtopbm {{path/to/input_file.ybm}} > {{path/to/
output_file.pbm}}
```

yes

Output something repeatedly.

This command is commonly used to answer yes to every prompt by install commands (such as **`apt-get`**).

More information: https://www.gnu.org/software/coreutils/manual/html_node/yes-invocation.html.

- Repeatedly output "message":

```
yes {{message}}
```

- Repeatedly output "y":

```
yes
```

- Accept everything prompted by the `apt-get` command:

```
yes | sudo apt-get install {{program}}
```

- Repeatedly output a newline to always accept the default option of a prompt:

```
yes ''
```

yesod

Helper tool for Yesod, a Haskell-based web framework.

All Yesod commands are invoked through the **`stack`** project manager.

More information: https://github.com/yesodweb/yesod.

- Create a new scaffolded site, with SQLite as backend, in the `my-project` directory:

  ```
  stack new {{my-project}} {{yesod-sqlite}}
  ```

- Install the Yesod CLI tool within a Yesod scaffolded site:

  ```
  stack build yesod-bin cabal-install --install-ghc
  ```

- Start development server:

  ```
  stack exec -- yesod devel
  ```

- Touch files with altered Template Haskell dependencies:

  ```
  stack exec -- yesod touch
  ```

- Deploy application using Keter (Yesod's deployment manager):

  ```
  stack exec -- yesod keter
  ```

ykinfo

Get basic information from a YubiKey.

More information: https://developers.yubico.com/yubikey-personalization/Manuals/ykinfo.1.html.

- Display all information from YubiKey:

```
ykinfo -a
```

- Get only serial in decimal from YubiKey:

```
ykinfo -s -q
```

- Get capabilities from YubiKey:

```
ykinfo -c
```

ykman config

Enable or disable YubiKey applications.

Note: You can use **`ykman info`** to see currently enabled applications.

More information: https://docs.yubico.com/software/yubikey/tools/ykman/Base_Commands.html#ykman-config-options-command-args.

- Enable an application over USB or NFC (`--enable` can be used multiple times to specify more applications):

  ```
  ykman config {{usb|nfc}} {{[-e|--enable]}} {{otp|u2f|fido2|oath|piv|openpgp|hsmauth}}
  ```

- Disable an application over USB or NFC (`--disable` can be used multiple times to specify more applications):

  ```
  ykman config {{usb|nfc}} {{[-d|--disable]}} {{otp|u2f|fido2|oath|piv|openpgp|hsmauth}}
  ```

- Disable all applications over NFC:

  ```
  ykman config nfc {{[-D|--disable-all]}}
  ```

ykman fido

Manage YubiKey FIDO applications.

More information: https://docs.yubico.com/software/yubikey/tools/ykman/FIDO_Commands.html.

- Display general information about the FIDO2 application:

```
ykman fido info
```

- Change the FIDO pin:

```
ykman fido access change-pin
```

- List resident credentials stored on the YubiKey:

```
ykman fido credentials list
```

- Delete a resident credential from the YubiKey:

```
ykman fido credentials delete {{id}}
```

- List fingerprints stored on the YubiKey (requires a key with a fingerprint sensor):

```
ykman fido fingerprints list
```

- Add a new fingerprint to the YubiKey:

```
ykman fido fingerprints add {{name}}
```

- Delete a fingerprint from the YubiKey:

```
ykman fido fingerprints delete {{name}}
```

- Wipe all FIDO credentials (you have to do this after exceeding the number of PIN retry attempts):

```
ykman fido reset
```

ykman oath

Manage the OATH YubiKey application.

A **`keyword`** can be a part of the name or the issuer.

More information: https://docs.yubico.com/software/yubikey/tools/ykman/OATH_Commands.html.

- Display general information about the OATH application:

  ```
  ykman oath info
  ```

- Change the password used to protect OATH accounts (add `--clear` to remove it):

  ```
  ykman oath access change
  ```

- Add a new account (the issuer is optional):

  ```
  ykman oath accounts add {{[-i|--issuer]}} {{issuer}} {{name}}
  ```

- List all accounts (with their issuers):

  ```
  ykman oath accounts list
  ```

- List all accounts with their current TOTP/HOTP codes (optionally filtering the list with a keyword):

  ```
  ykman oath accounts code {{keyword}}
  ```

- Rename an account:

  ```
  ykman oath accounts rename {{keyword}} {{issuer:name|name}}
  ```

- Delete an account:

  ```
  ykman oath accounts delete {{keyword}}
  ```

- Delete all accounts and restore factory settings:

  ```
  ykman oath reset
  ```

ykman openpgp

Manage the OpenPGP YubiKey application.

Note: You need to use **`gpg --card-edit`** for some settings.

More information: https://docs.yubico.com/software/yubikey/tools/ykman/OpenPGP_Commands.html.

- Display general information about the OpenPGP application:

  ```
  ykman openpgp info
  ```

- Set the number of retry attempts for the User PIN, Reset Code, and Admin PIN, respectively:

  ```
  ykman openpgp access set-retries {{3}} {{3}} {{3}}
  ```

- Change the User PIN, Reset Code or Admin PIN:

  ```
  ykman openpgp access change-{{pin|reset-code|admin-pin}}
  ```

- Factory reset the OpenPGP application (you have to do this after exceeding the number of Admin PIN retry attempts):

  ```
  ykman openpgp reset
  ```

ykman

YubiKey Manager - configure YubiKeys.

If there are multiple YubiKeys connected, you have to add **`--device serial_number`** before a subcommand.

More information: https://docs.yubico.com/software/yubikey/tools/ykman/index.html.

- Display general information about a YubiKey (serial number, firmware version, capabilities, etc.):

```
ykman info
```

- List connected YubiKeys with short, one-line descriptions (including the serial number):

```
ykman list
```

- View documentation for enabling and disabling applications:

```
tldr ykman config
```

- View documentation for managing the FIDO applications:

```
tldr ykman fido
```

- View documentation for managing the OATH application:

```
tldr ykman oath
```

- View documentation for managing the OpenPGP application:

```
tldr ykman openpgp
```

yolo

Train, validate or infer models on various tasks and versions.

More information: https://docs.ultralytics.com/cli/.

- Create a copy of the default configuration in your current working directory:

  ```
  yolo task=init
  ```

- Train the object detection, instance segment, or classification model with the specified configuration file:

  ```
  yolo task={{detect|segment|classify}} mode=train cfg={{path/to/config.yaml}}
  ```

you-get

Download media contents (videos, audios, images) from the Web.

See also: **`yt-dlp`**, **`youtube-viewer`**, **`instaloader`**.

More information: https://you-get.org/#getting-started.

- Print media information about a specific media on the web:

```
you-get {{[-i|--info]}} {{https://example.com/video?id=value}}
```

- Download a media from a specific URL:

```
you-get {{https://example.com/video?id=value}}
```

- Search on Google Videos and download:

```
you-get {{keywords}}
```

- Download a media to a specific location:

```
you-get {{[-o|--output-dir]}} {{path/to/directory}} {{[-O|--output-filename]}} {{filename}} {{https://example.com/watch?v=value}}
```

- Download a media using a proxy:

```
you-get {{[-x|--http-proxy]}} {{proxy_server}} {{https://example.com/watch?v=value}}
```

youtube-dl

Download videos from YouTube and other websites.

See also: **`yt-dlp`**, **`ytfzf`**, **`you-get`**.

More information: https://rg3.github.io/youtube-dl/.

- Download a video or playlist:

```
youtube-dl '{{https://www.youtube.com/watch?v=oHg5SJYRHA0}}'
```

- List all formats that a video or playlist is available in:

```
youtube-dl {{[-F|--list-formats]}} '{{https://
www.youtube.com/watch?v=Mwa0_nE9H7A}}'
```

- Download a video or playlist at a specific quality:

```
youtube-dl {{[-f|--format]}} "{{best[height<=480]}}"
'{{https://www.youtube.com/watch?v=oHg5SJYRHA0}}'
```

- Download the audio from a video and convert it to an MP3:

```
youtube-dl {{[-x|--extract-audio]}} --audio-format {{mp3}}
'{{url}}'
```

- Download the best quality audio and video and merge them:

```
youtube-dl {{[-f|--format]}} bestvideo+bestaudio '{{url}}'
```

- Download video(s) as MP4 files with custom filenames:

```
youtube-dl {{[-f|--format]}} {{mp4}} {{[-o|--output]}} "{{%
(playlist_index)s-%(title)s by %(uploader)s on %
(upload_date)s in %(playlist)s.%(ext)s}}" '{{url}}'
```

- Download a particular language's subtitles along with the video:

```
youtube-dl --sub-lang {{en}} --write-sub '{{https://
www.youtube.com/watch?v=Mwa0_nE9H7A}}'
```

- Download a playlist and extract MP3s from it:

```
youtube-dl {{[-f|--format]}} "bestaudio" {{[-c|--continue]}}
{{[-w|--no-overwrites]}} {{[-i|--ignore-errors]}} {{[-x|--
extract-audio]}} --audio-format mp3 {{[-o|--output]}} "%
(title)s.%(ext)s" '{{url_to_playlist}}'
```

youtube-viewer

Search and play videos from YouTube.

See also: **`you-get`**, **`ytfzf`**, **`yt-dlp`**.

More information: https://github.com/trizen/youtube-viewer.

- Search for a video:

```
youtube-viewer {{search_term}}
```

- Log in to your YouTube account:

```
youtube-viewer --login
```

- Watch a video with a specific URL in VLC:

```
youtube-viewer --player={{vlc}} {{https://youtube.com/watch?v=dQw4w9WgXcQ}}
```

- Display a search prompt and play the selected video in 720p:

```
youtube-viewer -{{7}}
```

yq

A lightweight and portable YAML processor.

More information: https://mikefarah.gitbook.io/yq/.

- Output a YAML file, in pretty-print format (v4+):

```
yq eval {{path/to/file.yaml}}
```

- Output a YAML file, in pretty-print format (v3):

```
yq read {{path/to/file.yaml}} {{[-C|--colors]}}
```

- Output the first element in a YAML file that contains only an array (v4+):

```
yq eval '.[0]' {{path/to/file.yaml}}
```

- Output the first element in a YAML file that contains only an array (v3):

```
yq read {{path/to/file.yaml}} '[0]'
```

- Set (or overwrite) a key to a value in a file (v4+):

```
yq eval '.{{key}} = "{{value}}"' {{[-i|--inplace]}} {{path/
to/file.yaml}}
```

- Set (or overwrite) a key to a value in a file (v3):

```
yq write {{[-i|--inplace]}} {{path/to/file.yaml}} '{{key}}'
'{{value}}'
```

- Merge two files and print to `stdout` (v4+):

```
yq eval-all 'select(filename == "{{path/to/file1.yaml}}") *
select(filename == "{{path/to/file2.yaml}}")' {{path/to/
file1.yaml}} {{path/to/file2.yaml}}
```

- Merge two files and print to `stdout` (v3):

```
yq merge {{path/to/file1.yaml}} {{path/to/file2.yaml}} {{[-
C|--colors]}}
```

yt-dlp

A youtube-dl fork with additional features and fixes.

Download videos from YouTube and other websites.

See also: **ytfzf**.

More information: https://github.com/yt-dlp/yt-dlp#usage-and-options.

- Download a video or playlist (with the default options from command below):

```
yt-dlp "{{https://www.youtube.com/watch?v=oHg5SJYRHA0}}"
```

- List the available downloadable formats for a video:

```
yt-dlp {{[-F|--list-formats]}} "{{https://www.youtube.com/
watch?v=oHg5SJYRHA0}}"
```

- Download a video or playlist using the best MP4 video available (default is "bv*+ba/b"):

```
yt-dlp {{[-f|--format]}} "{{bv*[ext=mp4]+ba[ext=m4a]/
b[ext=mp4]}}" "{{https://www.youtube.com/watch?
v=oHg5SJYRHA0}}"
```

- Extract audio from a video (requires ffmpeg or ffprobe):

```
yt-dlp {{[-x|--extract-audio]}} "{{https://www.youtube.com/
watch?v=oHg5SJYRHA0}}"
```

- Specify audio format and audio quality of extracted audio (between 0 (best) and 10 (worst), default = 5):

```
yt-dlp {{[-x|--extract-audio]}} --audio-format {{mp3}} --
audio-quality {{0}} "{{https://www.youtube.com/watch?
v=oHg5SJYRHA0}}"
```

- Download only the second, fourth, fifth, sixth, and last items in a playlist (the first item is 1, not 0):

```
yt-dlp {{[-I|--playlist-items]}} 2,4:6,-1 "{{https://
youtube.com/playlist?list=PLbzoR-
pLrL6pTJfLQ3UwtB-3V4fimdqnA}}"
```

- Download all playlists of a YouTube channel/user keeping each playlist in a separate directory:

```
yt-dlp {{[-o|--output]}} "{{%(uploader)s/%(playlist)s/%
(playlist_index)s - %(title)s.%(ext)s}}" "{{https://
www.youtube.com/user/TheLinuxFoundation/playlists}}"
```

- Download a Udemy course keeping each chapter in a separate directory:

```
yt-dlp {{[-u|--username]}} {{user}} {{[-p|--password]}}
{{password}} {{[-P|--paths]}} "{{path/to/directory}}" {{[-
o|--output]}} "{{%(playlist)s/%(chapter_number)s - %
(chapter)s/%(title)s.%(ext)s}}" "{{https://www.udemy.com/
java-tutorial}}"
```

ytmdl

Download songs from YouTube and automatically add metadata.

Fetch song information (artist, album, cover art) from iTunes, Spotify, and other sources.

More information: https://github.com/deepjyoti30/ytmdl#usage.

- Download a song by name (with interactive selection):

```
ytmdl {{song_name}}
```

- Download the first result without prompting:

```
ytmdl {{[-q|--quiet]}} {{song_name}}
```

- Download a song to a specific directory:

```
ytmdl {{[-o|--output-dir]}} {{path/to/directory}}
{{song_name}}
```

- Download a song from a YouTube URL:

```
ytmdl --url https://www.youtube.com/watch?v={{oHg5SJYRHA0}}
```

- Download a song in a specific format (mp3, m4a, or opus):

```
ytmdl --format {{mp3|m4a|opus}} {{song_name}}
```

- Download a song with artist and album information:

```
ytmdl --artist {{artist_name}} --album {{album_name}}
{{song_name}}
```

- Download a list of songs from a text file:

```
ytmdl --list {{path/to/list.txt}}
```

- Display help:

```
ytmdl {{[-h|--help]}}
```

yuvsplittoppm

Convert three subsampled Abekas YUV files to one PPM image.

More information: https://netpbm.sourceforge.net/doc/yuvsplittoppm.html.

- Read Akebas YUV bytes from three files starting with basename, merge them into a single PPM image and store it in the specified output file:

```
yuvsplittoppm {{basename}} {{width}} {{height}} > {{path/to/output_file.ppm}}
```

yuvtoppm

Convert Abekas YUV bytes to PPM.

More information: https://netpbm.sourceforge.net/doc/yuvtoppm.html.

- Read Akebas YUV bytes from the specified input file, convert them to a PPM image and store them in the specified output file:

```
yuvtoppm {{width}} {{height}} {{path/to/input_file.yuv}} > {{path/to/output_file.ppm}}
```

yuy2topam

Convert YUY2 bytes to PAM.

More information: https://netpbm.sourceforge.net/doc/yuy2topam.html.

- Convert YUY2 bytes to PAM:

```
yuy2topam {{[-w|-width]}} {{value}} {{[-h|-height]}} {{value}} {{path/to/file.yuy2}} > {{path/to/file.pam}}
```

z

Tracks the most used (by frequency) directories and enables quickly navigating to them using string patterns or **regex**.

More information: https://github.com/rupa/z.

- Go to a directory that contains `string` in the name:

```
z string
```

- Go to a directory that contains `string1` and then `string2`:

```
z string1 string2
```

- Go to the highest-ranked directory matching `string`:

```
z -r string
```

- Go to the most recently accessed directory matching `string`:

```
z -t string
```

- List all directories in `z`'s database matching `string`:

```
z -l string
```

- Remove the current directory from `z`'s database:

```
z -x
```

- Restrict matches to subdirectories of the current directory:

```
z -c {{string}}
```

zapier analytics

Show the status of the analytics that are collected. It is also used to change what is collected.

More information: https://platform.zapier.com/reference/cli#analytics.

- Show the status of collected analytics:

```
zapier analytics
```

- Change how much information is collected:

```
zapier analytics {{[-m|--mode]}} {{enabled|anonymous|
disabled}}
```

- Show extra debugging output:

```
zapier analytics {{[-m|--mode]}} {{enabled|anonymous|
disabled}} {{[-d|--debug]}}
```

zapier build

Build a pushable **`zip`** of a Zapier integration.

More information: https://platform.zapier.com/reference/cli#build.

- Create a build:

  ```
  zapier build
  ```

- Disable smart file inclusion (will only include files required by `index.js`):

  ```
  zapier build --disable-dependency-detection
  ```

- Show extra debugging output:

  ```
  zapier build {{[-d|--debug]}}
  ```

zapier convert

Convert a Visual Builder integration to a CLI integration.

More information: https://platform.zapier.com/reference/cli#convert.

- Convert a visual builder integration:

```
zapier convert {{integration_id}} {{path/to/directory}}
```

- Convert a visual builder integration with a specific version:

```
zapier convert {{integration_id}} {{path/to/directory}} {{[-
v|--version]}}={{version}}
```

- Show extra debugging output:

```
zapier convert --debug
```

zapier history

List the history for a Zapier integration.

More information: https://github.com/zapier/zapier-platform/blob/main/packages/cli/docs/cli.md#history.

- List the history for an integration:

```
zapier history
```

- Display additional debug output:

```
zapier history {{[-d|--debug]}}
```

- List the history with a different output structure:

```
zapier history {{[-f|--format]}} {{plain|json|raw|row|table}}
```

zapier init

Initialize a new Zapier integration.

More information: https://platform.zapier.com/reference/cli#init.

- Initialize a new Zapier integration:

```
zapier init {{path/to/directory}}
```

- Initialize a new Zapier integration with a specific template:

```
zapier init {{path/to/directory}} {{[-t|--template]}}
{{basic-auth|callback|custom-auth|digest-auth|dynamic-
dropdown|files|minimal|oauth1-trello|oauth2|search-or-create|
session-auth|typescript}}
```

- Show extra debugging output:

```
zapier init {{[-d|--debug]}}
```

zapier login

Log in to a Zapier account.

More information: https://github.com/zapier/zapier-platform/blob/main/packages/cli/docs/cli.md#login.

- Log in to a Zapier account:

```
zapier login
```

- Log in to a Zapier account with a deploy key for SSO:

```
zapier login {{[-s|--sso]}}
```

- Display additional debug output:

```
zapier login {{[-d|--debug]}}
```

zapier push

Build and upload a Zapier integration.

More information: https://platform.zapier.com/reference/cli#push.

- Push an integration to Zapier:

  ```
  zapier push
  ```

- Disable smart file inclusion (will only include files required by `index.js`):

  ```
  zapier push --disable-dependency-detection
  ```

- Show extra debugging output:

  ```
  zapier push {{[-d|--debug]}}
  ```

zapier scaffold

Add a starting trigger, create, search, or resource to an integration.

More information: https://platform.zapier.com/reference/cli#scaffold.

- Scaffold a new trigger, create, search, or resource:

```
zapier scaffold {{trigger|search|create|resource}} {{noun}}
```

- Specify a custom destination directory for the scaffolded files:

```
zapier scaffold {{trigger|search|create|resource}} {{noun}}
{{[-d|--dest]}}={{path/to/directory}}
```

- Overwrite existing files when scaffolding:

```
zapier scaffold {{trigger|search|create|resource}} {{noun}}
{{[-f|--force]}}
```

- Exclude comments from the scaffolded files:

```
zapier scaffold {{trigger|search|create|resource}} {{noun}}
--no-help
```

- Show extra debugging output:

```
zapier scaffold {{[-d|--debug]}}
```

zapier test

Test a Zapier integration via the **`test`** script in your **`package.json`**.

More information: https://github.com/zapier/zapier-platform/blob/main/packages/cli/docs/cli.md#test.

- Test an integration:

  ```
  zapier test
  ```

- Test an integration while skipping validation:

  ```
  zapier test --skip-validate
  ```

- Test an integration with additional debug output:

  ```
  zapier test {{[-d|--debug]}}
  ```

zapier validate

Validate a Zapier integration.

More information: https://github.com/zapier/zapier-platform/blob/main/packages/cli/docs/cli.md#validate.

- Validate an integration:

```
zapier validate
```

- Validate an integration without style checks:

```
zapier validate --without-style
```

- Validate an integration without running the build script:

```
zapier validate --skip-build
```

- Validate an integration with additional debug output:

```
zapier validate {{[-d|--debug]}}
```

- Validate an integration with a different output structure:

```
zapier validate {{[-f|--format]}} {{plain|json|raw|row|
table}}
```

zapier versions

List the versions for a Zapier integration.

More information: https://github.com/zapier/zapier-platform/blob/main/packages/cli/docs/cli.md#validate.

- List the versions for an integration:

```
zapier versions
```

- Include deprecated versions:

```
zapier versions {{[-a|--all]}}
```

- Display additional debug output:

```
zapier versions {{[-d|--debug]}}
```

- List the versions with a different output structure:

```
zapier versions {{[-f|--format]}} {{plain|json|raw|row|
table}}
```

zapier

Create, automate, and manage zapier integrations.

Some subcommands such as **`build`**, **`init`**, **`scaffold`**, **`push`**, **`test`**, etc. have their own usage documentation.

More information: https://platform.zapier.com/reference/cli.

- Connect to a Zapier account:

```
zapier login
```

- Initialize a new Zapier integration with a project template:

```
zapier init {{path/to/directory}}
```

- Add a starting trigger, create, search, or resource to your integration:

```
zapier scaffold {{trigger|create|search|resource}} {{name}}
```

- Test an integration:

```
zapier test
```

- Build and upload an integration to Zapier:

```
zapier push
```

- Display help:

```
zapier help
```

- Display help for a specific command:

```
zapier help {{command}}
```

zbarimg

Scan and decode bar codes from image file(s).

More information: https://manned.org/zbarimg.

- Process an image file:

```
zbarimg {{image_file}}
```

zcat

Print data from **`gzip`** compressed files to **`stdout`**.

More information: https://www.gnu.org/software/gzip/manual/gzip.html.

- Print the uncompressed contents of a `gzip` archive to `stdout`:

```
zcat {{path/to/file.txt.gz}}
```

- Print compression details of a `gzip` archive to `stdout`:

```
zcat {{[-l|--list]}} {{path/to/file.txt.gz}}
```

- Test the integrity of a compressed file verbosely:

```
zcat {{[-v|--verbose]}} {{[-t|--test]}} {{path/to/file.txt.gz}}
```

- Suppress all warnings when decompressing a file:

```
zcat {{[-q|--quiet]}} {{path/to/file.txt.gz}}
```

- Avoid any system crashes when decompressing a file (slower output):

```
zcat --synchronous {{path/to/file.txt.gz}}
```

zcmp

Compare compressed files.

More information: https://manned.org/zcmp.

- Invoke `cmp` on two files compressed via `gzip`:

```
zcmp {{path/to/file1.gz}} {{path/to/file2.gz}}
```

- Compare a file to its gzipped version (assuming `.gz` exists already):

```
zcmp {{path/to/file}}
```

zdb

ZFS debugger.

More information: https://manned.org/zdb.

- Show detailed configuration of all mounted ZFS zpools:

```
zdb
```

- Show detailed configuration for a specific ZFS pool:

```
zdb {{[-C|--config]}} {{poolname}}
```

- Show statistics about number, size and deduplication of blocks:

```
zdb {{[-b|--block-stats]}} {{poolname}}
```

zdiff

Invoke **`diff`** on **`gzip`** archives.

More information: https://manned.org/zdiff.

- Compare two files, uncompressing them if necessary:

```
zdiff {{path/to/file1.gz}} {{path/to/file2.gz}}
```

- Compare a file to a `gzip` archive with the same name:

```
zdiff {{path/to/file}}
```

zed

Text editor designed to be fast, efficient and convenient.

More information: https://zed.dev/docs/#cli.

- Open specific paths in Zed:

```
zed {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Open a path in foreground and display logs:

```
zed {{path/to/project}} --foreground
```

- Open a path in new window:

```
zed {{path/to/project}} {{[-n|--new]}}
```

- Open a file at the given line number and column:

```
zed {{path/to/file}}:{{line_number}}:{{column_number}}
```

- Open a diff tab in Zed for two versions of a file:

```
zed --diff {{path/to/old_file}} {{path/to/new_file}}
```

zeditor

This command is an alias of **zed**.

- View documentation for the original command:

```
tldr zed
```

zeek

Passive network traffic analyzer.

Any output and log files will be saved to the current working directory.

More information: https://docs.zeek.org/en/lts/quickstart.html#zeek-as-a-command-line-utility.

- Analyze live traffic from a network interface:

```
sudo zeek --iface {{interface}}
```

- Analyze live traffic from a network interface and load custom scripts:

```
sudo zeek --iface {{interface}} {{script1 script2 ...}}
```

- Analyze live traffic from a network interface, without loading any scripts:

```
sudo zeek --bare-mode --iface {{interface}}
```

- Analyze live traffic from a network interface, applying a `tcpdump` filter:

```
sudo zeek --filter {{path/to/filter}} --iface {{interface}}
```

- Analyze live traffic from a network interface using a watchdog timer:

```
sudo zeek --watchdog --iface {{interface}}
```

- Analyze traffic from a PCAP file:

```
zeek --readfile {{path/to/file.trace}}
```

zegrep

Find extended **regex** patterns in compressed files using **egrep**.

More information: https://www.unix.com/man-page/freebsd/1/zegrep/.

- Search for extended regex (supporting ?, +, {}, () and |) in a compressed file (case-sensitive):

```
zegrep "{{search_pattern}}" {{path/to/file}}
```

- Search for extended regex (supporting ?, +, {}, () and |) in a compressed file (case-insensitive):

```
zegrep {{[-i|--ignore-case]}} "{{search_pattern}}" {{path/to/file}}
```

- Search for lines that do not match a pattern:

```
zegrep {{[-v|--invert-match]}} "{{search_pattern}}" {{path/to/file}}
```

- Print file name and line number for each match:

```
zegrep {{[-H|--with-filename]}} {{[-n|--line-number]}} "{{search_pattern}}" {{path/to/file}}
```

- Search for lines matching a pattern, printing only the matched text:

```
zegrep {{[-o|--only-matching]}} "{{search_pattern}}" {{path/to/file}}
```

- Recursively search files in a compressed file for a pattern:

```
zegrep {{[-r|--recursive]}} "{{search_pattern}}" {{path/to/file}}
```

zeisstopnm

Convert a Zeiss confocal file to Netbpm format.

More information: https://manned.org/zeisstopnm.1.

- Convert a Zeiss cofocal file into either `.pgm` or `.ppm` format:

```
zeisstopnm {{path/to/file}}
```

- Convert a Zeiss cofocal file to Netbpm format while explicitly specifying the target file type:

```
zeisstopnm -{{pgm|ppm}} {{path/to/file}}
```

zek

Generate a Go struct from XML.

More information: https://github.com/miku/zek#usage.

- Generate a Go struct from a given XML from `stdin` and display output on `stdout`:

```
cat {{path/to/input.xml}} | zek
```

- Generate a Go struct from a given XML from `stdin` and send output to a file:

```
curl {{[-s|--silent]}} {{https://url/to/xml}} | zek -o {{path/to/output.go}}
```

- Generate an example Go program from a given XML from `stdin` and send output to a file:

```
cat {{path/to/input.xml}} | zek -p -o {{path/to/output.go}}
```

zellij

Terminal multiplexer with batteries included.

See also: **`tmux`**, **`screen`**.

More information: https://zellij.dev/documentation/.

- Start a new named session:

```
zellij {{[-s|--session]}} {{name}}
```

- List existing sessions:

```
zellij {{[ls|list-sessions]}}
```

- Attach to the most recently used session:

```
zellij {{[a|attach]}}
```

- Open a new pane (inside a zellij session):

```
<Alt n>
```

- Detach from the current session (inside a zellij session):

```
<Ctrl o><d>
```

zfgrep

Matches fixed strings in possibly compressed files.

Equivalent to **`grep --fixed-strings`** with input decompressed first if necessary.

More information: https://manned.org/zfgrep.

- Search for an exact string in a file:

```
zfgrep {{search_string}} {{path/to/file}}
```

- Count the number of lines that match the given string in a file:

```
zfgrep {{[-c|--count]}} {{search_string}} {{path/to/file}}
```

- Show the line number in the file along with the matching lines:

```
zfgrep {{[-n|--line-number]}} {{search_string}} {{path/to/
file}}
```

- Display all lines except those that contain the search string:

```
zfgrep {{[-v|--invert-match]}} {{search_string}} {{path/to/
file}}
```

- List only filenames whose content matches the search string at least once:

```
zfgrep {{[-l|--files-with-matches]}} {{search_string}}
{{path/to/file1 path/to/file2 ...}}
```

zfs

Manage ZFS filesystems.

More information: https://manned.org/zfs.

- List all available zfs filesystems:

```
zfs list
```

- Create a new ZFS filesystem:

```
zfs create {{pool_name/filesystem_name}}
```

- Delete a ZFS filesystem:

```
zfs destroy {{pool_name/filesystem_name}}
```

- Create a Snapshot of a ZFS filesystem:

```
zfs snapshot {{pool_name/filesystem_name}}@{{snapshot_name}}
```

- Enable compression on a filesystem:

```
zfs set compression=on {{pool_name/filesystem_name}}
```

- Change mountpoint for a filesystem:

```
zfs set mountpoint=/{{path/to/mount_point}} {{pool_name/
filesystem_name}}
```

zgrep

Grep text patterns from files within compressed file (equivalent to **`grep -Z`**).

More information: https://manned.org/zgrep.

- Grep a pattern in a compressed file (case-sensitive):

```
zgrep {{pattern}} {{path/to/compressed_file}}
```

- Grep a pattern in a compressed file (case-insensitive):

```
zgrep {{[-i|--ignore-case]}} {{pattern}} {{path/to/
compressed_file}}
```

- Output count of lines containing matched pattern in a compressed file:

```
zgrep {{[-c|--count]}} {{pattern}} {{path/to/
compressed_file}}
```

- Display the lines which don't have the pattern present (Invert the search function):

```
zgrep {{[-v|--invert-match]}} {{pattern}} {{path/to/
compressed_file}}
```

- Grep a compressed file for multiple patterns:

```
zgrep {{[-e|--regexp]}} "{{pattern_1}}" {{[-e|--regexp]}}
"{{pattern_2}}" {{path/to/compressed_file}}
```

- Use extended `regex` (supporting `?`, `+`, `{}`, `()` and `|`):

```
zgrep {{[-E|--extended-regexp]}} {{regex}} {{path/to/file}}
```

- Print 3 lines of [C]ontext around, [B]efore, or [A]fter each match:

```
zgrep --{{context|before-context|after-context}} 3
{{pattern}} {{path/to/compressed_file}}
```

zig

The Zig compiler and toolchain.

More information: https://ziglang.org/documentation/master/.

- Compile the project in the current directory:

```
zig build
```

- Compile and run the project in the current directory:

```
zig build run
```

- Initialize a `zig build` project with library and executable:

```
zig init
```

- Create and run a test build:

```
zig test {{path/to/file.zig}}
```

- Cross compile, build and run a project for `x86_64` architecture and `windows` operating system:

```
zig build run -fwine -Dtarget=x86_64-windows
```

- Reformat Zig source into canonical form:

```
zig fmt {{path/to/file.zig}}
```

- Translate a C file to `zig`:

```
zig translate-c -lc {{path/to/file.c}}
```

- Use Zig as a drop-in C++ compiler:

```
zig c++ {{path/to/file.cpp}}
```

zinit

Flexible and blazing fast Zsh plugin manager.

More information: https://github.com/zdharma-continuum/zinit.

- Show all installed plugins:

```
zinit plugins
```

- Go to the directory of a plugin:

```
zinit cd {{repository_owner}}/{{repository_name}}
```

- Fetch the newest version of `zinit` from GitHub:

```
zinit self-update
```

- List status of all installed completions:

```
zinit completions
```

- Install completions for a plugin:

```
zinit creinstall {{repository_owner}}/{{repository_name}}
```

- Delete completion for a plugin:

```
zinit cuninstall {{repository_owner}}/{{repository_name}}
```

zint

Generate barcodes and QR codes.

More information: https://www.zint.org.uk/manual/chapter/4.

- Generate a barcode and save it:

```
zint --data "{{UTF-8 data}}" --output {{path/to/file}}
```

- Specify a code type for generation:

```
zint --barcode {{code_type}} --data "{{UTF-8 data}}" --output
{{path/to/file}}
```

- List all supported code types:

```
zint --types
```

zip

Package and compress (archive) files into a Zip archive.

See also: **`unzip`**.

More information: https://manned.org/zip.

- Add files/directories to a specific archive:

```
zip {{[-r|--recurse-paths]}} {{path/to/compressed.zip}}
{{path/to/file_or_directory1 path/to/file_or_directory2 ...}}
```

- Remove files/directories from a specific archive:

```
zip {{[-d|--delete]}} {{path/to/compressed.zip}} {{path/to/
file_or_directory1 path/to/file_or_directory2 ...}}
```

- Archive files/directories excluding specified ones:

```
zip {{[-r|--recurse-paths]}} {{path/to/compressed.zip}}
{{path/to/file_or_directory1 path/to/file_or_directory2 ...}}
{{[-x|--exclude]}} {{path/to/excluded_files_or_directories}}
```

- Archive files/directories with a specific compression level (`0` - the lowest, `9` - the highest):

```
zip {{[-r|--recurse-paths]}} -{{0..9}} {{path/to/
compressed.zip}} {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Create an encrypted archive with a specific password:

```
zip {{[-re|--recurse-paths --encrypt]}} {{path/to/
compressed.zip}} {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Archive files/directories to a multi-part split Zip archive (e.g. 3 GB parts):

```
zip {{[-rs|--recurse-paths --split-size]}} {{3g}} {{path/to/
compressed.zip}} {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Print a specific archive contents:

```
zip {{[-sf|--split-size --freshen]}} {{path/to/
compressed.zip}}
```

zip2john

Extract password hashes from Zip archives for use with John the Ripper password cracker.

This is a utility tool usually installed as part of the John the Ripper installation.

More information: https://www.openwall.com/john/.

- Extract the password hash from an archive, listing all files in the archive:

```
zip2john {{path/to/file.zip}}
```

- Extract the password hash using [o]nly a specific compressed file:

```
zip2john -o {{path/to/compressed_file}} {{path/to/file.zip}}
```

- Extract the password hash from a compressed file to a specific file (for use with John the Ripper):

```
zip2john -o {{path/to/compressed_file}} {{path/to/file.zip}}
> {{file.hash}}
```

zipalign

Zip archive alignment tool.

Part of the Android SDK build tools.

More information: https://developer.android.com/tools/zipalign.

- Align the data of a Zip file on 4-byte boundaries:

```
zipalign {{4}} {{path/to/input.zip}} {{path/to/output.zip}}
```

- Check that a Zip file is correctly aligned on 4-byte boundaries and display the results in a verbose manner:

```
zipalign -v -c {{4}} {{path/to/input.zip}}
```

zipcloak

Encrypt the contents within a Zip archive.

More information: https://manned.org/zipcloak.

- Encrypt the contents of a Zip archive:

```
zipcloak {{path/to/archive.zip}}
```

- Decrypt the contents of a Zip archive:

```
zipcloak {{[-d|--decrypt]}} {{path/to/archive.zip}}
```

- Output the encrypted contents into a new Zip archive:

```
zipcloak {{path/to/archive.zip}} {{[-O|--output-file]}}
{{path/to/encrypted.zip}}
```

zipgrep

Find patterns in files in a Zip archive using extended **regex** (supports **?**, **+**, **{}**, **()** and **|**).

More information: https://manned.org/zipgrep.

- Search for a pattern within a Zip archive:

```
zipgrep "{{search_pattern}}" {{path/to/file.zip}}
```

- Print file name and line number for each match:

```
zipgrep {{[-H|--with-filename]}} {{[-n|--line-number]}}
"{{search_pattern}}" {{path/to/file.zip}}
```

- Search for lines that do not match a pattern:

```
zipgrep {{[-v|--invert-match]}} "{{search_pattern}}" {{path/
to/file.zip}}
```

- Specify files inside a Zip archive from search:

```
zipgrep "{{search_pattern}}" {{path/to/file.zip}} {{file/to/
search1}} {{file/to/search2}}
```

- Exclude files inside a Zip archive from search:

```
zipgrep "{{search_pattern}}" {{path/to/file.zip}} {{[-x|--
line-regexp]}} {{file/to/exclude1}} {{file/to/exclude2}}
```

zipinfo

List detailed information about the contents of a Zip file.

More information: https://manned.org/zipinfo.

- List all files in a Zip file in long format (permissions, ownership, size, and modification date):

```
zipinfo {{path/to/archive.zip}}
```

- List all files in a Zip file:

```
zipinfo -1 {{path/to/archive.zip}}
```

zipnote

View, add, or edit a Zip archive's comments.

Files can also be renamed in the Zip archive.

More information: https://manned.org/zipnote.

- View the comments on a Zip archive:

```
zipnote {{path/to/file.zip}}
```

- Extract the comments on a Zip archive to a file:

```
zipnote {{path/to/file.zip}} > {{path/to/file.txt}}
```

- Add/Update comments in a Zip archive from a file:

```
zipnote < {{path/to/file.txt}} -w {{path/to/file.zip}}
```

zless

View **`gzip`** and **`xz`** compressed files.

More information: https://manned.org/zless.

- Page through a `gzip` compressed file with `less`:

```
zless {{file.txt.gz}}
```

zlib-flate

Raw zlib compression and decompression program.

Part of **`qpdf`**.

More information: https://manned.org/zlib-flate.

- Compress a file:

```
zlib-flate < {{path/to/input_file}} -compress > {{path/to/compressed.zlib}}
```

- Uncompress a file:

```
zlib-flate < {{path/to/compressed.zlib}} -uncompress > {{path/to/output_file}}
```

- Compress a file with a specified compression level. 0=Fastest (Worst), 9=Slowest (Best):

```
zlib-flate < {{path/to/input_file}} -compress={{compression_level}} > {{path/to/compressed.zlib}}
```

zm

A tool for managing articles of newspapers and blogs.

More information: https://github.com/ZERMZeitung/zm2.

- Make a new draft:

  ```
  zm new
  ```

- Edit a draft:

  ```
  zm edit
  ```

- Publish a draft and commit it with git:

  ```
  zm publish
  ```

zmap

Fast, open-source network scanner for Internet-wide surveys.

More information: https://manned.org/zmap.

- Scan a subnet or full IPv4 space for a specific TCP port (default: 80):

```
zmap {{SUBNETS}} {{[-p|--target-ports]}} {{port}}
```

- Scan specific ports or port ranges across a subnet:

```
zmap {{[-p|--target-ports]}} {{port1,port2-port3,...}}
{{SUBNETS}}
```

- Output results to a CSV file with custom fields:

```
zmap {{[-o|--output-file]}} {{path/to/output_file.csv}} {{[-
f|--output-fields]}}
"{{saddr,daddr,sport,dport}}" {{SUBNETS}}
```

- Limit the scan rate to a specific number of packets per second:

```
zmap {{[-r|--rate]}} {{packets_per_second}} {{SUBNETS}}
```

- Perform a dry run without sending packets:

```
zmap {{[-d|--dryrun]}} {{SUBNETS}}
```

- Exclude subnets using a blocklist file in CIDR notation:

```
zmap {{[-b|--blocklist-file]}} {{path/to/blocklist.txt}}
{{SUBNETS}}
```

- Set a specific source IP for scan packets:

```
zmap {{[-S|--source-ip]}} {{source_ip}} {{SUBNETS}}
```

- Cap the number/percentage of targets to probe (e.g. 1000 IP/port pairs):

```
zmap {{[-n|--max-targets]}} {{1000}} {{SUBNETS}} {{[-p|--
target-ports]}} {{port1,port2-port3}}
```

zmore

View **gzip** compressed files with **more**.

More information: https://manned.org/zmore.

- Open a compressed file:

  `zmore {{path/to/file.txt.gz}}`

- Display the next page of the file:

  `<Space>`

- Search for a pattern in the file (press `<n>` to go to next match):

  `</>{{regex}}`

- Exit:

  `<q>`

- Display interactive command help:

  `<h>`

zmv

Move or rename files matching a specified extended glob pattern.

See also: **`zcp`**, **`zln`**.

More information: https://zsh.sourceforge.net/Doc/Release/User-Contributions.html.

- Move files using a `regex`-like pattern:

```
zmv '{{(*).log}}' '{{$1.txt}}'
```

- Preview the result of a move, without making any actual changes:

```
zmv -n '{{(*).log}}' '{{$1.txt}}'
```

- Interactively move files, with a prompt before every change:

```
zmv -i '{{(*).log}}' '{{$1.txt}}'
```

- Verbosely print each action as it's being executed:

```
zmv -v '{{(*).log}}' '{{$1.txt}}'
```

znew

Recompress files from `.Z` to gzip format.

More information: https://manned.org/znew.

- Recompress a file from `.Z` to gzip format:

  `znew {{path/to/file1.Z}}`

- Recompress multiple files and display the achieved size reduction % per file:

  `znew -v {{path/to/file1.Z path/to/file2.Z ...}}`

- Recompress a file using the slowest compression method (for optimal compression):

  `znew -9 {{path/to/file1.Z}}`

- Recompress a file, [K]eeping the `.Z` file if it is smaller than the gzip file:

  `znew -K {{path/to/file1.Z}}`

zola

A static site generator in a single binary with everything built-in.

More information: https://www.getzola.org/documentation/getting-started/cli-usage/.

- Create the directory structure used by Zola at the given directory:

```
zola init {{my_site}}
```

- Build the whole site in the `public` directory after deleting it:

```
zola build
```

- Build the whole site into a different directory:

```
zola build --output-dir {{path/to/output_directory}}/
```

- Build and serve the site using a local server (default is `127.0.0.1:1111`):

```
zola serve
```

- Build all pages just like the build command would, but without writing any of the results to disk:

```
zola check
```

zopflipng

PNG compression utility.

More information: https://github.com/google/zopfli.

- Optimize a PNG:

```
zopflipng {{input.png}} {{output.png}}
```

- Optimize several PNGs and save with given prefix:

```
zopflipng --prefix={{prefix}} {{image1.png image2.png
image3.png ...}}
```

zotero

Manage your bibliographies.

More information: https://www.zotero.org/support.

- Run with the GUI:

```
zotero
```

- Run in headless mode:

```
zotero --headless
```

- Run with a specific profile:

```
zotero -P {{profile}}
```

- Run the Migration Assistant:

```
zotero --migration
```

zoxide

Keep track of the most frequently used directories.

Uses a ranking algorithm to navigate to the best match.

More information: https://github.com/ajeetdsouza/zoxide.

- Go to the highest-ranked directory that contains `string` in the name:

```
zoxide query string
```

- Go to the highest-ranked directory that contains `string1` and then `string2`:

```
zoxide query string1 string2
```

- Start an interactive directory search (requires `fzf`):

```
zoxide query --interactive
```

- Add a directory or increment its rank:

```
zoxide add {{path/to/directory}}
```

- Remove a directory from `zoxide`'s database:

```
zoxide remove {{path/to/directory}}
```

- Generate shell configuration for command aliases (`z`, `zi`):

```
zoxide init {{bash|elvish|fish|nushell|posix|powershell|tcsh|xonsh|zsh}}
```

zpaq

Incremental journaling backup utility and archiver.

More information: https://mattmahoney.net/dc/zpaqdoc.html.

- Add a file or directory to a new or existing archive:

```
zpaq {{[a|add]}} {{path/to/archive.zpaq}} {{path/to/file_or_directory}}
```

- Create or add to an encrypted archive:

```
zpaq {{[a|add]}} -k{{password}} {{path/to/archive.zpaq}} {{path/to/file_or_directory}}
```

- Extract the most recent versions of files:

```
zpaq {{[x|extract]}} {{path/to/archive.zpaq}}
```

- List the archive contents:

```
zpaq {{[l|list]}} {{path/to/archive.zpaq}}
```

- Set the level of compression (higher means more compression but slower):

```
zpaq {{[a|add]}} {{path/to/archive.zpaq}} -m{{1|2|3|4|5}} {{path/to/file_or_directory}}
```

- Extract the specified files from the archive that are not newer than the specified date:

```
zpaq {{[x|extract]}} {{path/to/archive.zpaq}} {{path/in/archive/to/extract}} -to {{path/to/output}} -until {{YYYY-MM-DD}}
```

zpool

Manage ZFS pools.

More information: https://manned.org/zpool.

- Show the configuration and status of all ZFS zpools:

```
zpool status
```

- Check a ZFS pool for errors (verifies the checksum of EVERY block). Very CPU and disk intensive:

```
zpool scrub {{pool_name}}
```

- List zpools available for import:

```
zpool import
```

- Import a zpool:

```
zpool import {{pool_name}}
```

- Export a zpool (unmount all filesystems):

```
zpool export {{pool_name}}
```

- Show the history of all pool operations:

```
zpool history {{pool_name}}
```

- Create a mirrored pool:

```
zpool create {{pool_name}} mirror {{disk1}} {{disk2}} mirror {{disk3}} {{disk4}}
```

- Add a cache (L2ARC) device to a zpool:

```
zpool add {{pool_name}} cache {{cache_disk}}
```

zrok

Expose local services and files to the internet.

Part of the OpenZiti project, offering secure, zero-trust sharing.

More information: https://docs.zrok.io.

- Request an invitation to use the public zrok service (run this first):

  ```
  zrok invite
  ```

- Enable the zrok environment with the token from an invitation email:

  ```
  zrok enable {{your_token}}
  ```

- Create a publicly accessible URL for a local web server:

  ```
  zrok share public {{http://localhost:8080}}
  ```

- Create a secure share accessible only with a unique token:

  ```
  zrok share private {{http://localhost:3000}}
  ```

- Access a private share created by another user:

  ```
  zrok access private {{share_token}}
  ```

- Serve the contents of a local directory as a simple website:

  ```
  zrok share public --backend-mode web {{path/to/directory}}
  ```

- Display the status of the zrok environment and active shares:

  ```
  zrok status
  ```

zrun

Transparently uncompress argument files to a command.

More information: https://manned.org/zrun.

- Run the specified command with uncompressed versions of the compressed argument files:

```
zrun {{cat}} {{path/to/file1.gz path/to/file2.bz2 ...}}
```

zsh

Z SHell, a Bash-compatible command-line interpreter.

See also: **`bash`**, **`histexpand`**.

More information: https://zsh.sourceforge.io/Doc/Release/Invocation.html#Invocation.

- Start an interactive shell session:

```
zsh
```

- Execute specific [c]ommands:

```
zsh -c "{{echo Hello world}}"
```

- Execute a specific script:

```
zsh {{path/to/script.zsh}}
```

- Check a specific script for syntax errors without executing it:

```
zsh --no-exec {{path/to/script.zsh}}
```

- Execute specific commands from `stdin`:

```
{{echo Hello world}} | zsh
```

- Execute a specific script, printing each command in the script before executing it:

```
zsh --xtrace {{path/to/script.zsh}}
```

- Start an interactive shell session in verbose mode, printing each command before executing it:

```
zsh --verbose
```

- Execute a specific command inside `zsh` with disabled glob patterns:

```
noglob {{command}}
```

zstd

Compress or decompress files with Zstandard compression.

More information: https://github.com/facebook/zstd.

- Compress a file into a new file with the `.zst` suffix:

```
zstd {{path/to/file}}
```

- Decompress a file:

```
zstd --decompress {{path/to/file.zst}}
```

- Decompress to `stdout`:

```
zstd --decompress --stdout {{path/to/file.zst}}
```

- Compress a file specifying the compression level, where 1=fastest, 19=slowest and 3=default:

```
zstd -{{level}} {{path/to/file}}
```

- Unlock higher compression levels (up to 22) using more memory (both for compression and decompression):

```
zstd --ultra -{{level}} {{path/to/file}}
```

zstdcat

This command is an alias of **`zstd --decompress --stdout`**.

- View documentation for the original command:

```
tldr zstd
```

zstdless

Open a **`zstd`** compressed file for interactive reading, allowing scrolling and search.

See also: **`zstd`**, **`less`**.

More information: https://manned.org/zstdless.

- Open a `zstd` compressed file:

```
zstdless {{path/to/file.zst}}
```

zstdmt

This command is an alias of **`zstd --threads 0`** (which sets the number of working threads to the number of physical CPU cores).

- View documentation for the original command:

```
tldr zstd
```

zsteg

Steganography detection tool for PNG and BMP file formats.

It detects LSB steganography, ZLIB-compressed data, OpenStego, Camouflage and LSB with the Eratosthenes set.

More information: https://github.com/zed-0xff/zsteg.

- Detect embedded data in a PNG:

```
zsteg {{path/to/image.png}}
```

- Detect embedded data in a BMP image, using all known methods:

```
zsteg --all {{path/to/image.bmp}}
```

- Detect embedded data in a PNG, iterating pixels vertically and using MSB first:

```
zsteg --msb --order yx {{path/to/image.png}}
```

- Detect embedded data in a BMP image, specifying the bits to consider:

```
zsteg --bits {{1,2,3|1-3}} {{path/to/image.bmp}}
```

- Detect embedded data in a PNG, extracting only prime pixels and inverting bits:

```
zsteg --prime --invert {{path/to/image.png}}
```

- Detect embedded data in a BMP image, specifying the minimum length of the strings to be found and the find mode:

```
zsteg --min-str-len {{10}} --strings {{first|all|longest|
none}} {{path/to/image.bmp}}
```

{

Multipurpose shell syntax.

More information: https://www.gnu.org/software/bash/manual/bash.html.

- Isolate variable names:

```
echo ${{{HOME}work}}
```

- Brace expand sequences:

```
echo {{{1..3}}} {{{a..c}}}{{{dir1,dir2,dir3}}}
```

- Check if `variable` is set before returning text:

```
echo ${{{variable:+variable is set and contains $variable}}}
```

- Set default values in case `variable` is unset:

```
echo ${{{variable:-default}}}
```

- Return `variable` length in characters:

```
echo ${{{#variable}}}
```

- Return a string slice:

```
echo ${{{variable:3:7}}}
```

- Recursively expand a `variable`:

```
echo ${{{!variable}}}
```

- Group command output together:

```
{ {{command1; command2; ...}} } | {{another_command}}
```

}

This shell keyword is used to close out **{**.

- View documentation for the { keyword:

```
tldr {
```

~

Expand to a directory.

More information: https://gnu.org/software/bash/manual/bash.html#Tilde-Expansion.

- List the current user's home directory contents:

```
ls ~
```

- List the home directory contents of another user:

```
ls ~{{username}}
```

- List the contents of the previous directory you were in:

```
ls ~-
```

Dos

CD

Change the current working directory.

More information: https://help.fdos.org/en/hhstndrd/command/cd.htm.

- Go to the specified directory:

```
CD {{PATH\TO\DIRECTORY}}
```

- Go up to the parent of the current directory:

```
CD ..
```

- Go to the root directory of the current drive:

```
CD \
```

CHDIR

This command is an alias of **CD**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} dos cd
```

CLS

Clear the screen and reset colors.

More information: https://help.fdos.org/en/hhstndrd/batch/cls.htm.

- Clear the screen resetting colors to white on black:

```
CLS
```

PATH

Display or set the search path for executable files.

This command is available in **FreeDOS** only.

More information: https://help.fdos.org/en/hhstndrd/batch/path.htm.

- Display the current search path:

```
PATH
```

- Set the semicolon-separated list of directories to search in:

```
PATH {{path/to/directory1;path/to/directory2;...}}
```

- Append a directory to the search path:

```
PATH %PATH%;{{path/to/directory}}
```

- Add a directory to the beginning of the search path:

```
PATH {{path/to/directory}};%PATH%
```

- Clear the search path:

```
PATH;
```

- Display help:

```
PATH /?
```

Freebsd

base64

Encode or decode file or **`stdin`** to/from base64, to **`stdout`** or another file.

More information: https://man.freebsd.org/cgi/man.cgi?query=base64.

- Encode a file to `stdout`:

```
base64 {{[-i|--input]}} {{path/to/file}}
```

- Encode a file to the specified output file:

```
base64 {{[-i|--input]}} {{path/to/input_file}} {{[-o|--output]}} {{path/to/output_file}}
```

- Wrap encoded output at a specific width (`0` disables wrapping):

```
base64 {{[-b|--break]}} {{0|76|...}} {{path/to/file}}
```

- Decode a file to `stdout`:

```
base64 {{[-d|--decode]}} {{[-i|--input]}} {{path/to/file}}
```

- Encode from `stdin` to `stdout`:

```
{{command}} | base64
```

- Decode from `stdin` to `stdout`:

```
{{command}} | base64 {{[-d|--decode]}}
```

cal

Display a calendar with the current day highlighted.

More information: https://man.freebsd.org/cgi/man.cgi?cal.

- Display a calendar for the current month:

```
cal
```

- Display a calendar for a specific year:

```
cal {{year}}
```

- Display a calendar for a specific month and year:

```
cal {{month}} {{year}}
```

- Display the whole calendar for the current year:

```
cal -y
```

- Don't [h]ighlight today and display [3] months spanning the date:

```
cal -h -3 {{month}} {{year}}
```

- Display the 2 months [B]efore and 3 [A]fter a specific [m]onth of the current year:

```
cal -A 3 -B 2 {{month}}
```

- Display [j]ulian days (starting from one, numbered from January 1):

```
cal -j
```

chfn

This command is an alias of **chpass**.

- View documentation for the original command:

```
tldr chpass
```

chpass

Add or change user database information, including login shell and password.

See also: **passwd**.

More information: https://man.freebsd.org/cgi/man.cgi?chpass.

- Add or change user database information for the current user interactively:

```
su -c chpass
```

- Set a specific login [s]hell for the current user:

```
chpass -s {{path/to/shell}}
```

- Set a login [s]hell for a specific user:

```
chpass -s {{path/to/shell}} {{username}}
```

- Change the account [e]xpire time (in seconds from the epoch, UTC):

```
su -c 'chpass -e {{time}} {{username}}'
```

- Change a user's password:

```
su -c 'chpass -p {{encrypted_password}} {{username}}'
```

- Specify the [h]ostname or address of an NIS server to query:

```
su -c 'chpass -h {{hostname}} {{username}}'
```

- Specify a particular NIS [d]omain (system domain name by default):

```
su -c 'chpass -d {{domain}} {{username}}'
```

chsh

This command is an alias of **chpass**.

- View documentation for the original command:

```
tldr chpass
```

df

Display an overview of the filesystem disk space usage.

More information: https://man.freebsd.org/cgi/man.cgi?df.

- Display all filesystems and their disk usage (using 512-byte units):

```
df
```

- Use [h]uman-readable units (based on powers of 1024) and display a grand total:

```
df -h -c
```

- Use [H]uman-readable units (based on powers of 1000):

```
df -{{-si|H}}
```

- Display the filesystem containing the specified file or directory:

```
df {{path/to/file_or_directory}}
```

- Include statistics on the number of free and used [i]nodes, including the filesystem [T]ypes:

```
df -iT
```

- Use [k]ibibyte (1024 byte) units when showing size figures:

```
df -k
```

- Display information in a [P]ortable way:

```
df -P
```

ipmitool

Interface with the Intelligent Platform Management Interface (IPMI).

More information: https://man.freebsd.org/cgi/man.cgi?query=ipmitool.

- Load the IPMI kernel module for local connections:

```
kldload ipmi.ko
```

- Open IPMI shell on the local hardware:

```
ipmitool shell
```

- Open IPMI shell on a remote host:

```
ipmitool -H {{ip_address}} -U {{user_name}} shell
```

look

Display lines beginning with a prefix in a sorted file.

See also: **`grep`**, **`sort`**.

More information: https://man.freebsd.org/cgi/man.cgi?look.

- Search for lines beginning with a specific prefix in a specific file:

```
look {{prefix}} {{path/to/file}}
```

- Case-insensitively search only on alphanumeric characters:

```
look {{[-f|--ignore-case]}} {{[-d|--alphanum]}} {{prefix}} {{path/to/file}}
```

- Specify a string termination character (space by default):

```
look {{[-t|--terminate]}} {{,}}
```

- Search in `/usr/share/dict/words` (`--ignore-case` and `--alphanum` are assumed):

```
look {{prefix}}
```

pkg

FreeBSD package manager.

More information: https://man.freebsd.org/cgi/man.cgi?pkg.

- Install a new package:

```
pkg install {{package}}
```

- Delete a package:

```
pkg delete {{package}}
```

- Upgrade all packages:

```
pkg upgrade
```

- Search for a package:

```
pkg search {{keyword}}
```

- List installed packages:

```
pkg info
```

- Remove unneeded dependencies:

```
pkg autoremove
```

procstat

Display detailed information about processes in FreeBSD.

More information: https://man.freebsd.org/cgi/man.cgi?query=procstat.

- Display file descriptors of a specific process:

```
procstat fds {{pid}}
```

- Show virtual memory mappings of a process:

```
procstat vm {{pid}}
```

- Display process arguments:

```
procstat arguments {{pid}}
```

- Show resource limits of a process:

```
procstat rlimit {{pid}}
```

sed

Edit text in a scriptable manner.

See also: **awk**, **ed**.

More information: https://www.freebsd.org/cgi/man.cgi?sed.

- Replace all `apple` (basic `regex`) occurrences with `mango` (basic `regex`) in all input lines and print the result to `stdout`:

```
{{command}} | sed 's/apple/mango/g'
```

- Execute a specific script [f]ile and print the result to `stdout`:

```
{{command}} | sed -f {{path/to/script.sed}}
```

- Delay opening each file until a command containing the related `w` function or flag is applied to a line of input:

```
{{command}} | sed -fa {{path/to/script.sed}}
```

- Replace all `apple` (extended `regex`) occurrences with `APPLE` (extended `regex`) in all input lines and print the result to `stdout`:

```
{{command}} | sed -E 's/(apple)/\U\1/g'
```

- Print just a first line to `stdout`:

```
{{command}} | sed -n '1p'
```

- Replace all `apple` (basic `regex`) occurrences with `mango` (basic `regex`) in a specific file and overwrite the original file in place:

```
sed -i 's/apple/mango/g' {{path/to/file}}
```

sockstat

List open Internet or UNIX domain sockets.

More information: https://man.freebsd.org/cgi/man.cgi?sockstat.

- View which users/processes are [l]istening on which ports:

```
sockstat -l
```

- Show information for IPv[4]/IPv[6] sockets [l]istening on specific [p]orts using a specific [P]rotocol:

```
sockstat -{{4|6}} -l -P {{tcp|udp|sctp|divert}} -p {{port1,port2...}}
```

- Also show [c]onnected sockets, not resolving [n]umeric UIDs to user names and using a [w]ider field size:

```
sockstat -cnw
```

- Only show sockets that belong to a specific [j]ail ID or name in [v]erbose mode:

```
sockstat -jv
```

- Display the protocol [s]tate and the remote [U]DP encapsulation port number, if applicable (these are currently only implemented for SCTP and TCP):

```
sockstat -sU
```

- Display the [C]ongestion control module and the protocol [S]tack, if applicable (these are currently only implemented for TCP):

```
sockstat -CS
```

- Only show Internet sockets if the local and foreign addresses are not in the loopback network prefix 127.0.0.0/8, or do not contain the IPv6 loopback address ::1:

```
sockstat -L
```

- Do not show the header ([q]uiet mode), showing [u]nix sockets and displaying the `inp_gencnt`:

```
sockstat -qui
```

ypchfn

This command is an alias of **chpass**.

- View documentation for the original command:

```
tldr chpass
```

ypchpass

This command is an alias of **chpass**.

- View documentation for the original command:

  ```
  tldr chpass
  ```

ypchsh

This command is an alias of **chpass**.

- View documentation for the original command:

```
tldr chpass
```

Linux

a2disconf

Disable an Apache configuration file on Debian-based OSes.

More information: https://manned.org/a2disconf.8.

- Disable a configuration file:

```
sudo a2disconf {{configuration_file}}
```

- Don't show informative messages:

```
sudo a2disconf {{[-q|--quiet]}} {{configuration_file}}
```

a2dismod

Disable an Apache module on Debian-based OSes.

More information: https://manned.org/a2dismod.8.

- Disable a module:

```
sudo a2dismod {{module}}
```

- Don't show informative messages:

```
sudo a2dismod {{[-q|--quiet]}} {{module}}
```

a2dissite

Disable an Apache virtual host on Debian-based OSes.

More information: https://manned.org/a2dissite.8.

- Disable a virtual host:

```
sudo a2dissite {{virtual_host}}
```

- Don't show informative messages:

```
sudo a2dissite {{[-q|--quiet]}} {{virtual_host}}
```

a2enconf

Enable an Apache configuration file on Debian-based OSes.

More information: https://manned.org/a2enconf.8.

- Enable a configuration file:

```
sudo a2enconf {{configuration_file}}
```

- Don't show informative messages:

```
sudo a2enconf {{[-q|--quiet]}} {{configuration_file}}
```

a2enmod

Enable an Apache module on Debian-based OSes.

More information: https://manned.org/a2enmod.8.

- Enable a module:

```
sudo a2enmod {{module}}
```

- Don't show informative messages:

```
sudo a2enmod {{[-q|--quiet]}} {{module}}
```

a2ensite

Enable an Apache virtual host on Debian-based OSes.

More information: https://manned.org/a2ensite.8.

- Enable a virtual host:

```
sudo a2ensite {{virtual_host}}
```

- Don't show informative messages:

```
sudo a2ensite {{[-q|--quiet]}} {{virtual_host}}
```

a2query

Retrieve runtime configuration from Apache on Debian-based OSes.

More information: https://manned.org/a2query.

- List enabled Apache modules:

```
sudo a2query -m
```

- Check if a specific module is installed:

```
sudo a2query -m {{module_name}}
```

- List enabled virtual hosts:

```
sudo a2query -s
```

- Display the currently enabled Multi Processing Module:

```
sudo a2query -M
```

- Display Apache version:

```
sudo a2query -v
```

aa-audit

Set AppArmor security profiles to audit mode.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-audit.8.

- Set a profile to audit mode:

```
sudo aa-audit {{profile_name}}
```

- Set multiple profiles to audit mode:

```
sudo aa-audit {{profile1 profile2 ...}}
```

- Set a profile to audit mode from a specific directory:

```
sudo aa-audit {{[-d|--dir]}} /{{path/to/profiles}} {{profile_name}}
```

- Force audit mode even if already applied:

```
sudo aa-audit --force {{profile_name}}
```

- Set a profile to audit mode without reloading it:

```
sudo aa-audit --no-reload {{profile_name}}
```

- Remove audit mode for a profile:

```
sudo aa-audit {{[-r|--remove]}} {{profile_name}}
```

- Display help:

```
aa-audit {{[-h|--help]}}
```

aa-cleanprof

Clean AppArmor security profiles by removing unused rules.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-cleanprof.8.

- Clean a profile to remove unused rules:

```
sudo aa-cleanprof {{profile_name}}
```

- Clean multiple profiles at once:

```
sudo aa-cleanprof {{profile1 profile2 ...}}
```

- Specify the directory containing profiles:

```
sudo aa-cleanprof {{[-d|--dir]}} /{{path/to/profiles}} {{profile_name}}
```

- Run silently without prompts:

```
sudo aa-cleanprof {{[-s|--silent]}} {{profile_name}}
```

- Prevent profile reload after cleaning:

```
sudo aa-cleanprof --no-reload {{profile_name}}
```

- Display help:

```
aa-cleanprof {{[-h|--help]}}
```

aa-complain

Set an AppArmor policy to complain mode.

See also: **`aa-disable`**, **`aa-enforce`**, **`aa-status`**.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-complain.8.

- Set policy to complain mode:

```
sudo aa-complain {{path/to/profile1 path/to/profile2 ...}}
```

- Set policies to complain mode:

```
sudo aa-complain {{[-d|--dir]}} {{path/to/profiles}}
```

aa-decode

Decode AppArmor audit logs into a human-readable format.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-decode.8.

- Decode a hex string:

```
aa-decode {{hexstring}}
```

- Decode a log file:

```
sudo aa-decode {{logfile}}
```

- Decode logs from `stdin` (e.g., redirected file):

```
sudo aa-decode - < {{logfile}}
```

- Display help:

```
aa-decode {{[-h|--help]}}
```

aa-disable

Disable AppArmor security policies.

See also: **`aa-complain`**, **`aa-enforce`**, **`aa-status`**.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-disable.8.

- Disable profile:

```
sudo aa-disable {{path/to/profile1 path/to/profile2 ...}}
```

- Disable profiles in a directory (defaults to `/etc/apparmor.d`):

```
sudo aa-disable --dir {{path/to/profiles}}
```

aa-enforce

Set an AppArmor profile to enforce mode.

See also: **`aa-complain`**, **`aa-disable`**, **`aa-status`**.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-enforce.8.

- Enable profile:

```
sudo aa-enforce {{[-d|--dir]}} {{path/to/profile}}
```

- Enable profiles:

```
sudo aa-enforce {{path/to/profile1 path/to/profile2 ...}}
```

aa-genprof

Generate AppArmor security profiles by monitoring program behavior.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-genprof.8.

- Start generating a profile for a program:

```
sudo aa-genprof {{program_path}}
```

- Specify a custom directory for profiles:

```
sudo aa-genprof {{[-d|--dir]}} /{{path/to/profiles}} {{program_path}}
```

- Specify a custom logfile for profiling:

```
sudo aa-genprof {{[-f|--file]}} /{{path/to/logfile}} {{program_path}}
```

- Display help:

```
aa-genprof {{[-h|--help]}}
```

aa-logprof

Interactively update AppArmor security profiles based on logged violations.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-logprof.8.

- Interactively review and update profiles based on system logs:

```
sudo aa-logprof
```

- Use a specific directory for AppArmor profiles:

```
sudo aa-logprof {{[-d|--dir]}} /{{path/to/profiles}}
```

- Use a specific log file instead of the default:

```
sudo aa-logprof {{[-f|--file]}} /{{path/to/logfile}}
```

- Ignore all log entries before the specified mark:

```
sudo aa-logprof {{[-m|--logmark]}} "{{log_marker_text}}"
```

- Display help:

```
aa-logprof {{[-h|--help]}}
```

aa-mergeprof

Merge AppArmor security profile files into the profile directory.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-mergeprof.8.

- Merge one or more profile files into the default profile directory:

```
sudo aa-mergeprof {{file1 file2 ...}}
```

- Merge profile files into a specific directory:

```
sudo aa-mergeprof {{[-d|--dir]}} /{{path/to/profiles}}
{{file1 file2 ...}}
```

- Display help:

```
aa-mergeprof {{[-h|--help]}}
```

aa-remove-unknown

Remove AppArmor profiles that are no longer present in the configuration directory.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-remove-unknown.8.

- Perform a dry run to see which profiles would be removed:

```
sudo aa-remove-unknown -n
```

- Actually remove the profiles:

```
sudo aa-remove-unknown
```

- Display help:

```
aa-remove-unknown {{[-h|--help]}}
```

aa-status

List currently loaded AppArmor modules.

See also: **`aa-complain`**, **`aa-disable`**, **`aa-enforce`**.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-status.8.

- Check status:

```
sudo aa-status
```

- Display status in JSON format:

```
sudo aa-status --json
```

- Display status in pretty JSON format:

```
sudo aa-status --pretty-json
```

- Display the number of loaded policies:

```
sudo aa-status --profiled
```

- Display the number of loaded enforicing policies:

```
sudo aa-status --enforced
```

- Display the number of loaded non-enforcing policies:

```
sudo aa-status --complaining
```

- Display the number of loaded enforcing policies that kill tasks:

```
sudo aa-status --kill
```

aa-teardown

Tear down all AppArmor profiles and disable AppArmor enforcement.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-teardown.8.

- Disable all AppArmor profiles and stop enforcement:

```
sudo aa-teardown
```

- Display help:

```
aa-teardown {{[-h|--help]}}
```

aa-unconfined

List processes with open TCP/UDP ports that do not have AppArmor profiles loaded.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-unconfined.8.

- List unconfined processes using the `ss` command (default):

```
sudo aa-unconfined
```

- Use `netstat` instead of `ss` to detect open network sockets:

```
sudo aa-unconfined --with-netstat
```

- Show all processes from /proc with TCP/UDP ports and no AppArmor profiles (more detailed):

```
sudo aa-unconfined --paranoid
```

- Display help:

```
aa-unconfined {{[-h|--help]}}
```

aa-update-browser

Update AppArmor browser profiles to use supported abstractions.

Part of the AppArmor suite.

More information: https://manned.org/aa-update-browser.

- [l]ist available browser abstraction profiles:

```
sudo aa-update-browser -l
```

- Show what changes would be made to a profile without applying them ([d]ry-run):

```
sudo aa-update-browser -d {{path/to/profile}}
```

- [u]pdate a profile with specific abstractions:

```
sudo aa-update-browser -u {{abstraction1,abstraction2,...}} {{path/to/profile}}
```

- Remove all abstractions from a profile:

```
sudo aa-update-browser -u '' {{path/to/profile}}
```

- Display help:

```
aa-update-browser -h
```

abbr

Manage abbreviations for the fish shell.

User-defined words are replaced with longer phrases after they are entered.

More information: https://fishshell.com/docs/current/cmds/abbr.html.

- Add a new abbreviation:

```
abbr {{[-a|--add]}} {{abbreviation_name}} {{command}} {{command_arguments}}
```

- Rename an existing abbreviation:

```
abbr --rename {{old_name}} {{new_name}}
```

- Erase an existing abbreviation:

```
abbr {{[-e|--erase]}} {{abbreviation_name}}
```

- Import the abbreviations defined on another host over SSH:

```
ssh {{host_name}} abbr {{[-s|--show]}} | source
```

abroot

Utility providing full immutability and atomicity by transacting between 2 root partition states (A⟺B).

Updates are performed using OCI images, to ensure that the system is always in a consistent state.

More information: https://github.com/Vanilla-OS/ABRoot.

- Add packages to the local image (Note: After executing this command, you need to apply these changes.):

```
sudo abroot pkg add {{package}}
```

- Remove packages from the local image (Note: After executing this command, you need to apply these changes.):

```
sudo abroot pkg remove {{package}}
```

- List packages in the local image:

```
sudo abroot pkg list
```

- Apply changes in the local image (Note: You need to reboot your system for these changes to be applied):

```
sudo abroot pkg apply
```

- Rollback your system to previous state:

```
sudo abroot rollback
```

- Edit/View kernel parameters:

```
sudo abroot kargs {{edit|show}}
```

- Display status:

```
sudo abroot status
```

- Display help:

```
abroot {{[-h|--help]}}
```

abrt-action-analyze-backtrace

Analyze C/C++ backtrace.

Generate duplication hash, backtrace rating, and identify crash function.

Save the data as new elements **`duphash`**, **`rating`**, **`crash_function`** in the problem directory.

More information: https://manned.org/abrt-action-analyze-backtrace.

- Analyze backtrace for the current working directory:

```
abrt-action-analyze-backtrace
```

- Analyze backtrace for a specific directory:

```
abrt-action-analyze-backtrace -d {{path/to/directory}}
```

- Analyze backtrace verbosely:

```
abrt-action-analyze-backtrace -v
```

abrt-action-analyze-c

Calculate UUID for a problem data directory with **`coredump`**.

More information: https://manned.org/abrt-action-analyze-c.

- Calculate and save the UUID for the current working directory:

```
abrt-action-analyze-c
```

- Calculate and save the UUID for a specific directory:

```
abrt-action-analyze-c -d {{path/to/directory}}
```

- Calculate and save the UUID verbosely:

```
abrt-action-analyze-c -v
```

abrt-cli

Automatic Bug Reporting Tool for Fedora-based systems.

Used to detect, analyze, and report application crashes.

More information: https://abrt.readthedocs.io/en/latest/usage.html.

- List detected problems:

```
abrt-cli list
```

- Show details of a specific problem:

```
abrt-cli info {{problem_id}}
```

- Remove a crash report:

```
abrt-cli remove {{problem_id}}
```

- Report a problem to the configured bug tracker (e.g. Bugzilla):

```
abrt-cli report {{problem_id}}
```

- Monitor a log file and trigger a program when a match is found:

```
abrt-watch-log -F {{error_string}} {{/var/log/myapp.log}}
{{notify-send "Crash detected"}}
```

- Generate a report for debugging manually:

```
abrt-cli report {{[-a|--analyze]}} {{problem_id}}
```

abrt

This command is an alias of **`abrt-cli`**.

- View documentation for the original command:

  `tldr abrt-cli`

ac

Print statistics on how long users have been connected.

More information: https://www.gnu.org/software/acct/manual/accounting.html#ac.

- Print how long the current user has been connected in hours:

```
ac
```

- Print how long users have been connected in hours:

```
ac {{[-p|--individual-totals]}}
```

- Print how long a particular user has been connected in hours:

```
ac {{[-p|--individual-totals]}} {{username}}
```

- Print how long a particular user has been connected in hours per day (with total):

```
ac {{[-d|--daily-totals]}} {{[-p|--individual-totals]}} {{username}}
```

- Also display additional details:

```
ac --compatibility
```

acountry

Print the country where an IPv4 address or hostname is located.

More information: https://manned.org/acountry.

- Print a country where an IPv4 address or host is located:

```
acountry {{example.com}}
```

- Print extra [d]ebugging output:

```
acountry -d {{example.com}}
```

- Print more [v]erbose information:

```
acountry -v {{example.com}}
```

acpi

Shows battery status or thermal information.

More information: https://manned.org/acpi.

- Show battery information:

```
acpi
```

- Show thermal information:

```
acpi {{[-t|--thermal]}}
```

- Show cooling device information:

```
acpi {{[-c|--cooling]}}
```

- Show thermal information in Fahrenheit:

```
acpi {{[-tf|--thermal --fahrenheit]}}
```

- Show all information:

```
acpi {{[-V|--everything]}}
```

- Extract information from `/proc` instead of `/sys`:

```
acpi {{[-p|--proc]}}
```

acpi_listen

Listen to ACPI events.

More information: https://manned.org/acpi_listen.

- Listen to any ACPI event while the daemon is running:

```
acpi_listen
```

- Display help:

```
acpi_listen {{[-h|--help]}}
```

add-apt-repository

Manage **apt** repository definitions.

More information: https://manned.org/add-apt-repository.

- Add a new apt repository:

```
add-apt-repository {{repository_spec}}
```

- Remove an apt repository:

```
add-apt-repository {{[-r|--remove]}} {{repository_spec}}
```

- Update the package cache after adding a repository:

```
add-apt-repository --update {{repository_spec}}
```

- Allow source packages to be downloaded from the repository:

```
add-apt-repository {{[-s|--enable-source]}}
{{repository_spec}}
```

addpart

Tell the Linux kernel about the existence of the specified partition.

A simple wrapper around the **`add partition`** ioctl.

More information: https://manned.org/addpart.

- Tell the kernel about the existence of the specified partition:

```
addpart {{device}} {{partition}} {{start}} {{length}}
```

addr2line

Convert addresses of a binary into file names and line numbers.

More information: https://manned.org/addr2line.

- Display the filename and line number of the source code from an instruction address of an executable:

```
addr2line {{[-e|--exe]}} {{path/to/executable}} {{address}}
```

- Display the function name, filename and line number:

```
addr2line {{[-e|--exe]}} {{path/to/executable}} {{[-f|--
functions]}} {{address}}
```

- Demangle the function name for C++ code:

```
addr2line {{[-e|--exe]}} {{path/to/executable}} {{[-f|--
functions]}} {{[-C|--demangle]}} {{address}}
```

adduser

User addition utility.

More information: https://manned.org/adduser.

- Create a new user with a default home directory and prompt the user to set a password:

```
adduser {{username}}
```

- Create a new user without a home directory:

```
adduser --no-create-home {{username}}
```

- Create a new user with a home directory at the specified path:

```
adduser --home {{path/to/home}} {{username}}
```

- Create a new user with the specified shell set as the login shell:

```
adduser --shell {{path/to/shell}} {{username}}
```

- Create a new user belonging to the specified group:

```
adduser --ingroup {{group}} {{username}}
```

adig

Print information received from Domain Name System (DNS) servers.

More information: https://manned.org/adig.

- Display A (default) record from DNS for hostname(s):

```
adig {{example.com}}
```

- Display extra [d]ebugging output:

```
adig -d {{example.com}}
```

- Connect to a specific DNS [s]erver:

```
adig -s {{1.2.3.4}} {{example.com}}
```

- Use a specific TCP port to connect to a DNS server:

```
adig -T {{port}} {{example.com}}
```

- Use a specific UDP port to connect to a DNS server:

```
adig -U {{port}} {{example.com}}
```

agetty

Alternative **`getty`**: Open a **`tty`** port, prompt for a login name, and invoke the **`/bin/login`** command.

It is normally invoked by **`init`**.

Note: The baud rate is the speed of data transfer between a terminal and a device over a serial connection.

More information: https://manned.org/agetty.

- Connect `stdin` to a port (relative to `/dev`) and optionally specify a baud rate (defaults to 9600):

```
agetty {{tty}} {{115200}}
```

- Assume `stdin` is already connected to a `tty` and set a timeout for the login:

```
agetty {{[-t|--timeout]}} {{timeout_in_seconds}} -
```

- Assume the `tty` is 8-bit, overriding the `TERM` environment variable set by `init`:

```
agetty {{[-8|--8bits]}} - {{term_var}}
```

- Skip the login (no login) and invoke, as root, another login program instead of `/bin/login`:

```
agetty {{[-n|--skip-login]}} {{[-l|--login-program]}} {{login_program}} {{tty}}
```

- Do not display the pre-login (issue) file (`/etc/issue` by default) before writing the login prompt:

```
agetty {{[-i|--noissue]}} -
```

- Change the root directory and write a specific fake host into the `utmp` file:

```
agetty {{[-r|--chroot]}} /{{path/to/root_directory}} {{[-H|--host]}} {{fake_host}} -
```

ahost

DNS lookup utility to display the A or AAAA record linked with a hostname or IP address.

More information: https://manned.org/ahost.

- Print an A or AAAA record associated with a hostname or IP address:

```
ahost {{example.com}}
```

- Display some extra debugging output:

```
ahost -d {{example.com}}
```

- Display the record with a specified type:

```
ahost -t {{a|aaaa|u}} {{example.com}}
```

aide

Advanced Intrusion Detection Environment to validate file integrity.

More information: https://manned.org/aide.

- Initialize the database:

```
sudo aide {{[-i|--init]}}
```

- Check the database for inconsistencies:

```
sudo aide {{[-C|--check]}}
```

- Compare two databases according to definitions in the config file:

```
sudo aide {{[-E|--compare]}}
```

- Check and update the database non-interactively:

```
sudo aide {{[-u|--update]}}
```

- Define a config file to override the default `./aide.conf`:

```
sudo aide {{[-c|--config]}} {{path/to/config_file}}
```

- Use `regex` to limit AIDE to a specific string:

```
sudo aide {{[-l|--limit]}} {{regex}}
```

- Send reporter results to a URL:

```
sudo aide {{[-r|--report]}} {{reporterurl}}
```

alien

Convert different installation packages to other formats.

See also: **`debtap`** for **`.deb`** conversion on Arch Linux.

More information: https://manned.org/alien.

- Convert a specific installation file to Debian format (`.deb` extension):

```
sudo alien {{[-d|--to-deb]}} {{path/to/file}}
```

- Convert a specific installation file to Red Hat format (`.rpm` extension):

```
sudo alien {{[-r|--to-rpm]}} {{path/to/file}}
```

- Convert a specific installation file to a Slackware installation file (`.tgz` extension):

```
sudo alien {{[-t|--to-tgz]}} {{path/to/file}}
```

- Convert a specific installation file to Debian format and install on the system:

```
sudo alien {{[-d|--to-deb]}} {{[-i|--install]}} {{path/to/file}}
```

alpine

An email client and Usenet newsgroup program with a pico/nano-inspired interface.

Supports most modern email services through IMAP.

More information: https://manned.org/alpine.

- Open alpine normally:

```
alpine
```

- Open alpine directly to the message composition screen to send an email to a given email address:

```
alpine {{email@example.net}}
```

- Quit alpine:

```
<q><y>
```

alsamixer

Graphical mixer for ALSA soundcard driver.

More information: https://manned.org/alsamixer.

- Select the soundcard to use:

```
alsamixer {{[-c|--card]}} {{soundcard_number}}
```

- Select mixer device to control:

```
alsamixer {{[-D|--device]}} {{device_name}}
```

- Increase volume:

```
<w>
```

- Decrease volume:

```
<x>
```

- Quit alsamixer:

```
<Esc>
```

alternatives

This command is an alias of **`update-alternatives`**.

- View documentation for the original command:

  ```
  tldr update-alternatives
  ```

amixer

Mixer for ALSA soundcard driver.

More information: https://manned.org/amixer.

- Turn up the master volume by 10%:

```
amixer -D pulse sset Master {{10%+}}
```

- Turn down the master volume by 10%:

```
amixer -D pulse sset Master {{10%-}}
```

anbox

Run Android applications on any Linux operating system.

More information: https://manned.org/anbox.

- Launch Anbox into the app manager:

```
anbox launch --package={{org.anbox.appmgr}} --
component={{org.anbox.appmgr.AppViewActivity}}
```

apache2ctl

Administrate the Apache HTTP web server.

This command comes with Debian based OSes, for RHEL based ones see **`httpd`**.

More information: https://manned.org/apache2ctl.8.

- Start the Apache daemon. Throw a message if it is already running:

```
sudo apache2ctl start
```

- Stop the Apache daemon:

```
sudo apache2ctl stop
```

- Restart the Apache daemon:

```
sudo apache2ctl restart
```

- Test syntax of the configuration file:

```
sudo apache2ctl -t
```

- List loaded modules:

```
sudo apache2ctl -M
```

apachectl

Control an Apache HTTP server.

More information: https://manned.org/apachectl.

- Start the server:

```
sudo apachectl start
```

- Restart the server:

```
sudo apachectl restart
```

- Stop the server:

```
sudo apachectl stop
```

- Test configuration file validity:

```
apachectl configtest
```

- Check server status (requires the lynx browser):

```
apachectl status
```

- Reload configuration without dropping connections:

```
sudo apachectl graceful
```

- Print full Apache configuration (not always supported):

```
apachectl -S
```

- Display help:

```
apachectl -h
```

apk

Alpine Linux package management tool.

More information: https://wiki.alpinelinux.org/wiki/Alpine_Package_Keeper.

- Update repository indexes and upgrade all packages:

```
apk upgrade {{[-U|--update-cache]}}
```

- Only update repository indexes:

```
apk update
```

- Install a new package:

```
apk add {{package}}
```

- Remove a package:

```
apk del {{package}}
```

- Repair/Reinstall a package without modifying main dependencies:

```
apk fix {{package}}
```

- Search for packages with a keyword in their name and list results with descriptions:

```
apk search {{[-v|--verbose]}} {{keyword}}
```

- Search for packages with a keyword in their description:

```
apk search {{[-d|--description]}} {{keyword}}
```

- Display information about a specific package:

```
apk info {{package}}
```

aplay

Sound player for ALSA soundcard driver.

More information: https://manned.org/aplay.

- Play a specific file (sampling rate, bit depth, etc. will be automatically determined for the file format):

```
aplay {{path/to/file}}
```

- Play the first 10 seconds of a specific file at 2500 Hz:

```
aplay {{[-d|--duration]}} {{10}} {{[-r|--rate]}} {{2500}} {{path/to/file}}
```

- Play the raw file as a 22050 Hz, mono, 8-bit, Mu-Law `.au` file:

```
aplay {{[-c|--channels]}} {{1}} {{[-t|--file-type]}} {{raw}} {{[-r|--rate]}} {{22050}} {{[-f|--format]}} {{mu_law}} {{path/to/file}}
```

- List available audio devices:

```
aplay {{[-l|--list-devices]}}
```

apparmor_parser

Load, compile, and manage AppArmor security profiles.

More information: https://gitlab.com/apparmor/apparmor/-/wikis/manpage_apparmor_parser.8.

- Load a profile into the kernel:

```
sudo apparmor_parser {{[-a|--add]}} {{profile_file}}
```

- Replace an existing profile:

```
sudo apparmor_parser {{[-r|--replace]}} {{profile_file}}
```

- Remove a profile from the kernel:

```
sudo apparmor_parser {{[-R|--remove]}} {{profile_name}}
```

- Load a profile in complain mode (logs violations but doesn't block):

```
sudo apparmor_parser {{[-C|--complain]}} {{[-r|--replace]}} {{path/to/profile}}
```

- Preprocess a profile (resolve includes) and write binary cache to file:

```
apparmor_parser {{[-p|--preprocess]}} {{[-o|--ofile]}} {{path/to/output.cache}} {{[-Q|--skip-kernel-load]}} {{path/to/profile}}
```

- Preprocess and print binary profile to `stdout` without loading:

```
apparmor_parser {{[-p|--preprocess]}} {{[-S|--stdout]}} {{[-Q|--skip-kernel-load]}} {{path/to/profile}}
```

- Replace a profile while skipping cache reads:

```
sudo apparmor_parser {{[-r|--replace]}} {{[-T|--skip-read-cache]}} {{path/to/profile}}
```

- Replace a profile, rebuild cache, and write it to a custom directory:

```
sudo apparmor_parser {{[-r|--replace]}} {{[-W|--write-cache]}} {{[-L|--cache-loc]}} /{{path/to/cache}} {{path/to/profile}}
```

apparmor_status

This command is an alias of **aa-status**.

- View documentation for the original command:

  ```
  tldr aa-status
  ```

apport-bug

File a bug report on Ubuntu.

More information: https://wiki.ubuntu.com/Apport.

- Report a bug about the whole system:

```
apport-bug
```

- Report a bug about a specific package:

```
apport-bug {{package}}
```

- Report a bug about a specific executable:

```
apport-bug {{path/to/executable}}
```

- Report a bug about a specific process:

```
apport-bug {{PID}}
```

apt-add-repository

This command is an alias of **`add-apt-repository`**.

- View documentation for the original command:

  ```
  tldr add-apt-repository
  ```

apt-cache

Debian and Ubuntu package query tool.

More information: https://manned.org/apt-cache.8.

- Search for a package in your current sources:

```
apt-cache search {{query}}
```

- Show information about a package:

```
apt-cache show {{package}}
```

- Show whether a package is installed and up to date:

```
apt-cache policy {{package}}
```

- Show dependencies for a package:

```
apt-cache depends {{package}}
```

- Show packages that depend on a particular package:

```
apt-cache rdepends {{package}}
```

apt-clone

Clone/backup/restore the package state of a Debian-based system.

More information: https://github.com/mvo5/apt-clone.

- Clone the package state of the current system into a specified directory:

```
apt-clone clone {{path/to/directory}}
```

- Create a clone file (`tar.gz`) for backup purposes:

```
apt-clone clone --destination {{path/to/backup.tar.gz}}
```

- Restore the package state from a clone file:

```
apt-clone restore {{path/to/backup.tar.gz}}
```

- Show information about a clone file (e.g., the release, architecture):

```
apt-clone info {{path/to/backup.tar.gz}}
```

- Check if the clone file can be restored on the current system:

```
apt-clone restore {{path/to/backup.tar.gz}} --destination
{{path/to/restore}}
```

apt-file

Search for files in **apt** packages, including ones not yet installed.

More information: https://manned.org/apt-file.1.

- Update the metadata database:

```
sudo apt update
```

- Search for packages that contain the specified file or path:

```
apt-file {{search|find}} {{path/to/file}}
```

- List the contents of a specific package:

```
apt-file {{show|list}} {{package}}
```

- Search for packages that match the regex:

```
apt-file {{search|find}} {{[-x|--regexp]}} {{regex}}
```

apt-get

Debian and Ubuntu package management utility.

Search for packages using **`apt-cache`**.

It is recommended to use **`apt`** when used interactively in Ubuntu versions 16.04 and later.

More information: https://manned.org/apt-get.8.

- Update the list of available packages and versions (it's recommended to run this before other `apt-get` commands):

```
sudo apt-get update
```

- Install a package, or update it to the latest available version:

```
sudo apt-get install {{package}}
```

- Remove a package:

```
sudo apt-get remove {{package}}
```

- Remove a package and its configuration files:

```
sudo apt-get purge {{package}}
```

- Upgrade all installed packages to their newest available versions:

```
sudo apt-get upgrade
```

- Clean the local repository - removing package files (`.deb`) from interrupted downloads that can no longer be downloaded:

```
apt-get autoclean
```

- Remove all packages that are no longer needed:

```
sudo apt-get autoremove
```

- Upgrade installed packages (like `upgrade`), but remove obsolete packages and install additional packages to meet new dependencies:

```
sudo apt-get dist-upgrade
```

apt install

Install packages for Debian-based distributions.

More information: https://manned.org/apt.8.

- Install a package, or update it to the latest version:

```
sudo apt install {{package}}
```

- Display verbose package version information during installation or update:

```
sudo apt install {{[-V|--verbose-versions]}} {{package}}
```

apt-key

Key management utility for the APT Package Manager on Debian and Ubuntu.

Note: **`apt-key`** is now deprecated (except for the use of **`apt-key del`** in maintainer scripts).

More information: https://manned.org/apt-key.8.

- List trusted keys:

```
apt-key list
```

- Add a key to the trusted keystore:

```
apt-key add {{public_key_file.asc}}
```

- Delete a key from the trusted keystore:

```
apt-key del {{key_id}}
```

- Add a remote key to the trusted keystore:

```
wget {{[-qO|--quiet --output-document]}} - {{https://
host.tld/filename.key}} | apt-key add -
```

- Add a key from keyserver with only key ID:

```
apt-key adv --keyserver {{pgp.mit.edu}} --recv {{KEYID}}
```

apt-mark

Utility to change the status of installed packages.

More information: https://manned.org/apt-mark.8.

- Mark a package as automatically installed:

```
sudo apt-mark auto {{package}}
```

- Hold a package at its current version and prevent updates to it:

```
sudo apt-mark hold {{package}}
```

- Allow a package to be updated again:

```
sudo apt-mark unhold {{package}}
```

- Show manually installed packages:

```
apt-mark showmanual
```

- Show held packages that aren't being updated:

```
apt-mark showhold
```

apt moo

An **APT** easter egg.

More information: https://manned.org/apt.8.

- Print a cow easter egg:

```
apt moo
```

apt

Package manager for Debian-based distributions.

Intended as a user-friendly alternative to **`apt-get`** for interactive use.

For equivalent commands in other package managers, see https://wiki.archlinux.org/title/Pacman/Rosetta.

More information: https://manned.org/apt.8.

- Update the list of available packages and versions (recommended before running other `apt` commands):

```
sudo apt update
```

- Search packages by name or description:

```
apt search {{package}}
```

- Search packages by name only (supports wildcards like `*`):

```
apt list {{package}}
```

- Show detailed information about a package:

```
apt show {{package}}
```

- Install a package, or update it to the latest version:

```
sudo apt install {{package}}
```

- Remove a package (use `purge` instead to also remove configuration files):

```
sudo apt remove {{package}}
```

- Upgrade all installed packages to their latest versions:

```
sudo apt upgrade
```

- List all installed packages:

```
apt list {{[-i|--installed]}}
```

aptitude

Debian and Ubuntu package management utility.

More information: https://manned.org/aptitude.8.

- Synchronize list of packages and versions available. This should be run first, before running subsequent `aptitude` commands:

```
sudo aptitude update
```

- Install a new package and its dependencies:

```
sudo aptitude install {{package}}
```

- Search for a package:

```
aptitude search {{package}}
```

- Search for an installed package (`?installed` is an `aptitude` search term):

```
aptitude search '?installed({{package}})'
```

- Remove a package and all packages depending on it:

```
sudo aptitude remove {{package}}
```

- Upgrade installed packages to the newest available versions:

```
sudo aptitude upgrade
```

- Upgrade installed packages (like `aptitude upgrade`) including removing obsolete packages and installing additional packages to meet new package dependencies:

```
sudo aptitude full-upgrade
```

- Put an installed package on hold to prevent it from being automatically upgraded:

```
sudo aptitude hold '?installed({{package}})'
```

apx pkgmanagers

Manage package managers in **apx**.

Note: User-created package manager configurations are stored in **~/.local/share/apx/pkgmanagers**.

More information: https://github.com/Vanilla-OS/apx.

- Interactively create a new package manager configuration:

```
apx pkgmanagers create
```

- List all available package manager confirgurations:

```
apx pkgmanagers list
```

- Remove a package manager configuration:

```
apx pkgmanagers rm --name {{string}}
```

- Display information about a specific package manager:

```
apx pkgmanagers show {{name}}
```

apx stacks

Manage stacks in **apx**.

Note: User-created stack configurations are stored in **~/.local/share/apx/stacks**.

More information: https://github.com/Vanilla-OS/apx.

- Interactively create a new stack configuration:

```
apx stacks new
```

- Interactively update a stack configuration:

```
apx stacks update {{name}}
```

- List all available stack configurations:

```
apx stacks list
```

- Remove a specified stack configuration:

```
apx stacks rm --name {{string}}
```

- Import a stack configuration:

```
apx stacks import --input {{path/to/stack.yml}}
```

- Export the stack configuration (Note: The output flag is optional, it is exported to the current working directory by default):

```
apx stacks export --name {{string}} --output {{path/to/output_file}}
```

apx subsystems

Manage subsystems in **apx**.

Subsystems are containers that can be created based on pre-existing stacks.

More information: https://docs.vanillaos.org/docs/en/apx-manpage#subsystems.

- Interactively create a new subsystem:

```
apx subsystems new
```

- List all available subsystems:

```
apx subsystems list
```

- Reset a specific subsystem to its initial state:

```
apx subsystems reset {{[-n|--name]}} {{string}}
```

- Force reset a specific subsystem:

```
apx subsystems reset {{[-n|--name]}} {{string}} {{[-f|--
force]}}
```

- Remove a specific subsystem:

```
apx subsystems rm {{[-n|--name]}} {{string}}
```

- Force remove a specific subsystem:

```
apx subsystems rm {{[-n|--name]}} {{string}} {{[-f|--force]}}
```

apx

Package management utility with support for multiple sources, allowing you to install packages in subsystems.

More information: https://github.com/Vanilla-OS/apx.

- View documentation for managing package managers:

```
tldr apx pkgmanagers
```

- View documentation for managing stacks:

```
tldr apx stacks
```

- View documentation for managing subsystems:

```
tldr apx subsystems
```

arch-chroot

Enhanced **`chroot`** command to help in the Arch Linux installation process.

More information: https://manned.org/arch-chroot.8.

- Start an interactive shell (Bash, by default) in a new root directory:

```
arch-chroot {{path/to/new_root}}
```

- Specify the user (other than the current user) to run the shell as:

```
arch-chroot -u {{user}} {{path/to/new_root}}
```

- Run a custom command (instead of the default Bash) in the new root directory:

```
arch-chroot {{path/to/new_root}} {{command}} {{command_arguments}}
```

- Specify the shell, other than the default Bash (in this case, the `zsh` package should have been installed in the target system):

```
arch-chroot {{path/to/new_root}} {{zsh}}
```

archey

Simple tool for stylishly displaying system information.

More information: https://lclarkmichalek.github.io/archey3/.

- Show system information:

```
archey
```

archinstall

Guided Arch Linux installer with a twist.

More information: https://archinstall.readthedocs.io.

- Start the interactive installer:

  ```
  archinstall
  ```

- Start a preset installer:

  ```
  archinstall {{minimal|unattended}}
  ```

archivemount

Mount an archive for access as a filesystem.

More information: https://manned.org/archivemount.

- Mount an archive to a specific mountpoint:

```
archivemount {{path/to/archive}} {{path/to/mount_point}}
```

archlinux-java

Switch between installed Java environments.

More information: https://wiki.archlinux.org/title/Java#Switching_between_JVM.

- List installed Java environments:

```
archlinux-java status
```

- Return the short name of the current default Java environment:

```
archlinux-java get
```

- Set the default Java environment:

```
archlinux-java set {{java_environment}}
```

- Unset the default Java environment:

```
archlinux-java unset
```

- Fix an invalid/broken default Java environment configuration:

```
archlinux-java fix
```

arecord

Sound recorder for ALSA soundcard driver.

More information: https://manned.org/arecord.

- Record a snippet in "CD" quality (finish with <Ctrl c> when done):

```
arecord {{[-vv|--verbose --verbose]}} {{[-f|--format]}} cd
{{path/to/file.wav}}
```

- Record a snippet in "CD" quality, with a fixed duration of 10 seconds:

```
arecord {{[-vv|--verbose --verbose]}} {{[-f|--format]}} cd
{{[-d|--duration]}} {{10}} {{path/to/file.wav}}
```

- Record a snippet and save it as an MP3 (finish with <Ctrl c> when done):

```
arecord {{[-vv|--verbose --verbose]}} {{[-f|--format]}} cd
{{[-t|--file-type]}} raw | lame -r - {{path/to/file.mp3}}
```

- List all sound cards and digital audio devices:

```
arecord {{[-l|--list-devices]}}
```

- Allow interactive interface (e.g. use <Space> or <Enter> to play or pause):

```
arecord {{[-i|--interactive]}}
```

- Test your microphone by recording a 5 second sample and playing it back:

```
arecord {{[-d|--duration]}} 5 test-mic.wav && aplay test-
mic.wav && rm test-mic.wav
```

arithmetic

Quiz on simple arithmetic problems.

More information: https://manned.org/arithmetic.

- Start an arithmetic quiz:

  ```
  arithmetic
  ```

- Specify one or more arithmetic [o]peration symbols to get problems on them:

  ```
  arithmetic -o {{+|-|x|/}}
  ```

- Specify a [r]ange. Addition and multiplication problems would feature numbers between 0 and range, inclusive. Subtraction and division problems would have required result and number to be operated on, between 0 and range:

  ```
  arithmetic -r {{7}}
  ```

ARK: Survival Ascended

Create and start a headless ARK: Survival Ascended server.

More information: https://ark.wiki.gg/wiki/Server_configuration.

- Start the server with a specific map:

```
{{path/to/ArkAscendedServer}} {{TheIsland_WP}}
```

- Start the server with a specific session name, server password, and admin password:

```
{{path/to/ArkAscendedServer}} {{TheIsland_WP}}?
SessionName={{session_name}}?
ServerPassword={{server_password}}?
ServerAdminPassword={{admin_password}}
```

- Start the server with a specific port and set a maximum player count:

```
{{path/to/ArkAscendedServer}} {{TheIsland_WP}} -port={{7777}}
-WinLiveMaxPlayers={{1..70}}
```

- Enable PvE and disabling PvP:

```
{{path/to/ArkAscendedServer}} {{TheIsland_WP}}?ServerPVE=true
```

- Set a multiplier to scale the server difficulty, affecting the maximum level of wild creatures:

```
{{path/to/ArkAscendedServer}} {{TheIsland_WP}}?
DifficultyOffset={{1.0}}
```

- Disable creature animation optimization to prevent collision issues:

```
{{path/to/ArkAscendedServer}} {{TheIsland_WP}} -
AlwaysTickDedicatedSkeletalMeshes
```

- Enable specific mods by their ID (comma-separated):

```
{{path/to/ArkAscendedServer}} {{TheIsland_WP}} -
mods={{mod_id1, mod_id2, ...}}
```

- Allow connections from specific platforms:

```
{{path/to/ArkAscendedServer}} {{TheIsland_WP}} -
ServerPlatform={{PC+XSX+PS5}}
```

ARK: Survival Evolved

Create and start a headless ARK: Survival Evolved server.

More information: https://ark.wiki.gg/wiki/Server_configuration.

- Start the server with a specific map:

```
{{path/to/ShooterGameServer}} {{TheIsland}}
```

- Start the server with a specific session name, server password, and admin password:

```
{{path/to/ShooterGameServer}} {{TheIsland}}?
SessionName={{session_name}}?
ServerPassword={{server_password}}?
ServerAdminPassword={{admin_password}}
```

- Start the server with a specific port and set a maximum player count:

```
{{path/to/ShooterGameServer}} {{TheIsland}}?Port={{7777}}?
MaxPlayers={{1..70}}
```

- Enable PvE and disabling PvP:

```
{{path/to/ShooterGameServer}} {{TheIsland}}?ServerPVE=true
```

- Set a multiplier to scale the server difficulty, affecting the maximum level of wild creatures:

```
{{path/to/ShooterGameServer}} {{TheIsland}}?
DifficultyOffset={{1.0}}
```

- Enable a specific event:

```
{{path/to/ShooterGameServer}} {{TheIsland}} -
ActiveEvent={{Summer}}
```

- Enable automatic mod downloading, installation, and updating (Steam only):

```
{{path/to/ShooterGameServer}} {{TheIsland}} -automanagedmods
```

- Enable crossplay between Steam and Epic Games Store:

```
{{path/to/ShooterGameServer}} {{TheIsland}} -crossplay -
PublicIPForEpic={{ip_address}}
```

ark

KDE's archiving tool.

More information: https://docs.kde.org/stable5/en/ark/ark/.

- Extract a specific archive into the current directory:

```
ark {{[-b|--batch]}} {{path/to/archive}}
```

- Extract an archive into a specific directory:

```
ark {{[-b|--batch]}} {{[-o|--destination]}} {{path/to/
directory}} {{path/to/archive}}
```

- Create an archive if it does not exist and add specific files to it:

```
ark {{[-t|--add-to]}} {{path/to/archive}} {{path/to/file1
path/to/file2 ...}}
```

arpaname

Get the corresponding ARPA name for a IP addresses.

More information: https://manned.org/arpaname.

- Translate IP addresses (IPv4 and IPv6) to the corresponding ARPA name:

```
arpaname {{ip_address}}
```

arpspoof

Forge ARP replies to intercept packets.

More information: https://manned.org/arpspoof.

- Poison all hosts to intercept packets on [i]nterface for the host:

```
sudo arpspoof -i {{wlan0}} {{host_ip}}
```

- Poison [t]arget to intercept packets on [i]nterface for the host:

```
sudo arpspoof -i {{wlan0}} -t {{target_ip}} {{host_ip}}
```

- Poison both [t]arget and host to intercept packets on [i]nterface for the host:

```
sudo arpspoof -i {{wlan0}} -r -t {{target_ip}} {{host_ip}}
```

arptables

Manage ARP filtering rules using the **`nftables`** backend.

Part of the **`xtables-nft`** suite for ARP packet filtering.

More information: https://manned.org/arptables.

- List all ARP rules in the filter table:

```
sudo arptables {{[-L|--list]}}
```

- Append a rule to drop ARP packets from a specific IP address:

```
sudo arptables {{[-A|--append]}} INPUT {{[-s|--source-ip]}} {{192.168.0.1}} {{[-j|--jump]}} DROP
```

- Delete a specific rule from the INPUT chain by its rule number:

```
sudo arptables {{[-D|--delete]}} INPUT {{rule_number}}
```

- Flush all rules in the filter table:

```
sudo arptables {{[-F|--flush]}}
```

- Set the default policy of the OUTPUT chain to ACCEPT:

```
sudo arptables {{[-P|--policy]}} OUTPUT ACCEPT
```

- Save the current ARP rules to a file:

```
sudo arptables-save > {{path/to/file}}
```

as

Portable GNU assembler.

Primarily intended to assemble output from **`gcc`** to be used by **`ld`**.

More information: https://manned.org/as.

- Assemble a file, writing the output to `a.out`:

```
as {{path/to/file.s}}
```

- Assemble the output to a given file:

```
as {{path/to/file.s}} -o {{path/to/output_file.o}}
```

- Generate output faster by skipping whitespace and comment preprocessing. (Should only be used for trusted compilers):

```
as -f {{path/to/file.s}}
```

- Include a given path to the list of directories to search for files specified in `.include` directives:

```
as -I {{path/to/directory}} {{path/to/file.s}}
```

ascii

Show ASCII character aliases.

More information: http://www.catb.org/~esr/ascii/ascii.html.

- Show ASCII aliases of a character:

```
ascii {{a}}
```

- Show ASCII aliases in short, script-friendly mode:

```
ascii -t {{a}}
```

- Show ASCII aliases of multiple characters:

```
ascii -s {{tldr}}
```

- Show ASCII table in decimal:

```
ascii -d
```

- Show ASCII table in hexadecimal:

```
ascii -x
```

- Show ASCII table in octal:

```
ascii -o
```

- Show ASCII table in binary:

```
ascii -b
```

- Show options summary and complete ASCII table:

```
ascii
```

asciiart

Convert images to ASCII.

More information: https://github.com/nodanaonlyzuul/asciiart.

- Read an image from a file and print in ASCII:

```
asciiart {{path/to/image.jpg}}
```

- Read an image from a URL and print in ASCII:

```
asciiart {{www.example.com/image.jpg}}
```

- Choose the output width (default is 100):

```
asciiart {{[-w|--width]}} {{50}} {{path/to/image.jpg}}
```

- Colorize the ASCII output:

```
asciiart {{[-c|--color]}} {{path/to/image.jpg}}
```

- Choose the output format (default format is text):

```
asciiart {{[-f|--format]}} {{text|html}} {{path/to/
image.jpg}}
```

- Invert the character map:

```
asciiart {{[-i|--invert-chars]}} {{path/to/image.jpg}}
```

asterisk

Run and manage telephone and exchange (phone) server instances.

More information: https://docs.asterisk.org/Operation/.

- [r]econnect to a running server, and turn on logging 3 levels of [v]erbosity:

```
asterisk -r -vvv
```

- [r]econnect to a running server, run a single command, and return:

```
asterisk -r -x "{{command}}"
```

- Show chan_SIP clients (phones):

```
asterisk -r -x "sip show peers"
```

- Show active calls and channels:

```
asterisk -r -x "core show channels"
```

- Show voicemail mailboxes:

```
asterisk -r -x "voicemail show users"
```

- Terminate a channel:

```
asterisk -r -x "hangup request {{channel_ID}}"
```

- Reload chan_SIP configuration:

```
asterisk -r -x "sip reload"
```

at

Execute commands once at a later time.

Results will be sent to the users mail.

More information: https://manned.org/at.

- Start the `atd` daemon:

```
systemctl start atd
```

- Create commands interactively and execute them in 5 minutes (press `<Ctrl d>` when done):

```
at now + 5 minutes
```

- Create commands interactively and execute them at a specific time:

```
at {{hh:mm}}
```

- Execute a command from `stdin` at 10:00 AM today:

```
echo "{{command}}" | at 1000
```

- Execute commands from a given file next Tuesday:

```
at -f {{path/to/file}} 9:30 PM Tue
```

- List all queued jobs for the current user (same as `atq`):

```
at -l
```

- View a specied job:

```
at -c {{job_number}}
```

atop

Linux system and process monitor.

See also: **`htop`**, **`top`**, **`btop`**, **`btm`**, **`glances`**.

More information: https://manned.org/atop.

- Start:

```
atop
```

- Start and display memory consumption for each process:

```
atop -m
```

- Start and display disk information:

```
atop -d
```

- Start and display background process information:

```
atop -c
```

- Start and display thread-specific resource utilization information:

```
atop -y
```

- Start and display the number of processes for each user:

```
atop -au
```

- Display help about interactive commands:

```
<?>
```

audit2allow

Generate SELinux policy allow rules from audit logs.

Part of the **`policycoreutils-python-utils`** package.

See also: **`audit2why`**, **`ausearch`**, **`semodule`**.

More information: https://manned.org/audit2allow.

- Generate allow rules from recent audit denials and display them:

```
sudo audit2allow {{[-a|--all]}}
```

- Generate allow rules from a specific audit log file:

```
sudo audit2allow {{[-i|--input]}} {{path/to/audit.log}}
```

- Generate a policy module from recent audit denials:

```
sudo audit2allow {{[-a|--all]}} {{[-M|--module]}}
{{module_name}}
```

- Explain why SELinux denials occurred (same as `audit2why`):

```
sudo audit2allow {{[-a|--all]}} --why
```

- Display detailed information around generated messages:

```
sudo audit2allow {{[-a|--all]}} {{[-e|--explain]}}
```

- Use installed macros to generate a reference policy:

```
sudo audit2allow {{[-a|--all]}} {{[-R|--reference]}}
```

- Generate allow rules for a specific service:

```
sudo ausearch {{[-m|--message]}} avc {{[-c|--comm]}}
{{service_name}} | audit2allow {{[-M|--module]}}
{{policy_name}}
```

- Enable verbose output mode:

```
sudo audit2allow {{[-a|--all]}} {{[-v|--verbose]}}
```

audit2why

Explain SELinux denials from audit logs.

Part of the **policycoreutils-python-utils** package.

See also: **audit2allow**, **ausearch**, **sealert**.

More information: https://manned.org/audit2why.

- Explain the most recent SELinux denial:

```
sudo audit2why
```

- Explain SELinux denials from a specific audit log file:

```
sudo audit2why {{[-i|--input]}} {{path/to/audit.log}}
```

- Explain all SELinux denials from the audit log:

```
sudo ausearch {{[-m|--message]}} avc | audit2why
```

- Explain denials for a specific service:

```
sudo ausearch {{[-m|--message]}} avc {{[-c|--comm]}}
{{service_name}} | audit2why
```

auditctl

Utility to control the behavior, get status and manage rules of the Linux Auditing System.

More information: https://manned.org/auditctl.

- Display the [s]tatus of the audit system:

```
sudo auditctl -s
```

- [l]ist all currently loaded audit rules:

```
sudo auditctl -l
```

- [D]elete all audit rules:

```
sudo auditctl -D
```

- [e]nable/disable the audit system:

```
sudo auditctl -e {{1|0}}
```

- Watch a file for changes:

```
sudo auditctl -a always,exit -F arch=b64 -F path=/{{path/to/file}} -F perm=wa
```

- Recursively watch a directory for changes:

```
sudo auditctl -a always,exit -F arch=b64 -F dir=/{{path/to/directory}}/ -F perm=wa
```

- Display [h]elp:

```
auditctl -h
```

aur sync

Download and build AUR packages automatically.

Note: A local repository needs to be defined in **`/etc/pacman.conf`** and **`vifm`** needs to be installed for this to fully function.

More information: https://github.com/aurutils/aurutils.

- Download one or more packages and their dependencies from the AUR, build them, and add them to a local repository:

```
aur sync {{package1 package2 ...}}
```

- Upgrade local repository packages:

```
aur sync {{[-u|--upgrades]}}
```

- Clean build files after install:

```
aur sync {{[-C|--clean]}} {{package}}
```

- Install a package without viewing changes in Vim and do not confirm dependency installation:

```
aur sync --noview {{[-n|--noconfirm]}} {{package}}
```

- Ignore specific packages when upgrading:

```
aur sync {{[-u|--upgrades]}} --ignore
{{package1,package2,...}}
```

aur

Build packages from the AUR and manage local repositories.

Note: A local repository needs to be defined in **`/etc/pacman.conf`** and **`vifm`** needs to be installed for this to fully function.

More information: https://github.com/aurutils/aurutils.

- Initialize the repository that matches the path in `/etc/pacman.conf`:

```
repo-add {{path/to/database.db.tar.gz}}
```

- Search the AUR database for a package:

```
aur search {{keyword}}
```

- Download one or more packages and their dependencies from the AUR, build them, and add them to a local repository:

```
aur sync {{package1 package2 ...}}
```

- List packages available in your local repository:

```
aur repo {{[-l|--list]}}
```

- Upgrade local repository packages:

```
aur sync {{[-u|--upgrades]}}
```

- Clean build files after install:

```
aur sync {{[-C|--clean]}} {{package}}
```

- Install a package without viewing changes in Vim and do not confirm dependency installation:

```
aur sync --noview {{[-n|--noconfirm]}} {{package}}
```

- Remove a package form the repository metadata (does not remove the package file itself):

```
repo-remove {{path/to/database.db.tar.gz}} {{package}}
```

aura

The Aura Package Manager: a secure, multilingual package manager for Arch Linux and the AUR.

More information: https://github.com/fosskers/aura.

- Search for packages from the official repositories and AUR:

```
aura --aursync --both --search {{keyword|regex}}
```

- Install a package from the AUR:

```
aura --aursync {{package}}
```

- Update all AUR packages in a verbose mode and remove all make dependencies:

```
aura --aursync --diff --sysupgrade --delmakedeps --unsuppress
```

- Install a package from the official repositories:

```
aura --sync {{package}}
```

- Synchronize and update all packages from the official repositories:

```
aura --sync --refresh --sysupgrade
```

- Downgrade a package using the package cache:

```
aura --downgrade {{package}}
```

- Remove a package and its dependencies:

```
aura --remove --recursive --unneeded {{package}}
```

- Remove orphan packages (installed as dependencies but not required by any package):

```
aura --orphans --abandon
```

auracle

Interact with Arch Linux's User Repository, commonly referred to as the AUR.

More information: https://github.com/falconindy/auracle.

- Display AUR packages that match a `regex`:

  ```
  auracle search '{{regex}}'
  ```

- Display information about one or more AUR packages:

  ```
  auracle info {{package1 package2 ...}}
  ```

- Display the `PKGBUILD` file (build information) of one or more AUR packages:

  ```
  auracle show {{package1 package2 ...}}
  ```

- Display updates for installed AUR packages:

  ```
  auracle outdated
  ```

aurman

An Arch Linux utility to build and install packages from the Arch User Repository.

See also: **pacman**.

More information: https://github.com/polygamma/aurman.

- Synchronize and update all packages:

```
aurman {{[-S|--sync]}} {{[-y|--refresh]}} {{[-u|--sysupgrade]}}
```

- Synchronize and update all packages without show changes of `PKGBUILD` files:

```
aurman {{[-S|--sync]}} {{[-y|--refresh]}} {{[-u|--sysupgrade]}} --noedit
```

- Install a new package:

```
aurman {{[-S|--sync]}} {{package}}
```

- Install a new package without show changes of `PKGBUILD` files:

```
aurman {{[-S|--sync]}} --noedit {{package}}
```

- Install a new package without prompting:

```
aurman {{[-S|--sync]}} --noedit --noconfirm {{package}}
```

- Search the package database for a keyword from the official repositories and AUR:

```
aurman {{[-S|--sync]}} {{[-s|--search]}} {{keyword}}
```

- Remove a package and its dependencies:

```
aurman --remove --recursive --nosave {{package}}
```

- Clear the package cache (use two `--clean` flags to clean all packages):

```
aurman {{[-S|--sync]}} {{[-c|--clean]}}
```

aurpublish

Publish Arch User Repository packages.

More information: https://github.com/eli-schwartz/aurpublish.

- Verify `PKGBUILD` integrity, generate `.SRCINFO`, create a commit message template, and publish the package to the AUR:

```
aurpublish {{package_name}}
```

- Add githooks to the current repository:

```
aurpublish setup
```

- Display help:

```
aurpublish {{[-h|--help]}}
```

aurvote

Vote for packages in the Arch User Repository.

To be able to vote, the file **`~/.config/aurvote`** must exist and contain your AUR credentials.

More information: https://github.com/archlinuxfr/aurvote.

- Interactively create the file `~/.config/aurvote` containing your AUR username and password:

```
aurvote --configure
```

- Vote for one or more AUR packages:

```
aurvote {{package1 package2 ...}}
```

- Unvote one or more AUR packages:

```
aurvote {{[-u|--unvote]}} {{package1 package2 ...}}
```

- Check if one or more AUR packages have already been voted:

```
aurvote {{[-c|--check]}} {{package1 package2 ...}}
```

- Display help:

```
aurvote {{[-h|--help]}}
```

ausearch

Query the Linux audit log for events.

Part of the **`audit`** package.

See also: **`audit2why`**, **`audit2allow`**, **`aureport`**.

More information: https://manned.org/ausearch.

- Search for all SELinux AVC denial events:

```
sudo ausearch {{[-m|--message]}} avc
```

- Search for events related to a specific executable:

```
sudo ausearch {{[-c|--comm]}} {{httpd}}
```

- Search for events from a specific user:

```
sudo ausearch {{[-ui|--uid]}} {{1000}}
```

- Search for events in the last 10 minutes:

```
sudo ausearch {{[-ts|--start]}} recent
```

- Search for failed login attempts:

```
sudo ausearch {{[-m|--message]}} user_login {{[-sv|--
success]}} no
```

- Search for events related to a specific file:

```
sudo ausearch {{[-f|--file]}} {{path/to/file}}
```

- Display results in raw format for further processing:

```
sudo ausearch {{[-m|--message]}} avc --raw
```

ausyscall

Map syscall names and numbers.

More information: https://manned.org/ausyscall.

- Display syscall number of a specific system call:

```
ausyscall {{search_pattern}}
```

- Display name of a specific system call number:

```
ausyscall {{system_call_number}}
```

- Display all system calls for a specific architecture:

```
ausyscall {{architecture}} --dump
```

authconfig

Configure system authentication resources.

More information: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system-level_authentication_guide/authconfig-install.

- Display the current configuration (or dry run):

```
authconfig --test
```

- Configure the server to use a different password hashing algorithm:

```
authconfig --update --passalgo={{algorithm}}
```

- Enable LDAP authentication:

```
authconfig --update --enableldapauth
```

- Disable LDAP authentication:

```
authconfig --update --disableldapauth
```

- Enable Network Information Service (NIS):

```
authconfig --update --enablenis
```

- Enable Kerberos:

```
authconfig --update --enablekrb5
```

- Enable Winbind (Active Directory) authentication:

```
authconfig --update --enablewinbindauth
```

- Enable local authorization:

```
authconfig --update --enablelocauthorize
```

auto-cpufreq

Automatic CPU speed & power optimizer.

More information: https://github.com/AdnanHodzic/auto-cpufreq.

- Run `auto-cpufreq` in a specific mode:

```
sudo auto-cpufreq --{{monitor|live|update|remove|stats|force=governor}}
```

autopkgtest

Run tests on Debian packages.

More information: https://manpages.debian.org/bookworm/autopkgtest/autopkgtest.1.en.html.

- Build the package in the current directory and run all tests directly on the system:

```
autopkgtest -- {{null}}
```

- Run a specific test for the package in the current directory:

```
autopkgtest --test-name={{test_name}} -- {{null}}
```

- Download and build a specific package with `apt-get`, then run all tests:

```
autopkgtest {{package}} -- {{null}}
```

- Test the package in the current directory using a new root directory:

```
autopkgtest -- {{chroot}} {{path/to/new_root}}
```

- Test the package in the current directory without rebuilding it:

```
autopkgtest {{[-B|--no-built-binaries]}} -- {{null}}
```

autorandr

Automatically change screen layout.

More information: https://manned.org/autorandr.

- Save the current screen layout:

```
autorandr {{[-s|--save]}} {{profile_name}}
```

- Show the saved profiles:

```
autorandr
```

- Load the first detected profile:

```
autorandr {{[-c|--change]}}
```

- Load a specific profile:

```
autorandr {{[-l|--load]}} {{profile_name}}
```

- Set the default profile:

```
autorandr {{[-d|--default]}} {{profile_name}}
```

autorecon

A multi-threaded network reconnaissance tool which performs automated enumeration of services.

More information: https://github.com/Tib3rius/AutoRecon.

- Perform reconnaissance on target host(s) (detailed scan results will be dumped in `./results`):

```
sudo autorecon {{host_or_ip1,host_or_ip2,...}}
```

- Perform reconnaissance on target(s) from a file:

```
sudo autorecon {{[-t|--target-file]}} {{path/to/file}}
```

- Output results to a different directory:

```
sudo autorecon {{[-o|--output]}} {{path/to/results}}
{{host_or_ip1,host_or_ip2,...}}
```

- Limit scanning to specific ports and protocols (T for TCP, U for UDP, B for both):

```
sudo autorecon {{[-p|--ports]}} {{T:21-25,80,443,U:53,B:123}}
{{host_or_ip1,host_or_ip2,...}}
```

avahi-browse

Display services and hosts exposed on the local network via mDNS/DNS-SD.

Avahi is compatible with Bonjour (Zeroconf) found in Apple devices.

More information: https://manned.org/avahi-browse.

- List services available on the local network along with their addresses and ports, ignoring ones on the local machine:

```
avahi-browse {{[-a|--all]}} {{[-r|--resolve]}} {{[-l|--ignore-local]}}
```

- Quickly list services in the local network in SSV format for scripts:

```
avahi-browse {{[-a|--all]}} {{[-t|--terminate]}} {{[-p|--parsable]}}
```

- List domains in the neighbourhood:

```
avahi-browse {{[-D|--browse-domains]}}
```

- Limit the search to a particular domain:

```
avahi-browse {{[-a|--all]}} --domain={{domain}}
```

avahi-resolve-address

This command is an alias of **`avahi-resolve --address`**.

- View documentation for the original command:

  ```
  tldr avahi-resolve
  ```

avahi-resolve-host-name

This command is an alias of **`avahi-resolve --name`**.

- View documentation for the original command:

  ```
  tldr avahi-resolve
  ```

avahi-resolve

Translate between host names and IP Addresses.

More information: https://manned.org/avahi-resolve.

- Resolve a local service to its IPv4:

```
avahi-resolve -4 {{[-n|--name]}} {{service.local}}
```

- Resolve an IP to a hostname, verbosely:

```
avahi-resolve {{[-v|--verbose]}} {{[-a|--address]}} {{IP}}
```

avifenc

AV1 Image File Format (AVIF) encoder.

More information: https://aomediacodec.github.io/av1-avif/.

- Convert a specific PNG image to AVIF:

```
avifenc {{path/to/input.png}} {{path/to/output.avif}}
```

- Encode with a specific speed (6=default, 0=slowest and 10=fastest):

```
avifenc --speed {{2}} {{path/to/input.png}} {{path/to/
output.avif}}
```

away

Locks terminal with an away message.

More information: https://manned.org/away.

- Lock terminal and set away message:

```
away {{message}}
```

- Lock terminal and enable mail check:

```
away {{[-c|--mail]}} {{message}}
```

- Lock terminal and disable mail check:

```
away {{[-C|--nomail]}} {{message}}
```

- Lock terminal and sleep background tasks for number of seconds:

```
away {{[-t|--time]}} {{seconds}} {{message}}
```

- Lock terminal and check mail if at least one inbox hasn't received new mail:

```
away {{[-p|--persist]}} {{message}}
```

- Lock terminal and check mail until at least one inbox has received new mail:

```
away {{[-P|--nopersist]}} {{message}}
```

- Display help:

```
away {{[-h|--help]}}
```

- Display version:

```
away {{[-v|--version]}}
```

backlight_control

Control a linux machine's backlight using percentage values.

More information: https://github.com/Hendrikto/backlight_control.

- Increase/decrease the backlight by a specific percent count:

```
backlight_control {{+|-}}{{5}}
```

- Set the backlight strength to a specific percent count:

```
backlight_control {{90}}
```

- Display help:

```
backlight_control
```

balooctl

File indexing and searching framework for KDE Plasma.

More information: https://wiki.archlinux.org/index.php/Baloo.

- Display the status of the indexer:

```
balooctl status
```

- Enable/Disable the file indexer:

```
balooctl {{enable|disable}}
```

- Clean the index database:

```
balooctl purge
```

- Suspend the file indexer:

```
balooctl suspend
```

- Resume the file indexer:

```
balooctl resume
```

- Display the disk space used by Baloo:

```
balooctl indexSize
```

- Check for any unindexed files and index them:

```
balooctl check
```

- Display help:

```
balooctl {{[-h|--help]}}
```

batcat

This command is an alias of **bat**.

- View documentation for the original command:

  ```
  tldr bat
  ```

batch

Execute commands at a later time when the system load levels permit.

Results will be sent to the user's mail.

See also: **`at`**, **`atq`**, **`atrm`** **`mail`**.

More information: https://manned.org/batch.

- Start the `atd` daemon:

  `systemctl start atd`

- Execute commands from `stdin` (press `<Ctrl d>` when done):

  `batch`

- Execute a command from `stdin`:

  `echo "{{./make_db_backup.sh}}" | batch`

battop

An interactive viewer for the batteries installed in your notebook.

More information: https://github.com/svartalf/rust-battop.

- Display battery information:

```
battop
```

- Change battery information measurement [u]nit (default: human):

```
battop -u {{human|si}}
```

bcachefs device

Manage devices within a running **`bcachefs`** filesystem.

More information: https://bcachefs-docs.readthedocs.io/en/latest/mgmt-devicemanagement.html.

- Format and add a new device to an existing filesystem.:

```
sudo bcachefs device add --label {{group}}.{{name}} {{path/to/mountpoint}} {{path/to/device}}
```

- Migrate data off a device to prepare for removal:

```
bcachefs device evacuate {{path/to/device}}
```

- Permanently remove a device from a filesystem:

```
bcachefs device remove {{path/to/device}}
```

bcachefs

Manage **`bcachefs`** filesystems/devices.

Some subcommands such as **`device`** have their own usage documentation.

More information: https://bcachefs-docs.readthedocs.io/en/latest/index.html.

- Format a partition with `bcachefs`:

```
sudo bcachefs format {{path/to/partition}}
```

- Mount a `bcachefs` filesystem:

```
sudo bcachefs mount {{path/to/partition}} {{path/to/
mountpoint}}
```

- Create a RAID 0 filesystem where an SSD acts as a cache and an HDD acts as a long-term storage:

```
sudo bcachefs format {{[-l|--label]}} {{ssd.ssd1}} {{path/to/
ssd_partition}} {{[-l|--label]}} {{hdd.hdd1}} {{path/to/
hdd_partition}} --replicas 1 --foreground_target {{ssd}} --
promote_target {{ssd}} --background_target {{hdd}}
```

- Mount a multidevice filesystem:

```
sudo bcachefs mount {{path/to/partition1}}:{{path/to/
partition2}} {{path/to/mountpoint}}
```

- Display disk usage:

```
bcachefs fs usage {{[-h|--human-readable]}} {{path/to/
mountpoint}}
```

- Set replicas after formatting and mounting:

```
sudo bcachefs set-fs-option --metadata_replicas {{2}} --
data_replicas {{2}} {{path/to/partition}}
```

- Force `bcachefs` to ensure all files are replicated:

```
sudo bcachefs data rereplicate {{path/to/mountpoint}}
```

- Create a snapshot of a particular directory:

```
bcachefs subvolume snapshot {{path/to/directory}} {{path/to/
snapshot}}
```

bchunk

Convert CD images to a set of **`.iso`** and **`.cdr`** tracks.

More information: https://manned.org/bchunk.

- Convert binary CD into a standard iso9960 image file:

```
bchunk {{path/to/image.bin}} {{path/to/image.cue}} {{path/to/output}}
```

- Convert with verbose mode:

```
bchunk -v {{path/to/image.bin}} {{path/to/image.cue}} {{path/to/output}}
```

- Output audio files in WAV format:

```
bchunk -w {{path/to/image.bin}} {{path/to/image.cue}} {{path/to/output}}
```

beep

A utility to beep the PC speaker.

More information: https://github.com/spkr-beep/beep.

- Play a beep:

```
beep
```

- Play a beep that repeats:

```
beep -r {{repetitions}}
```

- Play a beep at a specified frequency (Hz) and duration (milliseconds):

```
beep -f {{frequency}} -l {{duration}}
```

- Play each new frequency and duration as a distinct beep:

```
beep -f {{frequency}} -l {{duration}} {{[-n|--new]}} -f
{{frequency}} -l {{duration}}
```

- Play the C major scale:

```
beep -f {{262}} {{[-n|--new]}} -f {{294}} {{[-n|--new]}} -f
{{330}} {{[-n|--new]}} -f {{349}} {{[-n|--new]}} -f {{392}}
{{[-n|--new]}} -f {{440}} {{[-n|--new]}} -f {{494}} {{[-n|--
new]}} -f {{523}}
```

betterdiscordctl

Manage BetterDiscord on Linux.

More information: https://github.com/bb010g/betterdiscordctl#manual.

- Install BetterDiscord on Discord Stable:

```
sudo betterdiscordctl install
```

- Install BetterDiscord on Discord Canary:

```
sudo betterdiscordctl {{[-f|--d-flavors]}} canary install
```

- Install BetterDiscord on Discord PTB:

```
sudo betterdiscordctl {{[-f|--d-flavors]}} ptb install
```

- Install BetterDiscord on Discord installed with Flatpak:

```
sudo betterdiscordctl {{[-i|--d-install]}} flatpak install
```

- Install BetterDiscord on Discord installed with Snap:

```
sudo betterdiscordctl {{[-i|--d-install]}} snap install
```

betterlockscreen

Simple, minimal lock screen.

More information: https://github.com/betterlockscreen/betterlockscreen#usage.

- Lock the screen:

```
betterlockscreen {{[-l|--lock]}}
```

- Change the lock screen background:

```
betterlockscreen {{[-u|--update]}} {{path/to/image.png}}
```

- Lock the screen, showing some custom text:

```
betterlockscreen {{[-l|--lock]}} pixel --text "{{custom lock screen text}}"
```

- Lock the screen, with a custom monitor off timeout in seconds:

```
betterlockscreen --off {{5}} {{[-l|--lock]}}
```

bindfs

Mount a directory elsewhere with different permissions.

More information: https://bindfs.org/docs/bindfs.1.html.

- Mount a directory with same permissions:

```
sudo bindfs {{path/to/directory}} {{path/to/mount_point}}
```

- Map filesystem objects owned by `user1` to be owned by `user2` (also applies in reverse to newly created files):

```
sudo bindfs --map={{user1}}/{{user2}} {{path/to/directory}} {{path/to/mount_point}}
```

- Unmount a directory:

```
sudo umount {{path/to/mount_point}}
```

bitwise

Multi base interactive calculator supporting dynamic base conversion and bit manipulation.

More information: https://github.com/mellowcandle/bitwise.

- Run using interactive mode:

```
bitwise
```

- Convert from decimal:

```
bitwise {{12345}}
```

- Convert from hexadecimal:

```
bitwise {{0x563d}}
```

- Convert a C-style calculation:

```
bitwise "{{0x123 + 0x20 - 30 / 50}}"
```

blastn

Nucleotide-Nucleotide BLAST.

More information: https://www.ncbi.nlm.nih.gov/books/NBK279684/table/appendices.T.blastn_application_options/.

- Align two or more sequences using megablast (default), with the e-value threshold of 1e-9, pairwise output format (default):

```
blastn -query {{query.fa}} -subject {{subject.fa}} -evalue {{1e-9}}
```

- Align two or more sequences using blastn:

```
blastn -task blastn -query {{query.fa}} -subject {{subject.fa}}
```

- Align two or more sequences, custom tabular output format, output to file:

```
blastn -query {{query.fa}} -subject {{subject.fa}} -outfmt '{{6 qseqid qlen qstart qend sseqid slen sstart send bitscore evalue pident}}' -out {{output.tsv}}
```

- Search nucleotide databases using a nucleotide query, 16 threads (CPUs) to use in the BLAST search, with a maximum number of 10 aligned sequences to keep:

```
blastn -query {{query.fa}} -db {{path/to/blast_db}} -num_threads {{16}} -max_target_seqs {{10}}
```

- Search the remote non-redundant nucleotide database using a nucleotide query:

```
blastn -query {{query.fa}} -db {{nt}} -remote
```

- Display help (use `-help` for detailed help):

```
blastn -h
```

blastp

Protein-Protein BLAST.

More information: https://www.ncbi.nlm.nih.gov/books/NBK279684/table/appendices.T.blastp_application_options/.

- Align two or more sequences using blastp, with the e-value threshold of 1e-9, pairwise output format, output to screen:

```
blastp -query {{query.fa}} -subject {{subject.fa}} -evalue {{1e-9}}
```

- Align two or more sequences using blastp-fast:

```
blastp -task blastp-fast -query {{query.fa}} -subject {{subject.fa}}
```

- Align two or more sequences, custom tabular output format, output to file:

```
blastp -query {{query.fa}} -subject {{subject.fa}} -outfmt '{{6 qseqid qlen qstart qend sseqid slen sstart send bitscore evalue pident}}' -out {{output.tsv}}
```

- Search protein databases using a protein query, 16 threads to use in the BLAST search, with a maximum number of 10 aligned sequences to keep:

```
blastp -query {{query.fa}} -db {{blast_database_name}} -num_threads {{16}} -max_target_seqs {{10}}
```

- Search the remote non-redundant protein database using a protein query:

```
blastp -query {{query.fa}} -db {{nr}} -remote
```

- Display help (use `-help` for detailed help):

```
blastp -h
```

bleachbit

Clean junk files on the filesystem.

More information: https://docs.bleachbit.org/doc/command-line-interface.html.

- Start the graphical user interface (GUI) version of Bleachbit:

```
bleachbit --gui
```

- Shred a file:

```
bleachbit {{[-s|--shred]}} {{path/to/file}}
```

- List available cleaner options:

```
bleachbit {{[-l|--list-cleaners]}}
```

- Preview the files that will be deleted and other changes that will be made before actually performing the clean-up operation:

```
bleachbit {{[-p|--preview]}} --preset {{cleaner1.option1
cleaner2.option2 ...}}
```

- Perform the clean-up operation and delete files:

```
bleachbit {{[-c|--clean]}} --preset {{cleaner1.option1
cleaner2.option2 ...}}
```

blight

Utility for changing the display brightness.

More information: https://github.com/gutjuri/blight.

- Set display brightness to 50%:

```
blight set {{50}} {{[-r|--relative]}}
```

- Show current display brightness:

```
blight show
```

- Print maximum display brightness:

```
blight max
```

- Increase display brightness in %:

```
blight inc {{number}} {{[-r|--relative]}}
```

- Decrease display brightness with internal units:

```
blight dec {{number}}
```

blkdiscard

Discards device sectors on storage devices. Useful for SSDs.

More information: https://manned.org/blkdiscard.

- Discard all sectors on a device, removing all data:

```
blkdiscard {{/dev/device}}
```

- Securely discard all blocks on a device, removing all data:

```
blkdiscard {{[-s|--secure]}} {{/dev/device}}
```

- Discard the first 100 MB of a device:

```
blkdiscard {{[-l|--length]}} {{100MB}} {{/dev/device}}
```

blkid

List all recognized partitions and their Universally Unique Identifier (UUID).

More information: https://manned.org/blkid.

- List all partitions:

```
sudo blkid
```

- List all partitions in a table, including current mountpoints:

```
sudo blkid {{[-o|--output]}} list
```

- Get the UUID of the filesystem on a partition:

```
sudo blkid {{[-s|--match-tag]}} UUID {{[-o|--output]}} value
{{/dev/sdXY}}
```

blkpr

Register, reserve, release, preempt, and clear persistent reservations on a block device that supports Persistent Reservations.

More information: https://manned.org/blkpr.

- Register (command) a new reservation with a given key on a given device:

```
blkpr {{[-c|--command]}} register {{[-k|--key]}} {{reservation_key}} {{path/to/device}}
```

- Set the type of an existing reservation to exclusive access:

```
blkpr {{[-c|--command]}} reserve {{[-k|--key]}} {{reservation_key}} {{[-t|--type]}} exclusive-access {{path/to/device}}
```

- Preempt the existing reservation with a given key and replace it with a new reservation:

```
blkpr {{[-c|--command]}} preempt {{[-K|--oldkey]}} {{old_key}} {{[-k|--key]}} {{new_key}} {{[-t|--type]}} write-exclusive {{path/to/device}}
```

- Release a reservation with a given key and type on a given device:

```
blkpr {{[-c|--command]}} release {{[-k|--key]}} {{reservation_key}} {{[-t|--type]}} {{reservation_type}} {{path/to/device}}
```

- Clear all reservations from a given device:

```
blkpr {{[-c|--command]}} clear {{[-k|--key]}} {{key}} {{path/to/device}}
```

blockdev

Manage, query, and manipulate block devices.

More information: https://manned.org/blockdev.

- Print a report for all devices:

```
sudo blockdev --report
```

- Print a report for a specific device:

```
sudo blockdev --report {{/dev/sdXY}}
```

- Get the size of a device in 512-byte sectors:

```
sudo blockdev --getsz {{/dev/sdXY}}
```

- Set read-only:

```
sudo blockdev --setro {{/dev/sdXY}}
```

- Set read-write:

```
sudo blockdev --setrw {{/dev/sdXY}}
```

- Flush buffers:

```
sudo blockdev --flushbufs {{/dev/sdXY}}
```

- Get the physical block size:

```
sudo blockdev --getpbsz {{/dev/sdXY}}
```

- Set the read-ahead value to 128 sectors:

```
sudo blockdev --setra 128 {{/dev/sdXY}}
```

bluebuild

Build Containerfiles and custom images based on your **`recipe.yml`**.

More information: https://github.com/blue-build/cli.

- Build a recipe:

```
bluebuild build {{path/to/recipe.yml}}
```

- Validate a recipe:

```
bluebuild validate {{path/to/recipe.yml}}
```

- Generate a Containerfile:

```
bluebuild generate {{[-o|--output]}} {{Containerfile}}
{{path/to/recipe.yml}}
```

- Generate an ISO from a recipe:

```
bluebuild generate-iso --output-dir {{path/to/
output_directory}} --iso-name {{iso_name.iso}} recipe {{path/
to/recipe.yml}}
```

- Display help:

```
bluebuild {{[-h|--help]}}
```

bluetoothctl

Manage Bluetooth devices.

See also: **`bluetui`**.

More information: https://manned.org/bluetoothctl.

- Enter the `bluetoothctl` shell:

```
bluetoothctl
```

- List all known devices:

```
bluetoothctl devices
```

- Power the Bluetooth controller on or off:

```
bluetoothctl power {{on|off}}
```

- Pair with a device:

```
bluetoothctl pair {{mac_address}}
```

- Remove a device:

```
bluetoothctl remove {{mac_address}}
```

- Connect to a paired device:

```
bluetoothctl connect {{mac_address}}
```

- Disconnect from a paired device:

```
bluetoothctl disconnect {{mac_address}}
```

- Display help:

```
bluetoothctl help
```

bluetoothd

Daemon to manage bluetooth devices.

More information: https://manned.org/bluetoothd.

- Start the daemon:

```
bluetoothd
```

- Start the daemon, logging to `stdout`:

```
bluetoothd {{[-n|--nodetach]}}
```

- Start the daemon with a specific configuration file (defaults to `/etc/bluetooth/main.conf`):

```
bluetoothd {{[-f|--configfile]}} {{path/to/file}}
```

- Start the daemon with verbose output to `stderr`:

```
bluetoothd {{[-d|--debug]}}
```

- Start the daemon with verbose output coming from specific files in the bluetoothd or plugins source:

```
bluetoothd {{[-d|--debug=]}}{{path/to/file1:path/to/file2:...}}
```

bluetui

Manage Bluetooth devices with a terminal UI.

See also: **`bluetoothctl`**.

More information: https://github.com/pythops/bluetui#-usage.

- Start the program:

```
bluetui
```

- Display help:

```
bluetui {{[-h|--help]}}
```

blurlock

A simple wrapper around the i3 screen locker **`i3lock`**, which blurs the screen.

See also: **`i3lock`**.

More information: https://gitlab.manjaro.org/packages/community/i3/i3exit/-/blob/master/blurlock.

- Lock the screen to a blurred screenshot of the current screen:

```
blurlock
```

- Lock the screen and disable the unlock indicator (removes feedback on keypress):

```
blurlock {{[-u|--no-unlock-indicator]}}
```

- Lock the screen and don't hide the mouse pointer:

```
blurlock {{[-p|--pointer]}} {{default}}
```

- Lock the screen and show the number of failed login attempts:

```
blurlock {{[-f|--show-failed-attempts]}}
```

bmon

Monitor bandwidth and capture network related statistics.

More information: https://github.com/tgraf/bmon.

- Display the list of all the interfaces:

```
bmon {{[-a|--show-all]}}
```

- Display data transfer rates in bits per second:

```
bmon {{[-b|--use-bit]}}
```

- Specify the policy to define which network interface(s) is/are displayed:

```
bmon {{[-p|--policy]}}
{{interface_1,interface_2,interface_3}}
```

- Specify the interval (in seconds) in which rate per counter is calculated:

```
bmon {{[-R|--rate-interval]}} {{2.0}}
```

boltctl

Control thunderbolt devices.

More information: https://manned.org/boltctl.

- List connected (and authorized) devices:

```
boltctl
```

- List connected devices, including unauthorized ones:

```
boltctl list
```

- Authorize a device temporarily:

```
boltctl authorize {{device_uuid}}
```

- Authorize and remember a device:

```
boltctl enroll {{device_uuid}}
```

- Revoke a previously authorized device:

```
boltctl forget {{device_uuid}}
```

- Show more information about a device:

```
boltctl info {{device_uuid}}
```

bootc switch

Target a new container image reference to boot.

More information: https://manned.org/bootc-switch.8.

- Change the base OS to a new container image from a registry:

```
sudo bootc switch {{image}}
```

- Change the base OS to a new container image from the local image storage of the root user:

```
sudo bootc switch --transport containers-storage {{image}}
```

- Change the base OS to a new container image stored in a tarball:

```
sudo bootc switch --transport oci-archive {{path/to/image.tar.gz}}
```

bootc

Boot and upgrade via container images.

Manages transactional, in-place operating system updates using OCI/Docker container images.

More information: https://manned.org/bootc.8.

- Show deployments in the order they will appear in the bootloader:

```
bootc status
```

- Check if any updates are available:

```
bootc upgrade --check
```

- Prepare a new update and reboot into it:

```
bootc upgrade --apply
```

- Change OS base to new container image:

```
bootc switch {{image}}
```

- Reboot into the previous ostree deployment:

```
bootc rollback
```

- Apply transactional changes to the system configuration:

```
bootc edit
```

bootctl

Control EFI firmware boot settings and manage boot loader.

More information: https://www.freedesktop.org/software/systemd/man/bootctl.html.

- Show information about the system firmware and the bootloaders:

```
bootctl
```

- Show all available bootloader entries:

```
bootctl list
```

- Set a flag to boot into the system firmware on the next boot (similar to `sudo systemctl reboot --firmware-setup`):

```
sudo bootctl reboot-to-firmware true
```

- Specify the path to the EFI system partition (defaults to `/efi/`, `/boot/` or `/boot/efi`):

```
bootctl --esp-path /{{path/to/efi_system_partition}}/
```

- Install `systemd-boot` into the EFI system partition:

```
sudo bootctl install
```

- Remove all installed versions of `systemd-boot` from the EFI system partition:

```
sudo bootctl remove
```

bore local

Start a local proxy to a remote server using Bore.

More information: https://github.com/ekzhang/bore#detailed-usage.

- Expose a local port to a remote Bore server:

```
bore local {{[-t|--to]}} {{remote_server_address}} {{local_port}}
```

- Expose a specific local host instead of `localhost`:

```
bore local {{[-l|--local-host]}} {{host}} {{[-t|--to]}} {{remote_server_address}} {{local_port}}
```

- Specify a remote server port explicitly:

```
bore local {{[-t|--to]}} {{remote_server_address}} {{[-p|--port]}} {{remote_port}} {{local_port}}
```

- Use a secret for authentication:

```
bore local {{[-t|--to]}} {{remote_server_address}} {{[-s|--secret]}} {{your_secret}} {{local_port}}
```

- Display help:

```
bore local {{[-h|--help]}}
```

bore

Start a local proxy to a remote server using Bore.

Some subcommands such as **`local`** have their own usage documentation.

More information: https://github.com/ekzhang/bore#detailed-usage.

- Expose a local port to a remote Bore server:

```
bore local {{[-t|--to]}} {{remote_server_address}} {{local_port}}
```

- Start a Bore server:

```
bore server
```

- Display help:

```
bore {{[-h|--help]}}
```

bpftool

Inspect and manipulate eBPF programs and maps in a simple way.

Some subcommands such as **`prog`** have their own usage documentation.

More information: https://manned.org/bpftool.

- List information about loaded eBPF programs:

```
bpftool prog list
```

- List eBPF program attachments in the kernel networking subsystem:

```
bpftool net list
```

- List all active links:

```
bpftool link list
```

- List all `raw_tracepoint`, `tracepoint`, `kprobe` attachments in the system:

```
bpftool perf list
```

- List `BPF Type Format (BTF)` data:

```
bpftool btf list
```

- List information about loaded maps:

```
bpftool map list
```

- Probe a network device "eth0" for supported eBPF features:

```
bpftool feature probe dev {{eth0}}
```

- Run commands in batch mode from a file:

```
bpftool batch file {{myfile}}
```

bpftrace

High-level tracing language for Linux eBPF.

More information: https://github.com/bpftrace/bpftrace/blob/master/man/adoc/bpftrace.adoc.

- List all available probes:

```
sudo bpftrace -l
```

- Run a one-liner program (e.g. syscall count by program):

```
sudo bpftrace -e '{{tracepoint:raw_syscalls:sys_enter
{ @[comm] = count(); }}}'
```

- Run a program from a file:

```
sudo bpftrace {{path/to/file}}
```

- Trace a program by PID:

```
sudo bpftrace -e '{{tracepoint:raw_syscalls:sys_enter /pid ==
123/ { @[comm] = count(); }}}'
```

- Do a dry run and display the output in eBPF format:

```
sudo bpftrace -d -e '{{one_line_program}}'
```

- Display version:

```
bpftrace {{[-V|--version]}}
```

br

Navigate directory trees interactively.

See also: **`broot`**.

More information: https://github.com/Canop/broot.

- Start and navigate the current directory tree interactively:

```
br
```

- Start displaying the size of files and directories:

```
br {{[-s|--sizes]}}
```

- Start displaying permissions:

```
br {{[-p|--permissions]}}
```

- Start displaying directories only:

```
br {{[-f|--only-folders]}}
```

- Start displaying hidden files and directories:

```
br {{[-h|--hidden]}}
```

braa

Ultra-fast mass SNMP scanner allowing multiple hosts simultaneously.

More information: https://github.com/mteg/braa.

- Walk the SNMP tree of host with public string querying all OIDs under `.1.3.6`:

```
braa public@{{ip}}:{{.1.3.6.*}}
```

- Query the whole subnet `ip_range` for `system.sysLocation.0`:

```
braa public@{{ip_range}}:{{.1.3.6.1.2.1.1.6.0}}
```

- Attempt to set the value of `system.sysLocation.0` to a specific workgroup:

```
braa private@{{ip}}:{{.1.3.6.1.2.1.1.6.0}}=s'{{workgroup}}'
```

brctl

Ethernet bridge administration.

More information: https://manned.org/brctl.

- Show a list with information about currently existing Ethernet bridges:

```
sudo brctl show
```

- Create a new Ethernet bridge interface:

```
sudo brctl add {{bridge_name}}
```

- Delete an existing Ethernet bridge interface:

```
sudo brctl del {{bridge_name}}
```

- Add an interface to an existing bridge:

```
sudo brctl addif {{bridge_name}} {{interface_name}}
```

- Remove an interface from an existing bridge:

```
sudo brctl delif {{bridge_name}} {{interface_name}}
```

bridge

Show and manipulate network bridge addresses and devices.

More information: https://manned.org/bridge.

- List all bridges and their interfaces:

```
bridge {{[l|link]}}
```

- Show port vlan information:

```
bridge {{[v|vlan]}}
```

- Assign a VLAN to a port:

```
sudo bridge {{[v|vlan]}} {{[a|add]}} dev {{lanX}} vid
{{vlan_id}} pvid {{tagged|untagged}}
```

- Remove a VLAN from a port:

```
sudo bridge {{[v|vlan]}} {{[d|delete]}} dev {{lanX}} vid
{{vlan_id}}
```

- Watch for changes in bridge interfaces:

```
bridge {{[mo|monitor]}}
```

- Display help:

```
bridge {{[h|help]}}
```

brightnessctl

Utility for reading and controlling device brightness for Linux operating systems.

More information: https://github.com/Hummer12007/brightnessctl#usage.

- List devices with changeable brightness:

```
brightnessctl {{[-l|--list]}}
```

- Print the current brightness of the default device:

```
brightnessctl {{[g|get]}}
```

- Print the current brightness of a specific device (can be a wildcard):

```
brightnessctl {{[g|get]}} {{[-d|--device]}} '{{device_name}}'
```

- Set the brightness to a specified percentage:

```
brightnessctl {{[s|set]}} {{50}}%
```

- Increase brightness by a specified percentage:

```
brightnessctl {{[s|set]}} +{{10}}%
```

- Decrease brightness by a specified percentage:

```
brightnessctl {{[s|set]}} {{10}}%-
```

broot

Navigate directory trees interactively.

See also: **`br`**.

More information: https://github.com/Canop/broot.

- Install or reinstall the `br` shell function:

```
broot --install
```

bspc

Configure and control **bspwm**, managing nodes, desktops, monitors, and more.

See also: **bspwm**.

More information: https://github.com/baskerville/bspwm.

- Define two virtual desktops:

```
bspc monitor {{[-d|--reset-desktops]}} {{desktop_name1}} {{desktop_name2}}
```

- Focus the given desktop:

```
bspc desktop {{[-f|--focus]}} {{number}}
```

- Close the windows rooted at the selected node:

```
bspc node {{[-c|--close]}}
```

- Send the selected node to the given desktop:

```
bspc node {{[-d|--to-desktop]}} {{number}}
```

- Toggle full screen mode for the selected node:

```
bspc node {{[-t|--state]}} ~fullscreen
```

- Set the value of a specific setting:

```
bspc config {{setting_name}} {{value}}
```

bspwm

A tiling window manager based on binary space partitioning.

See also: **`bspc`** for controlling it.

More information: https://github.com/baskerville/bspwm.

- Start `bspwm` (note that a pre-existing window manager must not be open when this command is run):

```
bspwm -c {{path/to/config}}
```

bsub

Submit batch jobs to LSF (Load Sharing Facility) scheduler.

More information: https://www.ibm.com/docs/spectrum-lsf/latest?topic=reference-bsub.

- Submit a script file as a job:

```
bsub {{path/to/script.sh}}
```

- Submit a job to a specific queue:

```
bsub -q {{queue_name}} make all
```

- Submit a job with a name and redirect output and error:

```
bsub -J {{job_name}} --output {{path/to/output.log}} --error
{{path/to/error.log}} {{path/to/script.sh}}
```

- Request 8 CPU cores and 16GB memory for a command:

```
bsub -n 8 -M 16G cargo build --release
```

- Run an interactive shell in the current session:

```
bsub -I bash
```

- Submit a job with a runtime limit of 45 minutes:

```
bsub -W 45 {{path/to/script.sh}}
```

btrbk

Create snapshots and remote backups of btrfs subvolumes.

More information: https://digint.ch/btrbk/doc/readme.html.

- Print statistics about configured subvolumes and snapshots:

```
sudo btrbk stats
```

- List configured subvolumes and snapshots:

```
sudo btrbk list
```

- Print what would happen in a run without making the displayed changes:

```
sudo btrbk {{[-v|--verbose]}} dryrun
```

- Run backup routines verbosely, show progress bar:

```
sudo btrbk --progress {{[-v|--verbose]}} run
```

- Only create snapshots for configured subvolumes:

```
sudo btrbk snapshot
```

btrfs balance

Balance block groups on a btrfs filesystem.

More information: https://btrfs.readthedocs.io/en/latest/btrfs-balance.html.

- Show the status of a running or paused balance operation:

```
sudo btrfs {{[b|balance]}} status {{path/to/btrfs_filesystem}}
```

- Balance all block groups (slow; rewrites all blocks in filesystem):

```
sudo btrfs {{[b|balance]}} start {{path/to/btrfs_filesystem}}
```

- Balance data block groups which are less than 15% utilized, running the operation in the background:

```
sudo btrfs {{[b|balance]}} start {{[--bg|--background]}} -dusage={{15}} {{path/to/btrfs_filesystem}}
```

- Balance a max of 10 metadata chunks with less than 20% utilization and at least 1 chunk on a given device `devid` (see `btrfs filesystem show`):

```
sudo btrfs {{[b|balance]}} start -musage={{20}},limit={{10}},devid={{devid}} {{path/to/btrfs_filesystem}}
```

- Convert data blocks to the raid6 and metadata to raid1c3 (see mkfs.btrfs(8) for profiles):

```
sudo btrfs {{[b|balance]}} start -dconvert={{raid6}} -mconvert={{raid1c3}} {{path/to/btrfs_filesystem}}
```

- Convert data blocks to raid1, skipping already converted chunks (e.g. after a previous cancelled conversion operation):

```
sudo btrfs {{[b|balance]}} start -dconvert={{raid1}},soft {{path/to/btrfs_filesystem}}
```

- Cancel, pause, or resume a running or paused balance operation:

```
sudo btrfs {{[b|balance]}} {{cancel|pause|resume}} {{path/to/btrfs_filesystem}}
```

btrfs check

Check or repair a btrfs filesystem.

More information: https://btrfs.readthedocs.io/en/latest/btrfs-check.html.

- Check a btrfs filesystem:

```
sudo btrfs {{[c|check]}} {{path/to/partition}}
```

- Check and repair a btrfs filesystem (dangerous):

```
sudo btrfs {{[c|check]}} --repair {{path/to/partition}}
```

- Show the progress of the check:

```
sudo btrfs {{[c|check]}} {{[-p|--progress]}} {{path/to/partition}}
```

- Verify the checksum of each data block (if the filesystem is good):

```
sudo btrfs {{[c|check]}} --check-data-csum {{path/to/partition}}
```

- Use the n-th superblock (n can be 0, 1 or 2):

```
sudo btrfs {{[c|check]}} {{[-s|--super]}} {{n}} {{path/to/partition}}
```

- Rebuild the checksum tree:

```
sudo btrfs {{[c|check]}} --repair --init-csum-tree {{path/to/partition}}
```

- Rebuild the extent tree:

```
sudo btrfs {{[c|check]}} --repair --init-extent-tree {{path/to/partition}}
```

btrfs device

Manage devices in a btrfs filesystem.

More information: https://btrfs.readthedocs.io/en/latest/btrfs-device.html.

- Add one or more devices to a btrfs filesystem:

```
sudo btrfs {{[d|device]}} {{[a|add]}} {{path/to/block_device1 path/to/block_device2 ...}} {{path/to/btrfs_filesystem}}
```

- Remove a device from a btrfs filesystem:

```
sudo btrfs {{[d|device]}} {{[rem|remove]}} {{path/to/device1|device_id1 path/to/device2|device_id2 ...}}
```

- Display error statistics:

```
sudo btrfs {{[d|device]}} {{[st|stats]}} {{path/to/btrfs_filesystem}}
```

- Scan all disks and inform the kernel of all detected btrfs filesystems:

```
sudo btrfs {{[d|device]}} {{[sc|scan]}} {{[-d|--all-devices]}}
```

- Display detailed per-disk allocation statistics:

```
sudo btrfs {{[d|device]}} {{[u|usage]}} {{path/to/btrfs_filesystem}}
```

btrfs filesystem

Manage btrfs filesystems.

More information: https://btrfs.readthedocs.io/en/latest/btrfs-filesystem.html.

- Show filesystem usage (optionally run as root to show detailed information):

```
btrfs {{[f|filesystem]}} {{[u|usage]}} {{path/to/btrfs_mount}}
```

- Show usage by individual devices:

```
sudo btrfs {{[f|filesystem]}} {{[sh|show]}} {{path/to/btrfs_mount}}
```

- Defragment a single file on a btrfs filesystem (avoid while a deduplication agent is running):

```
sudo btrfs {{[f|filesystem]}} {{[de|defragment]}} {{[-v|--verbose]}} {{path/to/file}}
```

- Defragment a directory recursively (does not cross subvolume boundaries):

```
sudo btrfs {{[f|filesystem]}} {{[de|defragment]}} {{[-v|--verbose]}} -r {{path/to/directory}}
```

- Force syncing unwritten data blocks to disk(s):

```
sudo btrfs {{[f|filesystem]}} {{[sy|sync]}} {{path/to/btrfs_mount}}
```

- Summarize disk usage for the files in a directory recursively:

```
sudo btrfs {{[f|filesystem]}} du {{[-s|--summarize]}} {{path/to/directory}}
```

- Create a swap file:

```
sudo btrfs {{[f|filesystem]}} {{[m|mkswapfile]}} --size {{8g}} --uuid {{clear|random|time|UUID_value}} {{path/to/swapfile}}
```

btrfs inspect-internal

Query internal information of a btrfs filesystem.

More information: https://btrfs.readthedocs.io/en/latest/btrfs-inspect-internal.html.

- Print superblock's information:

```
sudo btrfs {{[i|inspect-internal]}} {{[dump-s|dump-super]}} {{path/to/partition}}
```

- Print superblock's and all of its copies' information:

```
sudo btrfs {{[i|inspect-internal]}} {{[dump-s|dump-super]}} {{[-a|--all]}} {{path/to/partition}}
```

- Print filesystem's metadata information:

```
sudo btrfs {{[i|inspect-internal]}} {{[dump-t|dump-tree]}} {{path/to/partition}}
```

- Print list of files in inode n-th:

```
sudo btrfs {{[i|inspect-internal]}} {{[i|inode-resolve]}} {{n}} {{path/to/btrfs_mount}}
```

- Print list of files at a given logical address:

```
sudo btrfs {{[i|inspect-internal]}} {{[lo|logical-resolve]}} {{logical_address}} {{path/to/btrfs_mount}}
```

- Print stats of root, extent, csum and fs trees:

```
sudo btrfs {{[i|inspect-internal]}} {{[t|tree-stats]}} {{path/to/partition}}
```

btrfs property

Get, set, or list properties for a BTRFS filesystem object (files, directories, subvolumes, filesystems, or devices).

More information: https://btrfs.readthedocs.io/en/latest/btrfs-property.html.

- List available properties (and descriptions) for the given btrfs object:

```
sudo btrfs {{[p|property]}} {{[l|list]}} {{path/to/btrfs_object}}
```

- Get all properties for the given btrfs object:

```
sudo btrfs {{[p|property]}} {{[g|get]}} {{path/to/btrfs_object}}
```

- Get the `label` property for the given btrfs filesystem or device:

```
sudo btrfs {{[p|property]}} {{[g|get]}} {{path/to/btrfs_filesystem}} label
```

- Get all object type-specific properties for the given btrfs filesystem or device:

```
sudo btrfs {{[p|property]}} {{[g|get]}} -t {{subvol|filesystem|inode|device}} {{path/to/btrfs_filesystem}}
```

- Set the `compression` property for a given btrfs inode (either a file or directory):

```
sudo btrfs {{[p|property]}} {{[s|set]}} {{path/to/btrfs_inode}} compression {{zstd|zlib|lzo|none}}
```

btrfs rescue

Try to recover a damaged btrfs filesystem.

More information: https://btrfs.readthedocs.io/en/latest/btrfs-rescue.html.

- Rebuild the filesystem metadata tree (very slow):

```
sudo btrfs {{[resc|rescue]}} {{[ch|chunk-recover]}} {{path/to/partition}}
```

- Fix device size alignment related problems (e.g. unable to mount the filesystem with super total bytes mismatch):

```
sudo btrfs {{[resc|rescue]}} {{[fix-de|fix-device-size]}} {{path/to/partition}}
```

- Recover a corrupted superblock from correct copies (recover the root of filesystem tree):

```
sudo btrfs {{[resc|rescue]}} {{[s|super-recover]}} {{path/to/partition}}
```

- Recover from an interrupted transactions (fixes log replay problems):

```
sudo btrfs {{[resc|rescue]}} {{[z|zero-log]}} {{path/to/partition}}
```

- Create a `/dev/btrfs-control` control device when `mknod` is not installed:

```
sudo btrfs {{[resc|rescue]}} {{[c|create-control-device]}}
```

btrfs restore

Try to salvage files from a damaged btrfs filesystem.

More information: https://btrfs.readthedocs.io/en/latest/btrfs-restore.html.

- Restore all files from a btrfs filesystem to a given directory:

```
sudo btrfs {{[rest|restore]}} {{path/to/btrfs_device}} {{path/to/target_directory}}
```

- List (don't write) files to be restored from a btrfs filesystem:

```
sudo btrfs {{[rest|restore]}} {{[-D|--dry-run]}} {{path/to/btrfs_device}} {{path/to/target_directory}}
```

- Restore files matching a given `regex` ([c]ase-insensitive) files to be restored from a btrfs filesystem (all parent directories of target file(s) must match as well):

```
sudo btrfs {{[rest|restore]}} --path-regex {{regex}} -c {{path/to/btrfs_device}} {{path/to/target_directory}}
```

- Restore files from a btrfs filesystem using a specific root [t]ree `bytenr` (see `btrfs-find-root`):

```
sudo btrfs {{[rest|restore]}} -t {{bytenr}} {{path/to/btrfs_device}} {{path/to/target_directory}}
```

- Restore files from a btrfs filesystem (along with metadata, extended attributes, and Symlinks), overwriting files in the target:

```
sudo btrfs {{[rest|restore]}} {{[-m|--metadata]}} {{[-x|--xattr]}} {{[-S|--symlinks]}} {{[-o|--overwrite]}} {{path/to/btrfs_device}} {{path/to/target_directory}}
```

btrfs scrub

Scrub btrfs filesystems to verify data integrity.

It is recommended to run a scrub once a month.

More information: https://btrfs.readthedocs.io/en/latest/btrfs-scrub.html.

- Start a scrub:

```
sudo btrfs {{[sc|scrub]}} start {{path/to/btrfs_mount}}
```

- Show the status of an ongoing or last completed scrub:

```
sudo btrfs {{[sc|scrub]}} status {{path/to/btrfs_mount}}
```

- Cancel an ongoing scrub:

```
sudo btrfs {{[sc|scrub]}} {{[c|cancel]}} {{path/to/
btrfs_mount}}
```

- Resume a previously cancelled scrub:

```
sudo btrfs {{[sc|scrub]}} {{[r|resume]}} {{path/to/
btrfs_mount}}
```

- Start a scrub, but do not put the program in the [B]ackground:

```
sudo btrfs {{[sc|scrub]}} start -B {{path/to/btrfs_mount}}
```

- Start a scrub in quiet mode (does not print errors or statistics):

```
sudo btrfs {{[sc|scrub]}} start {{[-q|--quiet]}} {{path/to/
btrfs_mount}}
```

btrfs subvolume

Manage btrfs subvolumes and snapshots.

More information: https://btrfs.readthedocs.io/en/latest/btrfs-subvolume.html.

- Create a new empty subvolume:

```
sudo btrfs {{[su|subvolume]}} {{[c|create]}} {{path/to/new_subvolume}}
```

- List all subvolumes and snapshots in the specified filesystem:

```
sudo btrfs {{[su|subvolume]}} {{[l|list]}} {{path/to/btrfs_filesystem}}
```

- Delete a subvolume:

```
sudo btrfs {{[su|subvolume]}} {{[d|delete]}} {{path/to/subvolume}}
```

- Create a [r]ead-only snapshot of an existing subvolume:

```
sudo btrfs {{[su|subvolume]}} {{[sn|snapshot]}} -r {{path/to/source_subvolume}} {{path/to/target}}
```

- Create a read-write snapshot of an existing subvolume:

```
sudo btrfs {{[su|subvolume]}} {{[sn|snapshot]}} {{path/to/source_subvolume}} {{path/to/target}}
```

- Show detailed information about a subvolume:

```
sudo btrfs {{[su|subvolume]}} {{[sh|show]}} {{path/to/subvolume}}
```

btrfs version

Display btrfs-progs version.

More information: https://btrfs.readthedocs.io/en/latest/btrfs.html.

- Display help:

```
btrfs {{[v|version]}} --help
```

- Display version:

```
btrfs {{[v|version]}}
```

btrfs

A filesystem based on the copy-on-write (COW) principle for Linux.

Some subcommands such as **`device`** have their own usage documentation.

More information: https://btrfs.readthedocs.io/en/latest/btrfs.html.

- Create subvolume:

```
sudo btrfs {{[su|subvolume]}} {{[c|create]}} {{path/to/subvolume}}
```

- List subvolumes:

```
sudo btrfs {{[su|subvolume]}} {{[l|list]}} {{path/to/mount_point}}
```

- Show space usage information:

```
sudo btrfs {{[f|filesystem]}} df {{path/to/mount_point}}
```

- Enable quota:

```
sudo btrfs {{[qu|quota]}} {{[e|enable]}} {{path/to/subvolume}}
```

- Show quota:

```
sudo btrfs {{[qg|qgroup]}} {{[s|show]}} {{path/to/subvolume}}
```

bully

Brute-force the WPS pin of a wireless access point.

Necessary information must be gathered with **`airmon-ng`** and **`airodump-ng`** before using **`bully`**.

More information: https://salsa.debian.org/pkg-security-team/bully.

- Crack the password:

```
bully {{[-b|--bssid]}} "{{mac}}" {{[-c|--channel]}}
"{{channel}}" {{[-B|--bruteforce]}} "{{interface}}"
```

- Display help:

```
bully {{[-h|--help]}}
```

burpsuite

A GUI based application mainly used in web application penetration testing.

More information: https://portswigger.net/burp/documentation/desktop/troubleshooting/launch-from-command-line.

- Start Burp Suite:

```
burpsuite
```

- Start Burp Suite using the default configuration:

```
burpsuite --use-defaults
```

- Open a specific project file:

```
burpsuite --project-file={{path/to/file}}
```

- Load a specific configuration file:

```
burpsuite --config-file={{path/to/file}}
```

- Start without extensions:

```
burpsuite --disable-extensions
```

busctl

Introspect and monitor the D-Bus bus.

More information: https://www.freedesktop.org/software/systemd/man/busctl.html.

- Show all peers on the bus, by their service names:

```
busctl list
```

- Show process information and credentials of a bus service, a process, or the owner of the bus (if no parameter is specified):

```
busctl status {{service|pid}}
```

- Dump messages being exchanged. If no service is specified, show all messages on the bus:

```
busctl monitor {{service1 service2 ...}}
```

- Show an object tree of one or more services (or all services if no service is specified):

```
busctl tree {{service1 service2 ...}}
```

- Show interfaces, methods, properties and signals of the specified object on the specified service:

```
busctl introspect {{service}} {{path/to/object}}
```

- Retrieve the current value of one or more object properties:

```
busctl get-property {{service}} {{path/to/object}} {{interface_name}} {{property_name}}
```

- Invoke a method and show the response:

```
busctl call {{service}} {{path/to/object}} {{interface_name}} {{method_name}}
```

bwa

Burrows-Wheeler Alignment tool.

Short, low-divergent DNA sequences mapper against a large reference genome, such as the human genome.

More information: https://manned.org/bwa.

- Index the reference genome:

```
bwa index {{path/to/reference.fa}}
```

- Map single-end reads (sequences) to indexed genome using 32 [t]hreads and compress the result to save space:

```
bwa mem -t 32 {{path/to/reference.fa}} {{path/to/
read_single_end.fq.gz}} | gzip > {{path/to/
alignment_single_end.sam.gz}}
```

- Map pair-end reads (sequences) to the indexed genome using 32 [t]hreads and compress the result to save space:

```
bwa mem -t 32 {{path/to/reference.fa}} {{path/to/
read_pair_end_1.fq.gz}} {{path/to/read_pair_end_2.fq.gz}} |
gzip > {{path/to/alignment_pair_end.sam.gz}}
```

- Map pair-end reads (sequences) to the indexed genome using 32 [t]hreads with [M]arking shorter split hits as secondary for output SAM file compatibility in Picard software and compress the result:

```
bwa mem -M -t 32 {{path/to/reference.fa}} {{path/to/
read_pair_end_1.fq.gz}} {{path/to/read_pair_end_2.fq.gz}} |
gzip > {{path/to/alignment_pair_end.sam.gz}}
```

- Map pair-end reads (sequences) to indexed genome using 32 [t]hreads with FASTA/Q [C]omments (e.g. BC:Z:CGTAC) appending to a compressed result:

```
bwa mem -C -t 32 {{path/to/reference.fa}} {{path/to/
read_pair_end_1.fq.gz}} {{path/to/read_pair_end_2.fq.gz}} |
gzip > {{path/to/alignment_pair_end.sam.gz}}
```

bwrap

Run programs in a lightweight sandbox.

More information: https://manned.org/bwrap.

- Run a program in a read-only environment:

```
bwrap --ro-bind / / {{/bin/bash}}
```

- Give the environment access to devices, process information and create a `tmpfs` for it:

```
bwrap --dev-bind /dev /dev --proc /proc --ro-bind / / --
tmpfs /tmp {{/bin/bash}}
```

byzanz-record

Record the screen.

More information: https://manned.org/byzanz-record.

- Record the screen and write the recording to a file (by default, `byzanz-record` will only record for 10 seconds):

```
byzanz-record {{path/to/file.[byzanz|flv|gif|ogg|ogv|webm]}}
```

- Show information while and after recording:

```
byzanz-record {{[-v|--verbose]}} {{path/to/file.[byzanz|flv|gif|ogg|ogv|webm]}}
```

- Record the screen for a minute:

```
byzanz-record {{[-d|--duration]}} 60 {{path/to/file.[byzanz|flv|gif|ogg|ogv|webm]}}
```

- Delay recording for 10 seconds:

```
byzanz-record --delay 10 {{path/to/file.[byzanz|flv|gif|ogg|ogv|webm]}}
```

cacaclock

Display the current time as ASCII art.

More information: https://packages.debian.org/sid/caca-utils.

- Display the time:

```
cacaclock
```

- Change the font:

```
cacaclock {{[-f|--font]}} {{font}}
```

- Change the format using an `strftime` format specification:

```
cacaclock {{[-d|--dateformat]}} {{strftime_arguments}}
```

cacademo

Display a random ASCII art animation.

More information: https://packages.debian.org/sid/caca-utils.

- View an animation:

```
cacademo
```

cacafire

Display an animated ASCII fire.

More information: https://packages.debian.org/sid/caca-utils.

- Display the ASCII fire:

```
cacafire
```

cacaview

Display an image in PMN format.

More information: https://packages.debian.org/sid/caca-utils.

- Display an image:

```
cacaview {{path/to/image}}
```

caffeinate

Prevent desktop from sleeping.

More information: https://manned.org/caffeinate.

- Prevent desktop from sleeping (use <Ctrl c> to exit):

```
caffeinate
```

caffeine-indicator

Manually inhibit desktop idleness with a toggle.

More information: https://manned.org/caffeine-indicator.

- Manually inhibit desktop idleness with a toggle:

```
caffeine-indicator
```

caffeine

Prevent desktop idleness in full-screen mode.

More information: https://manned.org/caffeine.

- Start a caffeine server:

```
caffeine
```

- Display help:

```
caffeine {{[-h|--help]}}
```

- Display version:

```
caffeine {{[-V|--version]}}
```

cage

Open an applications in a kiosk mode.

See also: **`gamescope`**.

More information: https://github.com/cage-kiosk/cage/blob/master/cage.1.scd.

- Run an application:

```
cage {{application}}
```

- Give the application arguments:

```
cage -- {{application}} {{arguments}}
```

- Hide window [d]ecorations (this can lock you from accessing the terminal):

```
cage -d {{application}}
```

- Allow [s]witching terminals with `<Ctrl Alt F2>`:

```
cage -s {{application}}
```

- Display help:

```
cage -h
```

caja

Manage files and directories in the MATE desktop environment.

See also: **`nautilus`**, **`dolphin`**, **`thunar`**, **`ranger`**.

More information: https://manned.org/caja.

- Open the current user home directory:

```
caja
```

- Open specific directories in separate windows:

```
caja {{path/to/directory1 path/to/directory2 ...}}
```

- Open specific directories in tabs:

```
caja {{[-t|--tabs]}} {{path/to/directory1 path/to/
directory2 ...}}
```

- Open a directory with a specific window size:

```
caja {{[-g|--geometry]}} {{600}}x{{400}} {{path/to/
directory}}
```

- Close all windows:

```
caja {{[-q|--quit]}}
```

cal

Display a calendar with the current day highlighted.

More information: https://manned.org/cal.

- Display a calendar for the current month:

```
cal
```

- Display 3 months spanning the date:

```
cal {{[-3|--three]}}
```

- Display the whole calendar for the current year:

```
cal {{[-y|--year]}}
```

- Display the next twelve months:

```
cal {{[-Y|--twelve]}}
```

- Use Monday as the first day of the week:

```
cal {{[-m|--monday]}}
```

- Display a calendar for a specific year (4 digits):

```
cal {{year}}
```

- Display a calendar for a specific month and year:

```
cal {{month}} {{year}}
```

calcurse

A text-based calendar and scheduling application for the command-line.

More information: https://github.com/lfos/calcurse/blob/pu/doc/calcurse.1.txt.

- Start `calcurse` on interactive mode:

  ```
  calcurse
  ```

- Print the appointments and events for the current day and exit:

  ```
  calcurse {{[-a|--appointment]}}
  ```

- Remove all local calcurse items and import remote objects:

  ```
  calcurse-caldav --init=keep-remote
  ```

- Remove all remote objects and push local calcurse items:

  ```
  calcurse-caldav --init=keep-local
  ```

- Copy local objects to the CalDAV server and vice versa:

  ```
  calcurse-caldav --init=two-way
  ```

caligula

A user-friendly, lightweight TUI for disk imaging.

See also: **dd**.

More information: https://github.com/ifd3f/caligula.

- Flash an ISO to a drive:

```
caligula burn {{path/to/image.iso}}
```

- Flash an ISO to a drive without entering the hash interactively:

```
caligula burn {{path/to/image.iso}} {{[-s|--hash]}} {{hash}}
```

cam

Frontend tool for **`libcamera`**.

More information: https://libcamera.org/docs.html.

- List available cameras:

```
cam {{[-l|--list]}}
```

- List controls of a camera:

```
cam {{[-c|--camera]}} {{camera_index}} --list-controls
```

- Write frames to a folder:

```
cam {{[-c|--camera]}} {{camera_index}} {{[-C|--capture=]}}
{{frames_to_capture}} {{[-F|--file]}}
```

- Display camera feed in a window:

```
cam {{[-c|--camera]}} {{camera_index}} {{[-C|--capture]}}
{{[-S|--sdl]}}
```

cat

Print and concatenate files.

More information: https://www.gnu.org/software/coreutils/manual/html_node/cat-invocation.html.

- Print the contents of a file to `stdout`:

```
cat {{path/to/file}}
```

- Concatenate several files into an output file:

```
cat {{path/to/file1 path/to/file2 ...}} > {{path/to/output_file}}
```

- Append several files to an output file:

```
cat {{path/to/file1 path/to/file2 ...}} >> {{path/to/output_file}}
```

- Write to a file interactively:

```
cat > {{path/to/file}}
```

- Number all output lines:

```
cat {{[-n|--number]}} {{path/to/file}}
```

- Display all characters, including tabs, line endings, and non-printing characters:

```
cat {{[-A|--show-all]}} {{path/to/file}}
```

- Pass file contents to another program through `stdin`:

```
cat {{path/to/file}} | {{program}}
```

cbatticon

A lightweight and fast battery icon that sits in your system tray.

More information: https://github.com/valr/cbatticon.

- Show the battery icon in the system tray:

```
cbatticon
```

- Show the battery icon and set the update interval to 20 seconds:

```
cbatticon {{[-u|--update-interval]}} {{20}}
```

- List available icon types:

```
cbatticon {{[-t|--list-icon-types]}}
```

- Show the battery icon with a specific icon type:

```
cbatticon {{[-i|--icon-type]}} {{standard|notification|
symbolic}}
```

- List available power supplies:

```
cbatticon {{[-p|--list-power-supplies]}}
```

- Show the battery icon for a specific battery:

```
cbatticon {{BAT0}}
```

- Show the battery icon and which command to execute when the battery level reaches the set critical level:

```
cbatticon {{[-r|--critical-level]}} {{5}} {{[-c|--command-
critical-level]}} {{poweroff}}
```

cc

This command is an alias of **gcc**.

- View documentation for the original command:

```
tldr gcc
```

cdrdao

Read and write CDs in disc-at-once mode.

More information: https://manned.org/cdrdao.

- Read a CD and write its contents to a file:

```
cdrdao read-cd --device {{/dev/cdrom}} --read-raw
{{image.toc}}
```

cec-client

Manage serial bus CEC connections.

See also: **`cec-ctl`**.

More information: https://manned.org/man/cec-client.

- List all CEC adapters:

```
cec-client {{[-l|--list-devices]}}
```

- Start an interactive CEC session:

```
sudo cec-client
```

- Set the On-Screen Display name:

```
sudo cec-client {{[-o|--osd-name]}} {{name}}
```

- Send a single command:

```
echo {{on 0}} | sudo cec-client {{[-s|--single-command]}}
```

- Set a device to standby in interactive mode:

```
standby {{0}}
```

- Turn a device on in interactive mode:

```
on {{0}}
```

cec-ctl

Control kernel CEC devices.

See also: **`cec-client`**.

More information: https://manned.org/cec-ctl.

- List CEC devices:

```
cec-ctl --list-devices
```

- Monitor CEC traffic:

```
sudo cec-ctl {{[-m|--monitor]}}
```

- Show CEC topology:

```
cec-ctl {{[-S|--show-topology]}}
```

ceph

A unified storage system.

More information: https://ceph.io/en.

- Check cluster health status:

  ```
  ceph status
  ```

- Check cluster usage stats:

  ```
  ceph df
  ```

- Get the statistics for the placement groups in a cluster:

  ```
  ceph pg dump --format {{plain}}
  ```

- Create a storage pool:

  ```
  ceph osd pool create {{pool_name}} {{page_number}}
  ```

- Delete a storage pool:

  ```
  ceph osd pool delete {{pool_name}}
  ```

- Rename a storage pool:

  ```
  ceph osd pool rename {{current_name}} {{new_name}}
  ```

- Self-repair pool storage:

  ```
  ceph pg repair {{pool_name}}
  ```

certbot

The Let's Encrypt Agent for automatically obtaining and renewing TLS certificates.

Successor to **`letsencrypt`**.

More information: https://eff-certbot.readthedocs.io/en/latest/using.html.

- Obtain a new certificate via webroot authorization, but do not install it automatically:

```
sudo certbot certonly --webroot {{[-w|--webroot-path]}}
{{path/to/webroot}} {{[-d|--domain]}}
{{subdomain.example.com}}
```

- Obtain a new certificate via nginx authorization, installing the new certificate automatically:

```
sudo certbot --nginx {{[-d|--domain]}}
{{subdomain.example.com}}
```

- Obtain a new certificate via apache authorization, installing the new certificate automatically:

```
sudo certbot --apache {{[-d|--domain]}}
{{subdomain.example.com}}
```

- Renew all Let's Encrypt certificates that expire in 30 days or less (don't forget to restart any servers that use them afterwards):

```
sudo certbot renew
```

- Simulate the obtaining of a new certificate, but don't actually save any new certificates to disk:

```
sudo certbot --webroot {{[-w|--webroot-path]}} {{path/to/
webroot}} {{[-d|--domain]}} {{subdomain.example.com}} --dry-
run
```

- Obtain an untrusted test certificate instead:

```
sudo certbot --webroot {{[-w|--webroot-path]}} {{path/to/
webroot}} {{[-d|--domain]}} {{subdomain.example.com}} --test-
cert
```

certtool

Generate and manage X.509 certificates, keys, and PKI structures using GnuTLS.

More information: https://gnutls.org/manual/gnutls.html#certtool-Invocation.

- Generate a private key and save it to a file:

```
certtool {{[-p|--generate-privkey]}} --outfile {{path/to/private.key}}
```

- Generate a self-signed certificate using a private key and a template file:

```
certtool {{[-s|--generate-self-signed]}} --load-privkey {{path/to/private.key}} --template {{path/to/info.template}} --outfile {{path/to/certificate.crt}}
```

- Generate a certificate signing request (CSR):

```
certtool {{[-q|--generate-request]}} --load-privkey {{path/to/private.key}} --template {{path/to/info.template}} --outfile {{path/to/request.csr}}
```

- Generate a certificate authority (CA) certificate:

```
certtool {{[-s|--generate-self-signed]}} --load-privkey {{path/to/ca.key}} --template {{path/to/ca.template}} --outfile {{path/to/ca.crt}}
```

- Verify a certificate against a CA certificate:

```
certtool --verify --infile {{path/to/certificate.crt}} --load-ca-certificate {{path/to/ca.crt}}
```

cewl

URL spidering tool for making a cracking wordlist from web content.

More information: https://digi.ninja/projects/cewl.php#usage.

- Create a wordlist file from the given URL up to 2 links depth:

```
cewl {{[-d|--depth]}} 2 {{[-w|--write]}} {{path/to/wordlist.txt}} {{url}}
```

- Output an alphanumeric wordlist from the given URL with words of minimum 5 characters:

```
cewl --with-numbers {{[-m|--min_word_length]}} 5 {{url}}
```

- Output a wordlist from the given URL in debug mode including email addresses:

```
cewl --debug {{[-e|--email]}} {{url}}
```

- Output a wordlist from the given URL using HTTP Basic or Digest authentication:

```
cewl --auth_type {{basic|digest}} --auth_user {{username}} --auth_pass {{password}} {{url}}
```

- Output a wordlist from the given URL through a proxy:

```
cewl --proxy_host {{host}} --proxy_port {{port}} {{url}}
```

cfdisk

Manage partition tables and partitions on a hard disk using a curses UI.

More information: https://manned.org/cfdisk.

- Start the partition manipulator with a specific device:

```
sudo cfdisk {{/dev/sdX}}
```

- Create a new partition table for a specific device and manage it:

```
sudo cfdisk {{[-z|--zero]}} {{/dev/sdX}}
```

cgclassify

Move running tasks to **`cgroups`**.

More information: https://manned.org/cgclassify.

- Move the process with a specific PID to the control [g]roup student in the CPU hierarchy:

  ```
  cgclassify -g {{cpu:student}} {{1234}}
  ```

- Move the process with a specific PID to control groups based on the `/etc/cgrules.conf` configuration file:

  ```
  cgclassify {{1234}}
  ```

- Move the process with a specific PID to the control [g]roup student in the CPU hierarchy. Note: The daemon of the service `cgred` does not change `cgroups` of the specific PID and its children (based on `/etc/cgrules.conf`):

  ```
  cgclassify --sticky -g {{cpu:/student}} {{1234}}
  ```

cgcreate

Create cgroups, used to limit, measure, and control resources used by processes.

`cgroups` types can be **`memory`**, **`cpu`**, **`net_cls`**, etc.

More information: https://manned.org/cgcreate.

- Create a new [g]roup:

```
cgcreate -g {{group_type}}:{{group_name}}
```

- Create a new [g]roup with multiple cgroup types:

```
cgcreate -g {{group_type1}},{{group_type2}}:{{group_name}}
```

- Create a subgroup:

```
mkdir /sys/fs/cgroup/{{group_type}}/{{group_name}}/
{{subgroup_name}}
```

cgexec

Limit, measure, and control resources used by processes.

Multiple cgroup types (aka controllers) exist, such as **`cpu`**, **`memory`**, etc.

More information: https://manned.org/cgexec.

- Execute a process in a given c[g]roup with given controller:

```
cgexec -g {{controller}}:{{cgroup_name}} {{process_name}}
```

cgroups

Cgroups aka control groups is a Linux kernel feature for limiting, measuring, and controlling resource usage by processes.

Cgroups however is not a command, but rather a collection of commands, see the relevant pages below.

More information: https://www.kernel.org/doc/Documentation/cgroup-v2.txt.

- View documentation for `cgclassify`:

```
tldr cgclassify
```

- View documentation for `cgcreate`:

```
tldr cgcreate
```

- View documentation for `cgexec`:

```
tldr cgexec
```

chage

Change user account and password expiry information.

More information: https://manned.org/chage.

- List password information for the user:

```
chage {{[-l|--list]}} {{username}}
```

- Enable password expiration in 10 days:

```
sudo chage {{[-M|--maxdays]}} {{10}} {{username}}
```

- Disable password expiration:

```
sudo chage {{[-M|--maxdays]}} {{-1}} {{username}}
```

- Set account expiration date:

```
sudo chage {{[-E|--expiredate]}} {{YYYY-MM-DD}} {{username}}
```

- Force user to change password on next log in:

```
sudo chage {{[-d|--lastday]}} {{0}} {{username}}
```

- Re-enable an account:

```
sudo chage {{[-E|--expiredate]}} -1 {{username}}
```

chat

Automate conversations with a modem or serial device.

Commonly used to establish PPP (Point-to-Point Protocol) connections.

More information: https://manned.org/chat.8.

- Execute a chat script directly from the command line:

```
chat '{{expect_send_pairs}}'
```

- Execute a chat script from a file:

```
chat -f '{{path/to/chat_script}}'
```

- Set a custom timeout (in seconds) for expecting a response:

```
chat -t {{timeout_in_seconds}} '{{expect_send_pairs}}'
```

- Enable verbose output to log the conversation to `syslog`:

```
chat -v '{{expect_send_pairs}}'
```

- Use a report file to log specific strings received during the conversation:

```
chat -r {{path/to/report_file}} '{{expect_send_pairs}}'
```

- Dial a phone number using a variable, substituting `\T` in the script:

```
chat -T '{{phone_number}}' '{{"ATDT\\T CONNECT"}}'
```

- Include an abort condition if a specific string is received:

```
chat 'ABORT "{{error_string}}" {{expect_send_pairs}}'
```

chattr

Change attributes of files or directories.

More information: https://manned.org/chattr.

- Make a file or directory [i]mmutable to changes and deletion, even by superuser:

```
chattr +i {{path/to/file_or_directory}}
```

- Make a file or directory mutable:

```
chattr -i {{path/to/file_or_directory}}
```

- [R]ecursively make an entire directory and contents immutable:

```
chattr -R +i {{path/to/directory}}
```

- Mark a directory and its files to be interpreted in a case-insensitive manner (case-[F]olding):

```
chattr +F {{path/to/directory}}
```

- Set a file to only allow [a]ppending:

```
chattr +a {{path/to/file}}
```

chcat

Change SELinux security category for files.

Categories provide an additional level of access control based on MCS (Multi-Category Security).

See also: **chcon**, **semanage**.

More information: https://manned.org/chcat.

- List all available categories:

```
sudo chcat {{[-L|--list]}}
```

- Add a category to a file:

```
sudo chcat +{{CategoryName}} {{path/to/file}}
```

- Remove a category from a file:

```
sudo chcat -- -{{CategoryName}} {{path/to/file}}
```

- Set specific categories for a file (replacing existing ones):

```
sudo chcat {{CategoryName1,CategoryName2,...}} {{path/to/file}}
```

- Display the categories of a file:

```
ls {{[-Z|--context]}} {{path/to/file}}
```

- Remove all categories from a file:

```
sudo chcat {{[-d|--delete]}} {{path/to/file}}
```

chcon

Change SELinux security context of a file or files/directories.

See also: **`secon`**, **`restorecon`**, **`semanage-fcontext`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/chcon-invocation.html.

- View security context of a file:

```
ls {{[-lZ|-l --context]}} {{path/to/file}}
```

- Change the security context of a target file, using a reference file:

```
chcon --reference {{reference_file}} {{target_file}}
```

- Change the full SELinux security context of a file:

```
chcon {{user}}:{{role}}:{{type}}:{{range/level}} {{filename}}
```

- Change only the user part of SELinux security context:

```
chcon {{[-u|--user]}} {{user}} {{filename}}
```

- Change only the role part of SELinux security context:

```
chcon {{[-r|--role]}} {{role}} {{filename}}
```

- Change only the type part of SELinux security context:

```
chcon {{[-t|--type]}} {{type}} {{filename}}
```

- Change only the range/level part of SELinux security context:

```
chcon {{[-l|--range]}} {{range/level}} {{filename}}
```

chcpu

Enable/disable a system's CPUs.

More information: https://manned.org/chcpu.

- Disable one or more CPUs by their IDs:

```
chcpu {{[-d|--disable]}} {{1,3}}
```

- Enable one or more ranges of CPUs by their IDs:

```
chcpu {{[-e|--enable]}} {{1-3,5-7}}
```

check-dfsg-status

Report non-free packages installed on Debian-based OSes.

This command was formerly known as **`vrms`**.

More information: https://salsa.debian.org/debian/check-dfsg-status.

- List non-free and contrib packages (and their description):

```
check-dfsg-status
```

- Only output the package names:

```
check-dfsg-status {{[-s|--sparse]}}
```

check-language-support

Display a list of missing language packages on Ubuntu.

More information: https://manned.org/check-language-support.

- Display a list of missing language packages based on installed software and enabled locales:

```
check-language-support
```

- List packages for a specific locale:

```
check-language-support {{[-l|--language]}} {{en}}
```

- Display installed packages as well as missing ones:

```
check-language-support --show-installed
```

check-support-status

Identify installed Debian packages for which support has had to be limited or prematurely ended.

More information: https://manned.org/check-support-status.

- Display packages whose support is limited, has already ended or will end earlier than the distribution's end of life:

```
check-support-status
```

- Display only packages whose support has ended:

```
check-support-status --type {{ended}}
```

- Skip printing a headline:

```
check-support-status --no-heading
```

checkinstall

Track the local installation of a software package, and produce a binary package which can be used with a system's native package manager.

More information: https://checkinstall.izto.org/docs.php.

- Create and install a package with default settings:

```
sudo checkinstall {{[-y|--default]}}
```

- Create a package but don't install it:

```
sudo checkinstall --install={{no}}
```

- Create a package without documentation:

```
sudo checkinstall --nodoc
```

- Create a package and set the name:

```
sudo checkinstall --pkgname {{package}}
```

- Create a package and specify where to save it:

```
sudo checkinstall --pakdir {{path/to/directory}}
```

checkupdates-aur

Check pending updates from the Arch User Repository (AUR).

More information: https://metacpan.org/dist/OS-CheckUpdates-AUR.

- List pending updates for AUR packages:

```
checkupdates-aur
```

- List pending updates for AUR packages in debug mode:

```
CHECKUPDATES_DEBUG=1 checkupdates-aur
```

- Display help:

```
checkupdates-aur --help
```

checkupdates

Check pending updates in Arch Linux.

More information: https://manned.org/checkupdates.

- Synchronize the database and list pending updates:

```
checkupdates
```

- List pending updates without syncing the database:

```
checkupdates {{[-n|--nosync]}}
```

- Display the list of pending updates if it differs from the last time this option was used:

```
checkupdates {{[-c|--change]}}
```

- List pending updates and download the packages to the `pacman` cache (`/var/cache/pacman/pkg`):

```
checkupdates {{[-d|--download]}}
```

- List pending updates using a specific `pacman` database:

```
CHECKUPDATES_DB={{path/to/directory}} checkupdates
```

- Display help:

```
checkupdates {{[-h|--help]}}
```

chfn

Update **`finger`** info for a user.

More information: https://manned.org/chfn.

- Update a user's "Name" field in the output of `finger`:

```
chfn {{[-f|--full-name]}} {{new_display_name}} {{username}}
```

- Update a user's "Office Room Number" field for the output of `finger`:

```
chfn {{[-o|--office]}} {{new_office_room_number}} {{username}}
```

- Update a user's "Office Phone Number" field for the output of `finger`:

```
chfn {{[-p|--office-phone]}} {{new_office_telephone_number}} {{username}}
```

- Update a user's "Home Phone Number" field for the output of `finger`:

```
chfn {{[-h|--home-phone]}} {{new_home_telephone_number}} {{username}}
```

chkconfig

Manage the runlevel of services on CentOS 6.

More information: https://manned.org/chkconfig.

- List services with runlevel:

```
chkconfig --list
```

- Show a service's runlevel:

```
chkconfig --list {{ntpd}}
```

- Enable service at boot:

```
chkconfig {{sshd}} on
```

- Enable service at boot for runlevels 2, 3, 4, and 5:

```
chkconfig --level {{2345}} {{sshd}} on
```

- Disable service at boot:

```
chkconfig {{ntpd}} off
```

- Disable service at boot for runlevel 3:

```
chkconfig --level {{3}} {{ntpd}} off
```

chmem

Modify the state of memory blocks (online or offline) in a Linux system.

Typically used in virtualized environments to manage memory hotplug.

More information: https://manned.org/chmem.

- Set a memory block offline:

```
sudo chmem {{[-b|--block]}} {{[-d|--disable]}}
{{block_number}}
```

- Set a memory block online:

```
sudo chmem {{[-b|--block]}} {{[-e|--enable]}}
{{block_number}}
```

- Set a memory range offline using hexadecimal addresses:

```
sudo chmem {{[-d|--disable]}}
0x{{start_address}}-0x{{end_address}}
```

- Set a memory range online using hexadecimal addresses:

```
sudo chmem {{[-e|--enable]}}
0x{{start_address}}-0x{{end_address}}
```

- Set memory online and assign it to a specific zone (e.g., Movable):

```
sudo chmem {{[-e|--enable]}} 0x{{start_address}} {{[-z|--
zone]}} {{Movable}}
```

- Display help:

```
chmem {{[-h|--help]}}
```

chntpw

A utility that can edit windows registry, reset user password, promote users to administrator by modifying the Windows SAM.

Boot target machine with live cd like Kali Linux and run with elevated privileges.

More information: https://pogostick.net/~pnh/ntpasswd/MANUAL.txt.

- List all users in the SAM file:

```
chntpw -l {{path/to/sam_file}}
```

- Edit user interactively:

```
chntpw -u {{username}} {{path/to/sam_file}}
```

- Use chntpw interactively:

```
chntpw -i {{path/to/sam_file}}
```

choom

Display and change the adjust out-of-memory killer score.

More information: https://manned.org/choom.

- Display the OOM-killer score of the process with a specific ID:

```
choom {{[-p|--pid]}} {{pid}}
```

- Change the adjust OOM-killer score of a specific process:

```
choom {{[-p|--pid]}} {{pid}} {{[-n|--adjust]}}
{{-1000..+1000}}
```

- Run a command with a specific adjust OOM-killer score:

```
choom {{[-n|--adjust]}} {{-1000..+1000}} {{command}}
{{argument1 argument2 ...}}
```

chpasswd

Change the passwords for multiple users by using **`stdin`**.

More information: https://manned.org/chpasswd.8.

- Change the password for a specific user:

```
printf "{{username}}:{{new_password}}" | sudo chpasswd
```

- Change the passwords for multiple users (The input text must not contain any spaces.):

```
printf "{{username_1}}:{{new_password_1}}\n{{username_2}}:
{{new_password_2}}" | sudo chpasswd
```

- Change the password for a specific user, and specify it in encrypted form:

```
printf "{{username}}:{{new_encrypted_password}}" | sudo
chpasswd --encrypted
```

- Change the password for a specific user, and use a specific encryption for the stored password:

```
printf "{{username}}:{{new_password}}" | sudo chpasswd --
crypt-method {{NONE|DES|MD5|SHA256|SHA512}}
```

chrt

Manipulate the real-time attributes of a process.

More information: https://manned.org/chrt.

- Display attributes of a process:

```
chrt {{[-p|--pid]}} {{PID}}
```

- Display attributes of all threads of a process:

```
chrt {{[-a|--all-tasks]}} {{[-p|--pid]}} {{PID}}
```

- Display the min/max priority values that can be used with `chrt`:

```
chrt {{[-m|--max]}}
```

- Set the scheduling priority of a process:

```
chrt {{[-p|--pid]}} {{priority}} {{PID}}
```

- Set the scheduling policy of a process:

```
chrt --{{deadline|idle|batch|rr|fifo|other}} {{[-p|--pid]}}
{{priority}} {{PID}}
```

chsh

Change user's login shell.

Part of **util-linux**.

More information: https://manned.org/chsh.

- Set a specific login shell for the current user interactively:

```
chsh
```

- List available shells:

```
chsh {{[-l|--list-shells]}}
```

- Set a specific login shell for the current user:

```
chsh {{[-s|--shell]}} {{path/to/shell}}
```

- Set a login shell for a specific user:

```
sudo chsh {{[-s|--shell]}} {{path/to/shell}} {{username}}
```

chvt

Change foreground virtual terminal.

More information: https://manned.org/chvt.

- Change to a specific TTY:

```
sudo chvt {{3}}
```

cliphist

Manage clipboard history for Wayland compositors.

Works with **`wl-copy`** and **`wl-paste`**.

More information: https://github.com/sentriz/cliphist.

- List clipboard history entries:

```
cliphist list
```

- Select and copy a previous clipboard entry (using `fzf`):

```
cliphist list | fzf | cliphist decode | wl-copy
```

- Delete all stored clipboard entries:

```
cliphist wipe
```

- Delete a specific clipboard entry by ID:

```
cliphist delete {{id}}
```

- Save the current clipboard content manually:

```
wl-paste | cliphist store
```

cloud-init

Manage cloud instance initialization.

More information: https://cloudinit.readthedocs.io/en/latest/reference/cli.html.

- Display the status of the most recent cloud-init run:

```
cloud-init status
```

- Wait for cloud-init to finish running and then report status:

```
cloud-init status --wait
```

- List available top-level metadata keys to query:

```
cloud-init query --list-keys
```

- Query cached instance metadata for data:

```
cloud-init query {{dot_delimited_variable_path}}
```

- Clean logs and artifacts to allow cloud-init to rerun:

```
cloud-init clean
```

cmus

Command-line Music Player.

Use **<ArrowKeys>** to navigate, **<Enter>** to select, and numbers **<1>**-**<8>** switch between different views.

See also: **ncmpcpp**, **clementine**, **qmmp**.

More information: https://manned.org/cmus.

- Open `cmus` into the specified directory (this will become your new working directory):

  `cmus {{path/to/directory}}`

- Add file/directory to library:

  `<:>add {{path/to/file_or_directory}}`

- Refresh the metadata of songs in the library:

  `<:>update-cache`

- Search for songs, albums, or artists:

  `</>{{something}}`

- Pause/unpause current song:

  `<c>`

- Toggle shuffle mode on/off:

  `<s>`

- Quit `cmus`:

  `<q>`

cockpit-bridge

Relay messages and commands between the front end and server in the cockpit suite.

More information: https://cockpit-project.org/guide/latest/cockpit-bridge.1.html.

- List all cockpit packages:

```
cockpit-bridge --packages
```

- Display help:

```
cockpit-bridge --help
```

cockpit-desktop

Securely access Cockpit pages in a running session.

It starts **`cockpit-ws`** and a web browser in an isolated network space and a **`cockpit-bridge`** in a running user session.

More information: https://cockpit-project.org/guide/latest/cockpit-desktop.1.html.

- Open a page:

```
cockpit-desktop {{url}} {{SSH_host}}
```

- Open storage page:

```
cockpit-desktop {{/cockpit/@localhost/storage/index.html}}
```

cockpit-tls

TLS terminating HTTP proxy to encrypt traffic between a client and **`cockpit-ws`**.

More information: https://cockpit-project.org/guide/latest/cockpit-tls.8.html.

- Serve HTTP requests to a specific port instead of port `9090`:

  ```
  cockpit-tls --port {{port}}
  ```

- Display help:

  ```
  cockpit-tls --help
  ```

cockpit-ws

Communicate between the browser application and various configuration tools and services like **`cockpit-bridge`**.

More information: https://cockpit-project.org/guide/latest/cockpit-ws.8.html.

- Start with authentication via SSH at `127.0.0.1` with port `22` enabled:

  ```
  cockpit-ws --local-ssh
  ```

- Start an HTTP server on a specific port:

  ```
  cockpit-ws --port {{port}}
  ```

- Start and bind to a specific IP address (defaults to `0.0.0.0`):

  ```
  cockpit-ws --address {{ip_address}}
  ```

- Start without TLS:

  ```
  cockpit-ws --no-tls
  ```

- Display help:

  ```
  cockpit-ws --help
  ```

cointop

Track and monitor cryptocurrencies in the terminal.

More information: https://github.com/cointop-sh/cointop.

- Open TUI:

```
cointop
```

- Clear the cache:

```
cointop clean
```

- Display current holdings legibly:

```
cointop holdings --human
```

- Check price of coin(s):

```
cointop price --coins {{coin_name1,coin_name2,...}}
```

- Display version:

```
cointop version
```

collectd

System statistics collection daemon.

More information: https://manned.org/collectd.

- Test the configuration file and then exit:

  `collectd -t`

- Test plugin data collection functionality and then exit:

  `collectd -T`

- Start `collectd`:

  `collectd`

- Specify a custom configuration file location:

  `collectd -C {{path/to/file}}`

- Specify a custom PID file location:

  `collectd -P {{path/to/file}}`

- Don't fork into the background:

  `collectd -f`

- Display help and version:

  `collectd -h`

colrm

Remove columns from **`stdin`**.

More information: https://manned.org/colrm.

- Remove first column of `stdin`:

```
colrm {{1 1}}
```

- Remove from 3rd column till the end of each line:

```
colrm {{3}}
```

- Remove from the 3rd column till the 5th column of each line:

```
colrm {{3 5}}
```

column

Format **`stdin`** or a file into multiple columns.

Columns are filled before rows; the default separator is a whitespace.

More information: https://manned.org/column.

- Format the output of a command for a 30 characters wide display:

```
printf "header1 header2\nbar foo\n" | column {{[-c|--output-width]}} {{30}}
```

- Split columns automatically and auto-align them in a tabular format:

```
printf "header1 header2\nbar foo\n" | column {{[-t|--table]}}
```

- Specify the column delimiter character for the `--table` option (e.g. "," for CSV) (defaults to whitespace):

```
printf "header1,header2\nbar,foo\n" | column {{[-t|--table]}} {{[-s|--separator]}} {{,}}
```

- Fill rows before filling columns:

```
printf "header1\nbar\nfoobar\n" | column {{[-c|--output-width]}} {{30}} {{[-x|--fillrows]}}
```

compose

An alias to a **`run-mailcap`**'s action compose.

Originally **`run-mailcap`** is used to mime-type/file.

More information: https://manned.org/compose.

- Compose action can be used to compose any existing file or new on default mailcap edit tool:

  ```
  compose {{filename}}
  ```

- With `run-mailcap`:

  ```
  run-mailcap --action=compose {{filename}}
  ```

compress

Compress files using the Unix **`compress`** command.

More information: https://manned.org/compress.1.

- Compress specific files:

```
compress {{path/to/file1 path/to/file2 ...}}
```

- Compress specific files, ignore non-existent ones:

```
compress -f {{path/to/file1 path/to/file2 ...}}
```

- Specify the maximum compression bits (9-16 bits):

```
compress -b {{bits}}
```

- Write to `stdout` (no files are changed):

```
compress -c {{path/to/file}}
```

- Decompress files (functions like `uncompress`):

```
compress -d {{path/to/file}}
```

- Display compression percentage:

```
compress -v {{path/to/file}}
```

compseq

Calculate the composition of unique words in sequences.

More information: https://www.bioinformatics.nl/cgi-bin/emboss/help/compseq/.

- Count observed frequencies of words in a FASTA file, providing parameter values with interactive prompt:

```
compseq {{path/to/file.fasta}}
```

- Count observed frequencies of amino acid pairs from a FASTA file, save output to a text file:

```
compseq {{path/to/input_protein.fasta}} -word 2 {{path/to/output_file.comp}}
```

- Count observed frequencies of hexanucleotides from a FASTA file, save output to a text file and ignore zero counts:

```
compseq {{path/to/input_dna.fasta}} -word 6 {{path/to/output_file.comp}} -nozero
```

- Count observed frequencies of codons in a particular reading frame; ignoring any overlapping counts (i.e. move window across by word-length 3):

```
compseq -sequence {{path/to/input_rna.fasta}} -word 3 {{path/to/output_file.comp}} -nozero -frame {{1}}
```

- Count observed frequencies of codons frame-shifted by 3 positions; ignoring any overlapping counts (should report all codons except the first one):

```
compseq -sequence {{path/to/input_rna.fasta}} -word 3 {{path/to/output_file.comp}} -nozero -frame 3
```

- Count amino acid triplets in a FASTA file and compare to a previous run of `compseq` to calculate expected and normalised frequency values:

```
compseq -sequence {{path/to/human_proteome.fasta}} -word 3 {{path/to/output_file1.comp}} -nozero -infile {{path/to/output_file2.comp}}
```

- Approximate the above command without a previously prepared file, by calculating expected frequencies using the single base/residue frequencies in the supplied input sequence(s):

```
compseq -sequence {{path/to/human_proteome.fasta}} -word 3
{{path/to/output_file.comp}} -nozero -calcfreq
```

- Display help (use `-help -verbose` for more information on associated and general qualifiers):

```
compseq -help
```

compsize

Calculate the compression ratio of a set of files on a btrfs filesystem.

See also: **`btrfs filesystem`** for recompressing a file by defragmenting it.

More information: https://manned.org/compsize.

- Calculate the current compression ratio for a file or directory:

```
sudo compsize {{path/to/file_or_directory}}
```

- Don't traverse filesystem boundaries:

```
sudo compsize {{[-x|--one-file-system]}} {{path/to/
file_or_directory}}
```

- Show raw byte counts instead of human-readable sizes:

```
sudo compsize {{[-b|--bytes]}} {{path/to/file_or_directory}}
```

conky

Light-weight system monitor for X.

More information: https://github.com/brndnmtthws/conky.

- Launch with default, built-in config:

```
conky
```

- Create a new default config:

```
conky {{[-C|--print-config]}} > ~/.conkyrc
```

- Launch Conky with a given configuration file:

```
conky {{[-c|--config]}} {{path/to/config}}
```

- Start in the background (daemonize):

```
conky {{[-d|--daemonize]}}
```

- Align Conky on the desktop:

```
conky {{[-a|--alignment]}} {{top|bottom|middle}}_{{left|
right|middle}}
```

- Pause for 5 seconds at startup before launching:

```
conky {{[-p|--pause]}} {{5}}
```

conntrack

Interact with the Netfilter connection tracking system.

Search, list, inspect, modify, and delete connection flows.

More information: https://manned.org/conntrack.

- List all currently tracked connections:

```
conntrack {{[-L|--dump]}}
```

- Display a real-time event log of connection changes:

```
conntrack {{[-E|--event]}}
```

- Display a real-time event log of connection changes and associated timestamps:

```
conntrack {{[-E|--event]}} {{[-o|--output]}} timestamp
```

- Display a real-time event log of connection changes for a specific IP address:

```
conntrack {{[-E|--event]}} {{[-s|--orig-src]}} {{ip_address}}
```

- Delete all flows for a specific source IP address:

```
conntrack {{[-D|--delete]}} {{[-s|--orig-src]}}
{{ip_address}}
```

coredumpctl

Retrieve and process saved core dumps and metadata.

More information: https://www.freedesktop.org/software/systemd/man/coredumpctl.html.

- List all captured core dumps:

```
coredumpctl
```

- List captured core dumps for a program:

```
coredumpctl list {{program}}
```

- Show information about the core dumps matching a program with `PID`:

```
coredumpctl info {{PID}}
```

- Invoke debugger using the last core dump:

```
coredumpctl debug
```

- Invoke debugger using the last core dump of a program:

```
coredumpctl debug {{program}}
```

- Extract the last core dump of a program to a file:

```
coredumpctl {{[-o|--output]}} {{path/to/file}} dump {{program}}
```

- Skip debuginfod and pagination prompts and then print the backtrace when using `gdb`:

```
coredumpctl debug {{[-A|--debugger-arguments]}} "-iex 'set debuginfod enabled on' -iex 'set pagination off' -ex bt"
```

Counter Strike 2

Host a headless Counter Strike 2 server.

More information: https://developer.valvesoftware.com/wiki/Counter-Strike_2/Dedicated_Servers.

- Run a game with one map:

```
{{path/to/cs2}} -dedicated +map {{de_dust2}}
```

- Run a game with specified maximum number of players:

```
{{path/to/cs2}} -dedicated +map {{de_dust2}} -maxplayers {{64}}
```

- Run a game with specified server IP and port:

```
{{path/to/cs2}} -dedicated +map {{de_dust2}} -ip {{1.2.3.4}} -port {{27015}}
```

- Shut the server down:

```
quit
```

cpufreq-aperf

Calculate the average CPU frequency over a time period.

Requires root privileges.

More information: https://manned.org/cpufreq-aperf.

- Start calculating, defaulting to all CPU cores and 1 second refresh interval:

```
sudo cpufreq-aperf
```

- Start calculating for CPU 1 only:

```
sudo cpufreq-aperf {{[-c|--cpu]}} {{1}}
```

- Start calculating with a 3 second refresh interval for all CPU cores:

```
sudo cpufreq-aperf {{[-i|--interval]}} {{3}}
```

- Calculate only once:

```
sudo cpufreq-aperf {{[-o|--once]}}
```

cpufreq-info

Show CPU frequency information.

More information: https://manned.org/cpufreq-info.

- Show CPU frequency information for all CPUs:

```
cpufreq-info
```

- Show CPU frequency information for the specified CPU:

```
cpufreq-info {{[-c|--cpu]}} {{cpu_number}}
```

- Show the allowed minimum and maximum CPU frequency:

```
cpufreq-info {{[-l|--hwlimits]}}
```

- Show the current minimum and maximum CPU frequency and policy in table format:

```
cpufreq-info {{[-o|--proc]}}
```

- Show available CPU frequency policies:

```
cpufreq-info {{[-g|--governors]}}
```

- Show current CPU work frequency in a human-readable format, according to the cpufreq kernel module:

```
cpufreq-info {{[-f|--freq]}} {{[-m|--human]}}
```

- Show current CPU work frequency in a human-readable format, by reading it from hardware (only available to root):

```
sudo cpufreq-info {{[-w|--hwfreq]}} {{[-m|--human]}}
```

cpufreq-set

A tool to modify CPU frequency settings.

The frequency value should range between the output of command **`cpufreq-info -l`**.

More information: https://manned.org/cpufreq-set.

- Set the CPU frequency policy of CPU 1 to "userspace":

```
sudo cpufreq-set {{[-c|--cpu]}} {{1}} {{[-g|--governor]}} {{userspace}}
```

- Set the current minimum CPU frequency of CPU 1:

```
sudo cpufreq-set {{[-c|--cpu]}} {{1}} {{[-d|--min]}} {{min_frequency}}
```

- Set the current maximum CPU frequency of CPU 1:

```
sudo cpufreq-set {{[-c|--cpu]}} {{1}} {{[-u|--max]}} {{max_frequency}}
```

- Set the current work frequency of CPU 1:

```
sudo cpufreq-set {{[-c|--cpu]}} {{1}} {{[-f|--freq]}} {{work_frequency}}
```

cpuid

Display detailed information about all CPUs.

More information: https://manned.org/cpuid.1.

- Display information for all CPUs:

  ```
  cpuid
  ```

- Display information only for the current CPU:

  ```
  cpuid {{[-1|--one-cpu]}}
  ```

- Display raw hex information with no decoding:

  ```
  cpuid {{[-r|--raw]}}
  ```

cpulimit

A tool to throttle the CPU usage of other processes.

More information: https://manned.org/cpulimit.

- Limit an existing process with PID 1234 to only use 25% of the CPU:

```
cpulimit {{[-p|--pid]}} {{1234}} {{[-l|--limit]}} {{25%}}
```

- Limit an existing program by its executable name:

```
cpulimit {{[-e|--exe]}} {{program}} {{[-l|--limit]}} {{25}}
```

- Launch a given program and limit it to only use 50% of the CPU:

```
cpulimit {{[-l|--limit]}} {{50}} -- {{program argument1 argument2 ...}}
```

- Launch a program, limit its CPU usage to 50% and run cpulimit in the background:

```
cpulimit {{[-l|--limit]}} {{50}} {{[-b|--background]}} -- {{program}}
```

- Kill its process if the program's CPU usage goes over 50%:

```
cpulimit {{[-l|--limit]}} 50 {{[-k|--kill]}} -- {{program}}
```

- Throttle both it and its child processes so that none go about 25% CPU:

```
cpulimit {{[-l|--limit]}} {{25}} {{[-m|--monitor-forks]}} -- {{program}}
```

cpupower

Tools regarding CPU power and tuning options.

More information: https://manned.org/cpupower.

- List CPUs:

```
sudo cpupower {{[-c|--cpu]}} {{all}} info
```

- Print information about all cores:

```
sudo cpupower {{[-c|--cpu]}} {{all}} info
```

- Set all CPUs to a power-saving frequency governor:

```
sudo cpupower {{[-c|--cpu]}} {{all}} frequency-set --governor
{{powersave}}
```

- Print CPU 0's available frequency governors:

```
sudo cpupower {{[-c|--cpu]}} {{0}} frequency-info {{[-g|--
governors]}} | grep "analyzing\|governors"
```

- Print CPU 4's frequency from the hardware, in a human-readable format:

```
sudo cpupower {{[-c|--cpu]}} {{4}} frequency-info {{[-w|--
hwfreq]}} {{[-m|--human]}}
```

create-image

Create CPIO images.

More information: https://git.kernel.org/pub/scm/bluetooth/bluez.git/tree/tools/create-image.c.

- Create a CPIO archive from the current directory:

```
create-image {{[-o|--output]}} {{path/to/output.cpio}}
```

- Display help:

```
create-image {{[-h|--help]}}
```

create_ap

Create an AP (Access Point) at any channel.

More information: https://github.com/oblique/create_ap.

- Create an open network with no passphrase:

```
create_ap {{wlan0}} {{eth0}} {{access_point_ssid}}
```

- Use a WPA + WPA2 passphrase:

```
create_ap {{wlan0}} {{eth0}} {{access_point_ssid}}
{{passphrase}}
```

- Create an access point without Internet sharing:

```
create_ap -n {{wlan0}} {{access_point_ssid}} {{passphrase}}
```

- Create a bridged network with Internet sharing:

```
create_ap -m bridge {{wlan0}} {{eth0}} {{access_point_ssid}}
{{passphrase}}
```

- Create a bridged network with Internet sharing and a pre-configured bridge interface:

```
create_ap -m bridge {{wlan0}} {{br0}} {{access_point_ssid}}
{{passphrase}}
```

- Create an access port for Internet sharing from the same Wi-Fi interface:

```
create_ap {{wlan0}} {{wlan0}} {{access_point_ssid}}
{{passphrase}}
```

- Choose a different Wi-Fi adapter driver:

```
create_ap --driver {{wifi_adapter}} {{wlan0}} {{eth0}}
{{access_point_ssid}} {{passphrase}}
```

createrepo

Initialize an RPM repository in a directory, including all XML and SQLite files.

More information: https://manned.org/createrepo.

- Initialize a basic repository in a directory:

```
createrepo {{path/to/directory}}
```

- Initialize a repository, exclude test RPMs and display verbose logs:

```
createrepo {{[-v|--verbose]}} {{[-x|--excludes]}}
{{test_*.rpm}} {{path/to/directory}}
```

- Initialize a repository, using SHA1 as the checksum algorithm, and ignoring symbolic links:

```
createrepo {{[-S|--skip-symlinks]}} {{[-s|--checksum]}}
{{sha1}} {{path/to/directory}}
```

crond

Daemon to execute scheduled commands from crontab files.

More information: https://manned.org/crond.

- Start daemon in the background and check for scheduled commands:

```
crond
```

- Start daemon in the foreground and check for scheduled commands:

```
crond -n
```

- Send job output from the daemon to the [s]ystem log:

```
crond -s
```

- Override default limitations and accept custom crontables:

```
crond -p
```

- Inherit crontab file path from environment settings:

```
crond -P
```

cryptcat

Cryptcat is netcat with encryption capabilities.

More information: https://manned.org/cryptcat.

- [l]isten on a specified [p]ort and print any data received:

```
cryptcat -k {{password}} -l -p {{port}}
```

- Connect to a certain port:

```
cryptcat -k {{password}} {{ip_address}} {{port}}
```

- Specify the timeout ([w]):

```
cryptcat -k {{password}} -w {{timeout_in_seconds}}
{{ip_address}} {{port}}
```

- Scan ([z]) the open ports of a specified host:

```
cryptcat -v -z {{ip_address}} {{port}}
```

- Act as proxy and forward data from a local TCP port to the given remote host:

```
cryptcat -k {{password}} -l -p {{local_port}} | cryptcat -k
{{password}} {{hostname}} {{remote_port}}
```

cryptsetup luksFormat

Initialize a LUKS partition and the initial key slot (0) with a passphrase or keyfile.

Note: This operation overwrites all data on the partition.

More information: https://manned.org/cryptsetup-luksFormat.

- Initialize a LUKS volume with a passphrase:

```
cryptsetup luksFormat {{/dev/sdXY}}
```

- Initialize a LUKS volume with a keyfile:

```
cryptsetup luksFormat {{/dev/sdXY}} {{path/to/keyfile}}
```

- Initialize a LUKS volume with a passphrase and set its label:

```
cryptsetup luksFormat --label {{label}} {{/dev/sdXY}}
```

cryptsetup open

Create a decrypted mapping of an encrypted volume.

Note: With TRIM enabled, minimal data leakage in form of freed block information, perhaps sufficient to determine the filesystem in use may occur.

However, you still most likely want to enable it, because the data inside is still safe and SSDs without TRIM will wear out faster.

More information: https://manned.org/cryptsetup-open.

- Open a LUKS volume and create a decrypted mapping at `/dev/mapper/mapping_name`:

```
cryptsetup open {{/dev/sdXY}} {{mapping_name}}
```

- Use a keyfile instead of a passphrase:

```
cryptsetup open {{[-k|--key-file]}} {{path/to/file}} {{/dev/sdXY}} {{mapping_name}}
```

- Allow the use of TRIM on the device:

```
cryptsetup open --allow-discards {{/dev/sdXY}} {{mapping_name}}
```

- Write the `--allow-discards` option into the LUKS header (the option will then always be used when you open the device):

```
cryptsetup open --allow-discards --persistent {{/dev/sdXY}} {{mapping_name}}
```

- Open a LUKS volume and make the decrypted mapping read-only:

```
cryptsetup open {{[-r|--readonly]}} {{/dev/sdXY}} {{mapping_name}}
```

cryptsetup

Manage plain **dm-crypt** and LUKS (Linux Unified Key Setup) encrypted volumes.

Some subcommands such as **luksFormat** have their own usage documentation.

More information: https://manned.org/cryptsetup.

- Initialize a LUKS volume with a passphrase (overwrites all data on the partition):

```
cryptsetup luksFormat {{/dev/sdXY}}
```

- Open a LUKS volume and create a decrypted mapping at `/dev/mapper/mapping_name`:

```
cryptsetup open {{/dev/sdXY}} {{mapping_name}}
```

- Display information about a mapping:

```
cryptsetup status {{mapping_name}}
```

- Remove an existing mapping:

```
cryptsetup close {{mapping_name}}
```

- Change a LUKS volume's passphrase:

```
cryptsetup luksChangeKey {{/dev/sdXY}}
```

cs2

This command is an alias of **`counter strike 2`**.

- View documentation for the original command:

```
tldr counter strike 2
```

csplit

Split a file into pieces.

This generates files named "xx00", "xx01", and so on.

More information: https://www.gnu.org/software/coreutils/manual/html_node/csplit-invocation.html.

- Split a file at lines 5 and 23:

```
csplit {{path/to/file}} 5 23
```

- Split a file every 5 lines (this will fail if the total number of lines is not divisible by 5):

```
csplit {{path/to/file}} 5 {*}
```

- Split a file every 5 lines, ignoring exact-division error:

```
csplit {{[-k|--keep-files]}} {{path/to/file}} 5 {*}
```

- Split a file at line 5 and use a custom prefix for the output files:

```
csplit {{path/to/file}} 5 {{[-f|--prefix]}} {{prefix}}
```

- Split a file at a line matching a `regex`:

```
csplit {{path/to/file}} /{{regex}}/
```

ctop

Instantly visualize container performance and health with real-time metrics on CPU, memory, and block IO usage.

More information: https://github.com/bcicen/ctop#usage.

- Show only [a]ctive containers:

```
ctop -a
```

- [r]everse the container sort order:

```
ctop -r
```

- [i]nvert the default colors:

```
ctop -i
```

- Display [h]elp:

```
ctop -h
```

ctr

Manage **containerd** containers and images.

More information: https://containerd.io.

- List all containers (running and stopped):

```
ctr containers list
```

- List all images:

```
ctr images list
```

- Pull an image:

```
ctr images pull {{image}}
```

- Tag an image:

```
ctr images tag {{source_image}}:{{source_tag}}
{{target_image}}:{{target_tag}}
```

ctrlaltdel

Utility to control what happens when CTRL+ALT+DEL is pressed.

More information: https://manned.org/ctrlaltdel.

- Get current setting:

```
ctrlaltdel
```

- Set CTRL+ALT+DEL to reboot immediately, without any preparation:

```
sudo ctrlaltdel hard
```

- Set CTRL+ALT+DEL to reboot "normally", giving processes a chance to exit first (send SIGINT to PID1):

```
sudo ctrlaltdel soft
```

cu

Call Up another system and act as a dial-in/serial terminal or perform file transfers with no error checking.

More information: https://manned.org/cu.

- Open a given serial port:

```
sudo cu {{[-l|--line]}} {{/dev/ttyXYZ}}
```

- Open a given serial port with a given baud rate:

```
sudo cu {{[-l|--line]}} {{/dev/ttyXYZ}} {{[-s|--speed]}} {{115200}}
```

- Open a given serial port with a given baud rate and echo characters locally (half-duplex mode):

```
sudo cu {{[-l|--line]}} {{/dev/ttyXYZ}} {{[-s|--speed]}} {{115200}} {{[-h|--halfduplex]}}
```

- Open a given serial port with a given baud rate, parity, and no hardware or software flow control:

```
sudo cu {{[-l|--line]}} {{/dev/ttyXYZ}} {{[-s|--speed]}} {{115200}} --parity={{even|odd|none}} {{[-f|--nortscts]}} --nostop
```

- Exit the `cu` session when in connection:

```
<Enter><~><.>
```

- Display help:

```
cu --help
```

curlie

A frontend to **`curl`** that adds the ease of use of **`httpie`**.

More information: https://github.com/rs/curlie.

- Send a GET request:

```
curlie {{httpbin.org/get}}
```

- Send a POST request:

```
curlie post {{httpbin.org/post}} {{name=john}} {{age:=25}}
```

- Send a GET request with query parameters (e.g. `first_param=5&second_param=true`):

```
curlie get {{httpbin.org/get}} {{first_param==5}}
{{second_param==true}}
```

- Send a GET request with a custom header:

```
curlie get {{httpbin.org/get}} {{header-name:header-value}}
```

cuyo

Tetris like game.

More information: https://www.karimmi.de/cuyo/.

- Start a new game:

  ```
  cuyo
  ```

- Navigate the piece horizontally:

  ```
  {{<a>|<d>|<ArrowLeft>|<ArrowRight>}}
  ```

- Turn the piece:

  ```
  {{<w>|<ArrowUp>}}
  ```

- Hard drop the piece:

  ```
  {{<s>|<ArrowDown>}}
  ```

daemon

Turns other processes into daemons.

More information: https://manned.org/daemon.1.

- Run a command as a daemon:

```
daemon --name="{{name}}" {{command}}
```

- Run a command as a daemon which will restart if the command crashes:

```
daemon --name="{{name}}" {{[-r|--respawn]}} {{command}}
```

- Run a command as a daemon which will restart if it crashes, with two attempts every 10 seconds:

```
daemon --name="{{name}}" {{[-r|--respawn]}} --attempts=2 --delay=10 {{command}}
```

- Run a command as a daemon, writing logs to a specific file:

```
daemon --name="{{name}}" --errlog={{path/to/file.log}} {{command}}
```

- Kill a daemon (SIGTERM):

```
daemon --name="{{name}}" --stop
```

- List daemons:

```
daemon --list
```

daemonize

Run a command (that does not daemonize itself) as a Unix daemon.

More information: https://software.clapper.org/daemonize/.

- Run a command as a daemon:

```
daemonize {{command}} {{command_arguments}}
```

- Write the PID to the specified file:

```
daemonize -p {{path/to/pidfile}} {{command}}
{{command_arguments}}
```

- Use a lock file to ensure that only one instance runs at a time:

```
daemonize -l {{path/to/lockfile}} {{command}}
{{command_arguments}}
```

- Use the specified user account:

```
sudo daemonize -u {{user}} {{command}} {{command_arguments}}
```

darling

Run macOS software on Linux.

More information: https://darlinghq.org.

- Run a builtin command:

```
darling shell {{uname}}
```

- Run a specific program with arguments:

```
darling shell {{path/to/program}} {{program_argument_1 program_argument_2 ...}}
```

- Open a macOS shell:

```
darling shell
```

- Shutdown the service:

```
darling shutdown
```

datamash

Perform basic numeric, textual and statistical operations on input textual data files.

More information: https://www.gnu.org/software/datamash/manual/datamash.html#Invoking-datamash.

- Get max, min, mean and median of a single column of numbers:

```
seq 3 | datamash max 1 min 1 mean 1 median 1
```

- Get the mean of a single column of float numbers (floats must use "," and not "."):

```
echo -e '1.0\n2.5\n3.1\n4.3\n5.6\n5.7' | tr '.' ',' |
datamash mean 1
```

- Get the mean of a single column of numbers with a given decimal precision:

```
echo -e '1\n2\n3\n4\n5\n5' | datamash {{[-R|--round]}}
{{number_of_decimals_wanted}} mean 1
```

- Get the mean of a single column of numbers ignoring "Na" and "NaN" (literal) strings:

```
echo -e '1\n2\nNa\n3\nNaN' | datamash --narm mean 1
```

dbclient

Lightweight Dropbear Secure Shell client.

More information: https://manned.org/dbclient.

- Connect to a remote host:

```
dbclient {{user}}@{{host}}
```

- Connect to a remote host on [p]ort 2222:

```
dbclient {{user}}@{{host}} -p 2222
```

- Connect to a remote host using a specific [i]dentity key in dropbear format:

```
dbclient -i {{path/to/key_file}} {{user}}@{{host}}
```

- Run a command on the remote host with a [t]ty allocation allowing interaction with the remote command:

```
dbclient {{user}}@{{host}} -t {{command}} {{argument1 argument2 ...}}
```

- Connect and forward [A]gent connections to remote host:

```
dbclient -A {{user}}@{{host}}
```

dbus-daemon

The D-Bus message daemon, allowing multiple programs to exchange messages.

More information: https://dbus.freedesktop.org/doc/dbus-daemon.1.html.

- Run the daemon with a configuration file:

```
dbus-daemon --config-file {{path/to/file}}
```

- Run the daemon with the standard per-login-session message bus configuration:

```
dbus-daemon --session
```

- Run the daemon with the standard systemwide message bus configuration:

```
dbus-daemon --system
```

- Set the address to listen on and override the configuration value for it:

```
dbus-daemon --address {{address}}
```

- Output the process ID to `stdout`:

```
dbus-daemon --print-pid
```

- Force the message bus to write to the system log for messages:

```
dbus-daemon --syslog
```

dconf read

Read key values from dconf databases.

See also: **`dconf`**.

More information: https://manned.org/dconf.

- Print a specific key value:

```
dconf read /{{path/to/key}}
```

- Print a specific key [d]efault value:

```
dconf read -d /{{path/to/key}}
```

dconf reset

Reset key values in dconf databases.

See also: **`dconf`**.

More information: https://manned.org/dconf.

- Reset a specific key value:

```
dconf reset /{{path/to/key}}
```

- Reset a specific directory:

```
dconf reset -f /{{path/to/directory}}/
```

dconf write

Write key values in dconf databases.

See also: **`dconf`**.

More information: https://manned.org/dconf.

- Write a specific key value:

```
dconf write /{{path/to/key}} "{{value}}"
```

- Write a specific string key value:

```
dconf write /{{path/to/key}} "'{{string}}'"
```

- Write a specific integer key value:

```
dconf write /{{path/to/key}} "{{5}}"
```

- Write a specific boolean key value:

```
dconf write /{{path/to/key}} "{{true|false}}"
```

- Write a specific array key value:

```
dconf write /{{path/to/key}} "[{{'first', 'second', ...}}]"
```

- Write a specific empty array key value:

```
dconf write /{{path/to/key}} "@as []"
```

dconf

Manage dconf databases.

See also: **`dconf-read`**, **`dconf-reset`**, **`dconf-write`**, **`gsettings`**.

More information: https://manned.org/dconf.

- Print a specific key value:

```
dconf read /{{path/to/key}}
```

- Print a specific path sub-directories and sub-keys:

```
dconf list /{{path/to/directory}}/
```

- Write a specific key value:

```
dconf write /{{path/to/key}} "{{value}}"
```

- Reset a specific key value:

```
dconf reset /{{path/to/key}}
```

- Watch a specific key/directory for changes:

```
dconf watch /{{path/to/key|/path/to/directory}}/
```

- Dump a specific directory in INI file format:

```
dconf dump /{{path/to/directory}}/
```

dd

Convert and copy a file.

See also: **`caligula`**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/dd-invocation.html.

- Make a bootable USB drive from an isohybrid file (such as `archlinux-xxx.iso`) and show the progress:

```
sudo dd if={{path/to/file.iso}} of={{/dev/usb_drive}} status=progress
```

- Clone a drive to another drive with 4 MiB block size and flush writes before the command terminates:

```
sudo dd bs=4M conv=fsync if={{/dev/source_drive}} of={{/dev/dest_drive}}
```

- Generate a file with a specific number of random bytes by using kernel random driver:

```
dd bs={{100}} count={{1}} if=/dev/urandom of={{path/to/random_file}}
```

- Benchmark the write performance of a disk:

```
dd bs={{1M}} count={{1024}} if=/dev/zero of={{path/to/file_1GB}}
```

- Create a system backup, save it into an IMG file (can be restored later by swapping `if` and `of`), and show the progress:

```
dd if={{/dev/drive_device}} of={{path/to/file.img}} status=progress
```

- Check the progress of an ongoing `dd` operation (run this command from another shell):

```
progress
```

ddcutil

Control the settings of connected displays via DDC/CI.

This command requires the kernel module **`i2c-dev`** to be loaded.

See also: **`modprobe`**.

More information: https://www.ddcutil.com/commands/.

- List all compatible displays:

```
ddcutil detect
```

- Change the brightness (option 0x10) of display 1 to 50%:

```
ddcutil {{[-d|--display]}} {{1}} setvcp {{10}} {{50}}
```

- Increase the contrast (option 0x12) of display 1 by 5%:

```
ddcutil {{[-d|--display]}} {{1}} setvcp {{12}} {{+}} {{5}}
```

- Read the settings of display 1:

```
ddcutil {{[-d|--display]}} {{1}} getvcp {{ALL}}
```

ddrescue

Data recovery tool that reads data from damaged block devices.

More information: https://www.gnu.org/software/ddrescue/manual/ddrescue_manual.html#Invoking-ddrescue.

- Take an image of a device, creating a log file:

```
sudo ddrescue {{/dev/sdb}} {{path/to/image.dd}} {{path/to/log.txt}}
```

- Clone Disk A to Disk B, creating a log file:

```
sudo ddrescue {{[-f|--force]}} {{[-n|--no-scrape]}} {{/dev/sdX}} {{/dev/sdY}} {{path/to/log.txt}}
```

deb-get

`apt-get` functionality for **`.deb`** packages published in third party repositories or via direct download.

Works with Linux distributions which use **`apt-get`**.

More information: https://github.com/wimpysworld/deb-get.

- Update the list of available packages and versions:

```
deb-get update
```

- Search for a given package:

```
deb-get search {{package}}
```

- Show information about a package:

```
deb-get show {{package}}
```

- Install a package, or update it to the latest available version:

```
deb-get install {{package}}
```

- Remove a package (using `purge` instead also removes its configuration files):

```
deb-get remove {{package}}
```

- Upgrade all installed packages to their newest available versions:

```
deb-get upgrade
```

- List all available packages:

```
deb-get list
```

debchange

Maintain the debian/changelog file of a Debian source package.

More information: https://manned.org/debchange.1.

- Add a new version for a non-maintainer upload to the changelog:

```
debchange {{[-n|--nmu]}}
```

- Add a changelog entry to the current version:

```
debchange {{[-a|--append]}}
```

- Add a changelog entry to close the bug with specified ID:

```
debchange --closes {{bug_id}}
```

debman

Read man pages from uninstalled packages.

More information: https://manned.org/debman.1.

- Read a man page for a command that is provided by a specified [p]ackage:

```
debman -p {{package}} {{command}}
```

- Specify a [p]ackage version to download:

```
debman -p {{package}}={{version}} {{command}}
```

- Read a man page in a `.deb` [f]ile:

```
debman -f {{path/to/file.deb}} {{command}}
```

debootstrap

Create a basic Debian system.

More information: https://wiki.debian.org/Debootstrap.

- Create a Debian stable release system inside the `debian-root` directory:

```
sudo debootstrap stable {{path/to/debian-root}}/ http://deb.debian.org/debian
```

- Create a minimal system including only required packages:

```
sudo debootstrap --variant=minbase stable {{path/to/debian-root}}/
```

- Create a Debian Unstable system inside the `sid-root` directory with a local mirror:

```
sudo debootstrap sid {{path/to/sid-root}}/ {{file:///path/to/mirror}}/
```

- Switch to a bootstrapped system:

```
sudo chroot {{path/to/root}}
```

- List available releases:

```
ls /usr/share/debootstrap/scripts/
```

deborphan

Display orphan packages on operating systems using the APT package manager.

More information: https://manned.org/deborphan.

- Display library packages (from the "libs" section of the package repository) which are not required by another package:

```
deborphan
```

- List orphan packages from the "libs" section as well as orphan packages that have a name that looks like a library name:

```
deborphan --guess-all
```

- Find packages which are only recommended or suggested (but not required) by another package:

```
deborphan {{[-n|--nice-mode]}}
```

debsecan

Debian Security Analyzer, a tool to list vulnerabilities on a particular Debian installation.

More information: https://manned.org/debsecan.

- List vulnerable installed packages on the current host:

```
debsecan
```

- List vulnerable installed packages of a specific suite:

```
debsecan --suite {{release_code_name}}
```

- List only fixed vulnerabilities:

```
debsecan --suite {{release_code_name}} --only-fixed
```

- List only fixed vulnerabilities of unstable ("sid") and mail to root:

```
debsecan --suite {{sid}} --only-fixed --format {{report}} --
mailto {{root}} --update-history
```

- Upgrade vulnerable installed packages:

```
sudo apt upgrade $(debsecan --only-fixed --format
{{packages}})
```

debtap

Convert Debian packages into Arch Linux packages.

See also: **`pacman-upgrade`**.

More information: https://github.com/helixarch/debtap.

- Update debtap database (before the first run):

```
sudo debtap {{[-u|--update]}}
```

- Convert the specified package:

```
debtap {{path/to/package.deb}}
```

- Convert the specified package bypassing all questions, except for editing metadata files:

```
debtap {{[-q|--quiet]}} {{path/to/package.deb}}
```

- Generate a PKGBUILD file:

```
debtap {{[-p|--pkgbuild]}} {{path/to/package.deb}}
```

debugfs

An interactive ext2/ext3/ext4 filesystem debugger.

More information: https://manned.org/debugfs.

- Open the filesystem in read only mode:

```
debugfs {{/dev/sdXN}}
```

- Open the filesystem in read write mode:

```
debugfs -w {{/dev/sdXN}}
```

- Read commands from a specified file, execute them and then exit:

```
debugfs -f {{path/to/cmd_file}} {{/dev/sdXN}}
```

- View the filesystem stats in debugfs console:

```
stats
```

- Close the filesystem:

```
close -a
```

- List all available commands:

```
lr
```

debuginfod-find

Request debuginfo-related data.

More information: https://manned.org/debuginfod-find.

- Request data based on the `build_id`:

```
debuginfod-find -vv debuginfo {{build_id}}
```

debuild

Build a Debian package from source.

More information: https://manned.org/debuild.1.

- Build the package in the current directory:

```
debuild
```

- Build a binary package only:

```
debuild -b
```

- Do not run lintian after building the package:

```
debuild --no-lintian
```

decode-dimms

Decode RAM data.

More information: https://manned.org/decode-dimms.

- Display DIMM information:

```
decode-dimms
```

- Display help:

```
decode-dimms {{[-h|--help]}}
```

delpart

Ask the Linux kernel to forget about a partition.

More information: https://manned.org/delpart.

- Tell the kernel to forget about the first partition of `/dev/sda`:

```
sudo delpart {{/dev/sda}} {{1}}
```

deluser

Delete a user from the system.

More information: https://manned.org/deluser.

- Remove a user:

```
sudo deluser {{username}}
```

- Remove a user and their home directory:

```
sudo deluser --remove-home {{username}}
```

- Remove a user and their home, but backup their files into a `.tar.gz` file in the specified directory:

```
sudo deluser --backup-to {{path/to/backup_directory}} --remove-home {{username}}
```

- Remove a user, and all files owned by them:

```
sudo deluser --remove-all-files {{username}}
```

dex

DesktopEntry Execution is a program to generate and execute DesktopEntry files of the Application type.

More information: https://github.com/jceb/dex.

- Execute all programs in the autostart folders:

```
dex {{[-a|--autostart]}}
```

- Execute all programs in the specified folders:

```
dex {{[-a|--autostart]}} {{[-s|--search-paths]}} {{path/to/
directory1}}:{{path/to/directory2}}:{{path/to/directory3}}:
```

- Preview the programs would be executed in a GNOME specific autostart:

```
dex {{[-a|--autostart]}} {{[-e|--environment]}} {{GNOME}}
```

- Preview the programs would be executed in a regular autostart:

```
dex {{[-a|--autostart]}} {{[-d|--dry-run]}}
```

- Preview the value of the DesktopEntry property `Name`:

```
dex {{[-p|--property]}} {{Name}} {{path/to/file.desktop}}
```

- Create a DesktopEntry for a program in the current directory:

```
dex {{[-c|--create]}} {{path/to/file.desktop}}
```

- Execute a single program (with `Terminal=true` in the desktop file) in the given terminal:

```
dex --term {{terminal}} {{path/to/file.desktop}}
```

df

Display an overview of the filesystem disk space usage.

More information: https://www.gnu.org/software/coreutils/manual/html_node/df-invocation.html.

- Display all filesystems and their disk usage:

```
df
```

- Display all filesystems in human-readable form:

```
df {{[-h|--human-readable]}}
```

- Display the filesystem containing the specified file or directory:

```
df {{path/to/file_or_directory}}
```

- Include statistics on the number of free inodes:

```
df {{[-i|--inodes]}}
```

- Display filesystems but exclude the specified types:

```
df {{[-x|--exclude-type]}} {{squashfs}} {{[-x|--exclude-type]}} {{tmpfs}}
```

- Display filesystem types:

```
df {{[-T|--print-type]}}
```

dget

Download Debian packages.

More information: https://manned.org/dget.1.

- Download a binary package:

```
dget {{package}}
```

- Download and extract a package source from its `.dsc` file:

```
dget {{http://deb.debian.org/debian/pool/main/h/haskell-tldr/haskell-tldr_0.4.0-2.dsc}}
```

- Download a package source tarball from its `.dsc` file but don't extract it:

```
dget {{[-d|--download-only]}} {{http://deb.debian.org/debian/pool/main/h/haskell-tldr/haskell-tldr_0.4.0-2.dsc}}
```

dhcpcd

DHCP client.

More information: https://roy.marples.name/projects/dhcpcd.

- Release all address leases:

```
sudo dhcpcd {{[-k|--release]}}
```

- Request the DHCP server for new leases:

```
sudo dhcpcd {{[-n|--rebind]}}
```

dialog

Display dialog boxes on the terminal.

See also: **`gum`**, **`whiptail`**.

More information: https://manned.org/dialog.

- Display a message:

```
dialog --msgbox "{{Message}}" {{height}} {{width}}
```

- Prompt the user for text:

```
dialog --inputbox "{{Enter text:}}" {{8}} {{40}}
2>{{output.txt}}
```

- Prompt the user for a yes/no question:

```
dialog --yesno "{{Continue?}}" {{7}} {{40}}
```

- Display help:

```
dialog
```

diff3

Compare three files line by line.

More information: https://www.gnu.org/software/diffutils/manual/diffutils.html#Invoking-diff3.

- Compare files:

```
diff3 {{path/to/file1}} {{path/to/file2}} {{path/to/file3}}
```

- Show all changes, outlining conflicts:

```
diff3 {{[-A|--show-all]}} {{path/to/file1}} {{path/to/file2}} {{path/to/file3}}
```

diffimg

Calculate intersection between two images.

Note: The supported extensions are **`.png`**, **`.gif`**, **`.jpg`**, **`.ps`**.

More information: https://manned.org/diffimg.

- Calculate the intersection between images and output an image where each pixel is the difference between corresponding pixels in input images:

```
diffimg {{path/to/input_image1.ext}} {{path/to/
input_image2.ext}} {{path/to/output_image.ext}}
```

dir

List directory contents using one line per file, special characters are represented by backslash escape sequences.

Works as **`ls -C --escape`**.

More information: https://manned.org/dir.

- List all files, including hidden files:

```
dir {{[-a|--all]}}
```

- List files including their author (`-l` is required):

```
dir -l --author
```

- List files excluding those that match a specified blob pattern:

```
dir --hide {{pattern}}
```

- List subdirectories recursively:

```
dir {{[-R|--recursive]}}
```

- Display help:

```
dir --help
```

dirb

Scan HTTP-based webservers for directories and files.

More information: https://manned.org/dirb.

- Scan a webserver using the default wordlist:

```
dirb {{https://example.org}}
```

- Scan a webserver using a custom wordlist:

```
dirb {{https://example.org}} {{path/to/wordlist.txt}}
```

- Scan a webserver non-recursively:

```
dirb {{https://example.org}} -r
```

- Scan a webserver using a specified user-agent and cookie for HTTP-requests:

```
dirb {{https://example.org}} -a {{user_agent_string}} -c
{{cookie_string}}
```

dirbuster

Brute force directories and filenames on servers.

More information: https://www.kali.org/tools/dirbuster/.

- Start in GUI mode:

```
dirbuster -u {{http://example.com}}
```

- Start in headless (no GUI) mode:

```
dirbuster -H -u {{http://example.com}}
```

- Set the file extension list:

```
dirbuster -e {{txt,html}}
```

- Enable verbose output:

```
dirbuster -v
```

- Set the report location:

```
dirbuster -r {{path/to/report.txt}}
```

distrobox-create

Create a Distrobox container.

The container created will be tightly integrated with the host, allowing sharing of the user's HOME directory, external storage, external USB devices, graphical apps (X11/Wayland), and audio.

See also: **`distrobox`**.

More information: https://distrobox.it/usage/distrobox-create.

- Create a Distrobox container using the Ubuntu image:

```
distrobox-create {{container_name}} {{[-i|--image]}} {{ubuntu:latest}}
```

- Clone a Distrobox container:

```
distrobox-create {{[-c|--clone]}} {{container_name}} {{cloned_container_name}}
```

distrobox-enter

Enter a Distrobox container.

Default command executed is your SHELL, but you can specify different shells or entire commands to execute. If used inside a script, an application, or a service, you can use the **`--headless`** mode to disable the tty and interactivity.

See also: **`distrobox`**.

More information: https://distrobox.it/usage/distrobox-enter.

- Enter a Distrobox container:

```
distrobox-enter {{container_name}}
```

- Enter a Distrobox container and run a command at login:

```
distrobox-enter {{container_name}} -- {{sh -l}}
```

- Enter a Distrobox container without instantiating a tty:

```
distrobox-enter {{[-n|--name]}} {{container_name}} --
{{uptime --pretty}}
```

distrobox-export

Export app/service/binary from container to host OS.

See also: **`distrobox`**.

More information: https://distrobox.it/usage/distrobox-export.

- Export an app from the container to the host (the desktop entry/icon will show up in your host system's application list):

```
distrobox-export {{[-a|--app]}} {{package}} {{[-ef|--extra-flags]}} "--foreground"
```

- Export a binary from the container to the host:

```
distrobox-export {{[-b|--bin]}} {{path/to/binary}} {{[-ep|--export-path]}} {{path/to/binary_on_host}}
```

- Export a binary from the container to the host (i.e.`$HOME/.local/bin`):

```
distrobox-export {{[-b|--bin]}} {{path/to/binary}} {{[-ep|--export-path]}} {{path/to/export}}
```

- Export a service from the container to the host (`--sudo` will run the service as root inside the container):

```
distrobox-export --service {{package}} {{[-ef|--extra-flags]}} "--allow-newer-config" {{[-S|--sudo]}}
```

- Unexport/delete an exported application:

```
distrobox-export {{[-a|--app]}} {{package}} {{[-d|--delete]}}
```

distrobox-host-exec

Execute a command on the host from inside a Distrobox container.

See also: **`distrobox`**.

More information: https://distrobox.it/usage/distrobox-host-exec.

- Execute command on the host system from inside the Distrobox container:

```
distrobox-host-exec "{{command}}"
```

- Execute the `ls` command on the host system from inside the container:

```
distrobox-host-exec ls
```

distrobox-list

List all Distrobox containers.

Distrobox containers are listed separately from the rest of normal Podman or Docker containers.

See also: **`distrobox`**.

More information: https://distrobox.it/usage/distrobox-list.

- List all Distrobox containers:

```
distrobox-list
```

- List all Distrobox containers with verbose information:

```
distrobox-list {{[-v|--verbose]}}
```

distrobox-rm

Remove a Distrobox container.

See also: **`distrobox`**.

More information: https://distrobox.it/usage/distrobox-rm.

- Remove a Distrobox container (Tip: Stop the container before removing it):

```
distrobox-rm {{container_name}}
```

- Remove a Distrobox container forcefully:

```
distrobox-rm {{container_name}} {{[-f|--force]}}
```

distrobox-stop

Stop a Distrobox container.

See also: **`distrobox`**.

More information: https://distrobox.it/usage/distrobox-stop.

- Stop a Distrobox container:

```
distrobox-stop {{container_name}}
```

- Stop a Distrobox container non-interactively (without confirmation):

```
distrobox-stop {{[-n|--name]}} {{container_name}} {{[-Y|--yes]}}
```

distrobox-upgrade

Upgrade one or multiple Distrobox containers.

See also: **`distrobox`**.

More information: https://distrobox.it/usage/distrobox-upgrade.

- Upgrade a container using the container's native package manager:

```
distrobox-upgrade {{container_name}}
```

- Upgrade all containers using the container's native package managers:

```
distrobox-upgrade {{[-a|--all]}}
```

- Upgrade specific containers via the container's native package manager:

```
distrobox-upgrade {{container1 container2 ...}}
```

distrobox

Use any Linux distribution inside your terminal in a container. Install & use packages inside it while tightly integrating with the host OS, sharing storage (**home** directory) and hardware.

Note: It uses Podman or Docker to create your containers.

More information: https://github.com/89luca89/distrobox.

- View documentation for creating containers:

  `tldr distrobox-create`

- View documentation for listing container's information:

  `tldr distrobox-list`

- View documentation for entering the container:

  `tldr distrobox-enter`

- View documentation for executing a command on the host from inside a container:

  `tldr distrobox-host-exec`

- View documentation for exporting app/service/binary from the container to the host:

  `tldr distrobox-export`

- View documentation for upgrading containers:

  `tldr distrobox-upgrade`

- View documentation for stopping the containers:

  `tldr distrobox-stop`

- View documentation for removing the containers:

  `tldr distrobox-rm`

dkms

A framework that allows for dynamic building of kernel modules.

More information: https://manned.org/dkms.

- List currently installed modules:

```
dkms status
```

- Rebuild all modules for the currently running kernel:

```
sudo dkms autoinstall
```

- Install version 1.2.1 of the acpi_call module for the currently running kernel:

```
sudo dkms install -m {{acpi_call}} -v {{1.2.1}}
```

- Remove version 1.2.1 of the acpi_call module from all kernels:

```
sudo dkms remove -m {{acpi_call}} -v {{1.2.1}} --all
```

dm-tool

A tool to communicate with the display manager.

More information: https://manned.org/dm-tool.

- Show the greeter while keeping current desktop session open and waiting to be restored upon authentication by logged in user:

```
dm-tool switch-to-greeter
```

- Lock the current session:

```
dm-tool lock
```

- Switch to a specific user, showing an authentication prompt if required:

```
dm-tool switch-to-user {{username}} {{session}}
```

- Add a dynamic seat from within a running LightDM session:

```
dm-tool add-seat {{xlocal}} {{name}}={{value}}
```

dmenu

Dynamic menu.

Create a menu from a text input with each item on a new line.

More information: https://manned.org/dmenu.

- Display a menu of the output of the `ls` command:

```
{{ls}} | dmenu
```

- Display a menu with custom items separated by a new line (`\n`):

```
echo -e "{{red}}\n{{green}}\n{{blue}}" | dmenu
```

- Let the user choose between multiple items and save the selected one to a file:

```
echo -e "{{red}}\n{{green}}\n{{blue}}" | dmenu >
{{color.txt}}
```

- Launch dmenu on a specific monitor:

```
ls | dmenu -m {{1}}
```

- Display dmenu at the bottom of the screen:

```
ls | dmenu -b
```

dmesg

Write the kernel messages to **`stdout`**.

See also: **`journalctl`**.

More information: https://manned.org/dmesg.

- Show kernel messages:

```
sudo dmesg
```

- Show kernel error messages:

```
sudo dmesg {{[-l|--level]}} err
```

- Show kernel messages and keep [w]aiting for new ones, similar to `tail --follow` (available in kernels 3.5.0 and newer):

```
sudo dmesg {{[-w|--follow]}}
```

- Show how much physical memory is available on this system:

```
sudo dmesg | grep {{[-i|--ignore-case]}} memory
```

- Show kernel messages 1 page at a time:

```
sudo dmesg | less
```

- Show kernel messages with a timestamp (available in kernels 3.5.0 and newer):

```
sudo dmesg {{[-T|--ctime]}}
```

- Show kernel messages in human-readable form (available in kernels 3.5.0 and newer):

```
sudo dmesg {{[-H|--human]}}
```

- Colorize output (available in kernels 3.5.0 and newer):

```
sudo dmesg {{[-L|--color]}}
```

dmidecode

Display the DMI (alternatively known as SMBIOS) table contents in a human-readable format.

Requires root privileges.

More information: https://manned.org/dmidecode.

- Show all DMI table contents:

```
sudo dmidecode
```

- Show the BIOS version:

```
sudo dmidecode {{[-s|--string]}} bios-version
```

- Show the system's serial number:

```
sudo dmidecode {{[-s|--string]}} system-serial-number
```

- Show BIOS information:

```
sudo dmidecode {{[-t|--type]}} bios
```

- Show CPU information:

```
sudo dmidecode {{[-t|--type]}} processor
```

- Show memory information:

```
sudo dmidecode {{[-t|--type]}} memory
```

dnf builddep

Install dependencies to build a given package.

Not default to **`dnf`** but supported via **`dnf-plugins-core`**.

See also: **`dnf`**.

More information: https://dnf-plugins-core.readthedocs.io/en/latest/builddep.html.

- Install dependencies for a given package:

```
dnf builddep {{path/to/specification.spec}}
```

- Install dependencies for a given package but ignore unavailable:

```
dnf builddep --skip-unavailable {{path/to/
specification.spec}}
```

- Define the RPM macro to a given expression:

```
dnf builddep {{[-D|--define]}} '{{expression}}'
```

- Define an argument for a `.spec` file path:

```
dnf builddep --spec {{argument}}
```

- Define an argument for a `.rpm` file path:

```
dnf builddep --srpm {{argument}}
```

- Display help:

```
dnf builddep --help-cmd
```

dnf changelog

View the changelogs for a given package.

Not default to **`dnf`** but supported via **`dnf-plugins-core`**.

See also: **`dnf`**.

More information: https://dnf-plugins-core.readthedocs.io/en/latest/changelog.html.

- View all changelogs for a given package:

```
dnf changelog {{package}}
```

- View all changelogs for a given package after a specified date:

```
dnf changelog --since {{date}} {{package}}
```

- View the last n number of changelogs for a given package:

```
dnf changelog --count {{number}} {{package}}
```

- Show only new items for upgradeable packages:

```
dnf changelog --upgrades {{package}}
```

- Display help:

```
dnf changelog --help-cmd
```

dnf clean

Perform cleanup of temporary files kept for Red Hat-based repositories.

More information: https://dnf.readthedocs.io/en/latest/command_ref.html#clean-command.

- Remove cache files generated from the repository metadata:

```
dnf clean dbcache
```

- Mark the repository metadata as expired:

```
dnf clean expire-cache
```

- Remove repository metadata:

```
dnf clean metadata
```

- Remove any cached packages from the system:

```
dnf clean packages
```

- Clean up all DNF repository specific metadata and cached files (all of the above):

```
dnf clean all
```

dnf config-manager

Manage DNF configuration options and repositories on Fedora-based systems.

Not default to **`dnf`** but supported via **`dnf-plugins-core`**.

See also: **`dnf`**.

More information: https://dnf-plugins-core.readthedocs.io/en/latest/config_manager.html.

- Add (and enable) a repository from a URL:

```
dnf config-manager --add-repo={{repository_url}}
```

- Print current configuration values:

```
dnf config-manager --dump
```

- Enable a specific repository:

```
dnf config-manager {{[--enable|--set-enabled]}} {{repository_id}}
```

- Disable specified repositories:

```
dnf config-manager {{[--disable|--set-disabled]}} {{repository_id1 repository_id2 ...}}
```

- Set a configuration option for a repository:

```
dnf config-manager --setopt={{option}}={{value}}
```

- Display help:

```
dnf config-manager --help-cmd
```

dnf deplist

This command is an alias of **`dnf repoquery --deplist`**.

- View documentation for the original command:

  ```
  tldr dnf repoquery
  ```

dnf download

Download RPM packages from the DNF repositories.

Not default to **`dnf`** but supported via **`dnf-plugins-core`**.

See also: **`dnf`**.

More information: https://dnf-plugins-core.readthedocs.io/en/latest/download.html.

- Download the latest version of a package to the current directory:

```
dnf download {{package}}
```

- Download a package to a specific directory (the directory must exist):

```
dnf download {{package}} --destdir {{path/to/directory}}
```

- Print the URL where the RPM package can be downloaded from:

```
dnf download --url {{package}}
```

dnf group

Manage virtual collections of packages on Fedora-based systems.

More information: https://dnf.readthedocs.io/en/latest/command_ref.html#group-command.

- List DNF groups, showing installed and uninstalled status in a table:

```
dnf {{[grp|group]}} list
```

- Show DNF group info, including repository and optional packages:

```
dnf {{[grp|group]}} info {{group_name}}
```

- Install DNF group:

```
dnf {{[grp|group]}} install {{group_name}}
```

- Remove DNF group:

```
dnf {{[grp|group]}} remove {{group_name}}
```

- Upgrade DNF group:

```
dnf {{[grp|group]}} upgrade {{group_name}}
```

dnf install

Install packages on Red Hat-based distributions.

More information: https://dnf.readthedocs.io/en/latest/command_ref.html#install-examples.

- Install packages by name:

```
sudo dnf {{[in|install]}} {{package1 package2 ...}}
```

- Install a package from a local file:

```
sudo dnf {{[in|install]}} {{path/to/file}}
```

- Install a package from the internet:

```
sudo dnf {{[in|install]}} {{https://example.com/package.rpm}}
```

- Add the Extra Packages for Enterprise Linux (EPEL) repositories:

```
sudo dnf {{[in|install]}} https://dl.fedoraproject.org/pub/epel/epel-release-latest-{{10}}.noarch.rpm
```

- Add Remi's RPM repository:

```
sudo dnf {{[in|install]}} https://rpms.remirepo.net/enterprise/remi-release-{{8}}.rpm
```

dnf module

Manage package modularity.

More information: https://dnf.readthedocs.io/en/latest/command_ref.html#module-command.

- View the modularity overview:

```
dnf module list
```

- View modularity of a specific program:

```
dnf module list {{package_name}}
```

- Set a package to be enabled:

```
sudo dnf module enable {{package_name}}:{{stream}}
```

- Enable and install a specific version:

```
dnf module install {{package_name}}:{{stream}}
```

dnf repoquery

Query packages for information.

More information: https://dnf.readthedocs.io/en/latest/command_ref.html#repoquery-command.

- Query a package for its dependencies:

```
dnf {{[rq|repoquery]}} --deplist {{package}}
```

dnf reposync

Synchronize packages and metadata of a remote DNF repository to a local directory.

Not default to **`dnf`** but supported via **`dnf-plugins-core`**.

See also: **`dnf`**.

More information: https://dnf-plugins-core.readthedocs.io/en/latest/reposync.html.

- Synchronize all packages from the repository with id `repo_name` to a subdirectory `repo_name` of the current working directory:

```
dnf reposync --repoid {{repo_name}}
```

- Synchronize all packages and define a custom save location:

```
dnf reposync --repoid {{repo_name}} {{[-p|--download-path]}} {{path/to/directory}}
```

- Synchronize all packages and metadata:

```
dnf reposync --repoid {{repo_name}} --download-metadata
```

- Download only newest packages per-repo:

```
dnf reposync --repoid {{repo_name}} {{[-n|--newest-only]}}
```

- Just print URLs of what would be downloaded, don't download:

```
dnf reposync --repoid {{repo_name}} {{[-u|--urls]}}
```

- Try to set the timestamps of the downloaded files to those on the remote side:

```
dnf reposync --repoid {{repo_name}} --remote-time
```

dnf versionlock

Protect packages from updates to newer versions.

Not default to **`dnf`** but supported via **`dnf-plugins-core`**.

See also: **`dnf`**.

More information: https://dnf-plugins-core.readthedocs.io/en/latest/versionlock.html.

- List the current versionlock entries:

```
dnf versionlock
```

- Add a versionlock for all available packages matching the spec:

```
dnf versionlock add {{package}}
```

- Add an exclude (within versionlock) for the available packages matching the spec:

```
dnf versionlock exclude {{package}}
```

- Remove any matching versionlock entries:

```
dnf versionlock delete {{package}}
```

- Remove all versionlock entries:

```
dnf versionlock clear
```

dnf

Package management utility for RHEL, Fedora, and CentOS (replaces yum).

Some subcommands such as **`group`** and **`config-manager`** have their own usage documentation.

For equivalent commands in other package managers, see https://wiki.archlinux.org/title/Pacman/Rosetta.

More information: https://dnf.readthedocs.io.

- Upgrade installed packages to the newest available versions:

```
sudo dnf {{[up|upgrade]}}
```

- Search packages via keywords:

```
dnf {{[se|search]}} {{keyword1 keyword2 ...}}
```

- Display details about a package:

```
dnf {{[if|info]}} {{package}}
```

- Install a new package (use `--assumeyes` to confirm all prompts automatically):

```
sudo dnf {{[in|install]}} {{package1 package2 ...}}
```

- Remove a package:

```
sudo dnf {{[rm|remove]}} {{package1 package2 ...}}
```

- List installed packages:

```
dnf {{[ls|list]}} --installed
```

- Find which packages provide a given command:

```
dnf {{[wp|provides]}} {{command}}
```

- View all past operations:

```
dnf {{[hist|history]}}
```

dnf5 group

This command is an alias of **`dnf group`**.

Note: From Fedora 37 to 40 (inclusive), **`dnf`** runs DNF v4 while **`dnf5`** runs DNF v5.

- View documentation for the original command:

  ```
  tldr dnf group
  ```

dnf5

Package management utility for RHEL, Fedora, and CentOS (it replaces dnf, which in turn replaced yum).

DNF5 is a C++ rewrite of the DNF package manager featuring improved performance and a smaller size.

For equivalent commands in other package managers, see https://wiki.archlinux.org/title/Pacman/Rosetta.

More information: https://dnf5.readthedocs.io.

- Upgrade installed packages to the newest available versions:

```
sudo dnf5 upgrade
```

- Search packages via keywords:

```
dnf5 search {{keyword1 keyword2 ...}}
```

- Display details about a package:

```
dnf5 info {{package}}
```

- Install new packages (Note: Use `-y` to confirm all prompts automatically):

```
sudo dnf5 install {{package1 package2 ...}}
```

- Remove packages:

```
sudo dnf5 remove {{package1 package2 ...}}
```

- List installed packages:

```
dnf5 list --installed
```

- Find which packages provide a given command:

```
dnf5 provides {{command}}
```

- Remove or expire cached data:

```
sudo dnf5 clean all
```

dnsdomainname

Show the system's DNS domain name.

Note: The tool uses **`gethostname`** to get the hostname of the system and then **`getaddrinfo`** to resolve it into a canonical name.

More information: https://www.gnu.org/software/inetutils/manual/inetutils.html#dnsdomainname-invocation.

- Show the system's DNS domain name:

```
dnsdomainname
```

dnsmap

The dnsmap command scans a domain for common subdomains e.g. smtp.domain.org.

More information: https://github.com/resurrecting-open-source-projects/dnsmap.

- Scan for subdomains using the internal wordlist:

```
dnsmap {{example.com}}
```

- Specify a list of subdomains to check for:

```
dnsmap {{example.com}} -w {{path/to/wordlist.txt}}
```

- Store results to a CSV file:

```
dnsmap {{example.com}} -c {{path/to/file.csv}}
```

- Ignore 2 IPs that are false positives (up to 5 possible):

```
dnsmap {{example.com}} -i {{123.45.67.89,98.76.54.32}}
```

dnsrecon

DNS enumeration tool.

More information: https://github.com/darkoperator/dnsrecon.

- Scan a domain and save the results to an SQLite database:

```
dnsrecon {{[-d|--domain]}} {{example.com}} --db {{path/to/database.sqlite}}
```

- Scan a domain, specifying the nameserver and performing a zone transfer:

```
dnsrecon {{[-d|--domain]}} {{example.com}} {{[-n|--name_server]}} {{nameserver.example.com}} {{[-t|--type]}} axfr
```

- Scan a domain, using a brute-force attack and a dictionary of subdomains and hostnames:

```
dnsrecon {{[-d|--domain]}} {{example.com}} {{[-D|--dictionary]}} {{path/to/dictionary.txt}} {{[-t|--type]}} brt
```

- Scan a domain, performing a reverse lookup of IP ranges from the SPF record and saving the results to a JSON file:

```
dnsrecon {{[-d|--domain]}} {{example.com}} -s {{[-j|--json]}}
```

- Scan a domain, performing a Google enumeration and saving the results to a CSV file:

```
dnsrecon {{[-d|--domain]}} {{example.com}} -g {{[-c|--csv]}}
```

- Scan a domain, performing DNS cache snooping:

```
dnsrecon {{[-d|--domain]}} {{example.com}} {{[-t|--type]}} snoop {{[-n|--name_server]}} {{nameserver.example.com}} {{[-D|--dictionary]}} {{path/to/dictionary.txt}}
```

- Scan a domain, performing zone walking:

```
dnsrecon {{[-d|--domain]}} {{example.com}} {{[-t|--type]}} zonewalk
```

dnstracer

The dnstracer command determines where a DNS gets its information from.

More information: https://manned.org/dnstracer.

- Find out where your local DNS got the information on www.example.com:

```
dnstracer {{www.example.com}}
```

- Start with a [s]pecific DNS that you already know:

```
dnstracer -s {{dns.example.org}} {{www.example.com}}
```

- Only query IPv[4] servers:

```
dnstracer -4 {{www.example.com}}
```

- [r]etry each request 5 times on failure:

```
dnstracer -r {{5}} {{www.example.com}}
```

- Display all steps during execution:

```
dnstracer -v {{www.example.com}}
```

- Display an [o]verview of all received answers after execution:

```
dnstracer -o {{www.example.com}}
```

do-release-upgrade

The Ubuntu release upgrader.

More information: https://manned.org/do-release-upgrade.8.

- Upgrade to the latest release:

```
sudo do-release-upgrade
```

- Upgrade to the latest development release:

```
sudo do-release-upgrade {{[-d|--devel-release]}}
```

- Upgrade to the latest proposed release:

```
sudo do-release-upgrade {{[-p|--proposed]}}
```

dockerd

A persistent process to start and manage Docker containers.

More information: https://docs.docker.com/reference/cli/dockerd/.

- Run Docker daemon:

```
dockerd
```

- Run Docker daemon and configure it to listen to specific sockets (UNIX and TCP):

```
dockerd {{[-H|--host]}} unix://{{path/to/tmp.sock}} {{[-H|--host]}} tcp://{{ip}}
```

- Run with specific daemon PID file:

```
dockerd {{[-p|--pidfile]}} {{path/to/pid_file}}
```

- Run in debug mode:

```
dockerd {{[-D|--debug]}}
```

- Run and set a specific log level:

```
dockerd {{[-l|--log-level]}} {{debug|info|warn|error|fatal}}
```

dolphin

KDE's file manager to manage files and directories.

See also: **`nautilus`**, **`caja`**, **`thunar`**, **`ranger`**.

More information: https://apps.kde.org/dolphin/.

- Launch the file manager:

```
dolphin
```

- Open specific directories:

```
dolphin {{path/to/directory1 path/to/directory2 ...}}
```

- Open with specific files or directories selected:

```
dolphin --select {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- Open a new window:

```
dolphin --new-window
```

- Open specific directories in split view:

```
dolphin --split {{path/to/directory1}} {{path/to/directory2}}
```

- Launch the daemon (only required to use the D-Bus interface):

```
dolphin --daemon
```

- Display help:

```
dolphin {{[-h|--help]}}
```

dos2unix

Change DOS-style line endings to Unix-style.

Replaces CRLF with LF.

See also: **`unix2dos`**, **`unix2mac`**, **`mac2unix`**.

More information: https://manned.org/dos2unix.

- Change the line endings of a file:

```
dos2unix {{path/to/file}}
```

- Create a copy with Unix-style line endings:

```
dos2unix {{[-n|--newfile]}} {{path/to/file}} {{path/to/new_file}}
```

- Display file information:

```
dos2unix {{[-i|--info]}} {{path/to/file}}
```

- Keep/add/remove Byte Order Mark:

```
dos2unix --{{keep-bom|add-bom|remove-bom}} {{path/to/file}}
```

dphys-swapfile

Manage the swap file on Debian-based Linux systems.

More information: https://manned.org/dphys-swapfile.

- Disable the swap file:

```
dphys-swapfile swapoff
```

- Enable the swap file:

```
dphys-swapfile swapon
```

- Create a new swap file:

```
dphys-swapfile setup
```

dpigs

Show which installed packages occupy the most space on **apt** based systems.

More information: https://manned.org/dpigs.

- Display the n largest packages on the system:

```
dpigs {{[-n|--lines]}} {{n}}
```

- Use the specified file instead of the default dpkg status file:

```
dpigs {{[-s|--status]}} {{path/to/file}}
```

- Display the largest source packages of binary packages installed on the system:

```
dpigs {{[-S|--source]}}
```

- Display package sizes in human-readable format:

```
dpigs {{[-H|--human-readable]}}
```

- Display help:

```
dpigs {{[-h|--help]}}
```

dpkg-deb

Pack, unpack and provide information about Debian archives.

More information: https://manned.org/dpkg-deb.

- Display information about a package:

```
dpkg-deb {{[-I|--info]}} {{path/to/file.deb}}
```

- Display the package's name and version on one line:

```
dpkg-deb {{[-W|--show]}} {{path/to/file.deb}}
```

- List the package's contents:

```
dpkg-deb {{[-c|--contents]}} {{path/to/file.deb}}
```

- Extract package's contents into a directory:

```
dpkg-deb {{[-x|--extract]}} {{path/to/file.deb}} {{path/to/directory}}
```

- Create a package from a specified directory:

```
dpkg-deb {{[-b|--build]}} {{path/to/directory}}
```

dpkg-query

Display information about installed packages.

More information: https://manned.org/dpkg-query.1.

- List all installed packages:

```
dpkg-query {{[-l|--list]}}
```

- List installed packages matching a pattern:

```
dpkg-query {{[-l|--list]}} '{{libc6*}}'
```

- List all files installed by a package:

```
dpkg-query {{[-L|--listfiles]}} {{libc6}}
```

- Show information about a package:

```
dpkg-query {{[-s|--status]}} {{libc6}}
```

- Search for packages that own files matching a pattern:

```
dpkg-query {{[-S|--search]}} {{/etc/ld.so.conf.d}}
```

dpkg-reconfigure

Reconfigure an already installed package.

More information: https://manned.org/dpkg-reconfigure.8.

- Reconfigure one or more packages:

```
dpkg-reconfigure {{package1 package2 ...}}
```

- Reconfigure console font setup:

```
dpkg-reconfigure console-setup
```

- Reconfigure the timezone:

```
dpkg-reconfigure tzdata
```

dpkg

Debian package manager.

Some subcommands such as **deb** have their own usage documentation.

For equivalent commands in other package managers, see https://wiki.archlinux.org/title/Pacman/Rosetta.

More information: https://manned.org/dpkg.

- Install a package:

```
sudo dpkg {{[-i|--install]}} {{path/to/file.deb}}
```

- Remove a package:

```
sudo dpkg {{[-r|--remove]}} {{package}}
```

- List installed packages:

```
dpkg {{[-l|--list]}} {{pattern}}
```

- List a package's contents:

```
dpkg {{[-L|--listfiles]}} {{package}}
```

- List contents of a local package file:

```
dpkg {{[-c|--contents]}} {{path/to/file.deb}}
```

- Find out which package owns a file:

```
dpkg {{[-S|--search]}} {{path/to/file}}
```

- Purge an installed or already removed package, including configuration:

```
sudo dpkg {{[-P|--purge]}} {{package}}
```

dracut

Generate initramfs images to boot the Linux kernel.

Dracut uses options from configuration files in **`/etc/dracut.conf`**, **`/etc/dracut.conf.d/*.conf`** and **`/usr/lib/dracut/dracut.conf.d/*.conf`** by default.

More information: https://github.com/dracut-ng/dracut-ng/blob/main/man/dracut.8.adoc.

- Generate an initramfs image for the current kernel without overriding any options:

```
dracut
```

- Generate an initramfs image for the current kernel and overwrite the existing one:

```
dracut {{[-f|--force]}}
```

- Generate an initramfs image for a specific kernel:

```
dracut --kver {{kernel_version}}
```

- List available modules:

```
dracut --list-modules
```

drawing

Free basic raster image editor in GNOME desktop environment.

More information: https://maoschanz.github.io/drawing/.

- Start Drawing:

```
drawing
```

- Open specific files:

```
drawing {{path/to/image1 path/to/image2 ...}}
```

- Open specific files in a new window:

```
drawing --new-window {{path/to/image1 path/to/image2 ...}}
```

dropbearconvert

Convert between Dropbear and OpenSSH private key formats.

More information: https://manned.org/dropbearconvert.1.

- Convert an OpenSSH private key to the Dropbear format:

```
dropbearconvert openssh dropbear {{path/to/input_key}}
{{path/to/output_key}}
```

- Convert a Dropbear private key to the OpenSSH format:

```
dropbearconvert dropbear openssh {{path/to/input_key}}
{{path/to/output_key}}
```

dropbearkey

Generate SSH keys in Dropbear format.

More information: https://manned.org/dropbearkey.

- Generate an SSH key of [t]ype ed25519 and write it to key [f]ile:

```
dropbearkey -t {{ed25519}} -f {{path/to/key_file}}
```

- Generate an SSH key of [t]ype ecdsa and write it to key [f]ile:

```
dropbearkey -t {{ecdsa}} -f {{path/to/key_file}}
```

- Generate an SSH key of [t]ype RSA with 4096-bit key [s]ize and write it to key [f]ile:

```
dropbearkey -t {{rsa}} -s {{4096}} -f {{path/to/key_file}}
```

- Print the private key fingerprint and public key in key [f]ile:

```
dropbearkey -y -f {{path/to/key_file}}
```

dstat

Versatile tool for generating system resource statistics.

Note: dstat is deprecated and no longer maintained.

More information: https://github.com/dstat-real/dstat.

- Display CPU, disk, net, paging and system statistics:

```
dstat
```

- Display statistics every 5 seconds and 4 updates only:

```
dstat {{5}} {{4}}
```

- Display CPU and memory statistics only:

```
dstat {{[-c|--cpu]}} {{[-m|--mem]}}
```

- List all available dstat plugins:

```
dstat --list
```

- Display the process using the most memory and most CPU:

```
dstat --top-mem --top-cpu
```

- Display battery percentage and remaining battery time:

```
dstat --battery --battery-remain
```

dump.exfat

Show on-disk information of an exFAT filesystem.

More information: https://manned.org/dump.exfat.

- Print on-disk information for a given filesystem:

```
dump.exfat {{/dev/sdXY}}
```

dumpe2fs

Print the super block and blocks group information for ext2/ext3/ext4 filesystems.

Unmount the partition before running this command using **`umount {{device}}`**.

More information: https://manned.org/dumpe2fs.

- Display ext2, ext3 and ext4 filesystem information:

```
dumpe2fs {{/dev/sdXN}}
```

- Display the blocks which are reserved as bad in the filesystem:

```
dumpe2fs -b {{/dev/sdXN}}
```

- Force display filesystem information even with unrecognizable feature flags:

```
dumpe2fs -f {{/dev/sdXN}}
```

- Only display the superblock information and not any of the block group descriptor detail information:

```
dumpe2fs -h {{/dev/sdXN}}
```

- Print the detailed group information block numbers in hexadecimal format:

```
dumpe2fs -x {{/dev/sdXN}}
```

duperemove

Finds duplicate filesystem extents and optionally schedule them for deduplication.

An extent is small part of a file inside the filesystem.

On some filesystems one extent can be referenced multiple times, when parts of the content of the files are identical.

More information: https://markfasheh.github.io/duperemove/.

- Search for duplicate extents in a directory and show them:

```
duperemove -r {{path/to/directory}}
```

- Deduplicate duplicate extents on a Btrfs or XFS (experimental) filesystem:

```
duperemove -r -d {{path/to/directory}}
```

- Use a hash file to store extent hashes (less memory usage and can be reused on subsequent runs):

```
duperemove -r -d --hashfile={{path/to/hashfile}} {{path/to/directory}}
```

- Limit I/O threads (for hashing and dedupe stage) and CPU threads (for duplicate extent finding stage):

```
duperemove -r -d --hashfile={{path/to/hashfile}} --io-threads={{n}} --cpu-threads={{n}} {{path/to/directory}}
```

dutree

Analyze file system usage with colorful text-based trees.

More information: https://github.com/nachoparker/dutree#usage.

- Show a graphical tree of the current directory:

```
dutree
```

- Show a specific directory:

```
dutree {{path/to/directory}}
```

- Aggregate items smaller than a number of KB (or M for MB, or G for GB):

```
dutree --aggr {{number}}K
```

- Show subdirectories up to the specified depth (default is 1):

```
dutree --depth {{depth}}
```

- Skip directories for a fast local overview:

```
dutree --files-only
```

- Exclude hidden files:

```
dutree --no-hidden
```

dysk

Display filesystem information in a table.

More information: https://manned.org/dysk.

- Get a standard overview of your usual disks:

```
dysk
```

- Sort by free size:

```
dysk {{[-s|--sort]}} free
```

- Include only HDD disks:

```
dysk {{[-f|--filter]}} 'disk = HDD'
```

- Exclude SSD disks:

```
dysk {{[-f|--filter]}} 'disk <> SSD'
```

- Display disks with high utilization or low free space:

```
dysk {{[-f|--filter]}} 'use > 65% | free < 50G'
```

e2freefrag

Print the free space fragmentation information for ext2/ext3/ext4 filesystems.

More information: https://manned.org/e2freefrag.

- Check how many free blocks are present as contiguous and aligned free space:

```
e2freefrag {{/dev/sdXN}}
```

- Specify [c]hunk size in kilobytes to print how many free chunks are available:

```
e2freefrag -c {{chunk_size_in_kb}} {{/dev/sdXN}}
```

e2fsck

Check a Linux ext2/ext3/ext4 filesystem. The partition should be unmounted.

More information: https://manned.org/e2fsck.

- Check filesystem, reporting any damaged blocks:

```
sudo e2fsck {{/dev/sdXN}}
```

- Check filesystem and automatically repair ([p]reen) any damaged blocks:

```
sudo e2fsck -p {{/dev/sdXN}}
```

- Check filesystem in read only mode:

```
sudo e2fsck -c {{/dev/sdXN}}
```

- [f]orce checking even if the filesystem seems clean:

```
sudo e2fsck -f {{/dev/sdXN}}
```

- Perform an exhaustive, non-destructive read-write test for bad blocks and blacklist them:

```
sudo e2fsck -fccky {{/dev/sdXN}}
```

e2image

Save critical ext2/ext3/ext4 filesystem metadata to a file.

More information: https://manned.org/e2image.

- Write metadata located on device to a specific file:

```
e2image {{/dev/sdXN}} {{path/to/image_file}}
```

- Print metadata located on device to `stdout`:

```
e2image {{/dev/sdXN}} -
```

- Restore the filesystem metadata back to the device:

```
e2image -I {{/dev/sdXN}} {{path/to/image_file}}
```

- Create a large [r]aw sparse file with metadata at proper offsets:

```
e2image -r {{/dev/sdXN}} {{path/to/image_file}}
```

- Create a [Q]COW2 image file instead of a normal or raw image file:

```
e2image -Q {{/dev/sdXN}} {{path/to/image_file}}
```

e2label

Change the label on an ext2/ext3/ext4 filesystem.

More information: https://manned.org/e2label.

- Change the volume label on a specific ext partition:

```
sudo e2label {{/dev/sda1}} "{{label_name}}"
```

e2undo

Replay undo logs for an ext2/ext3/ext4 filesystem.

This can be used to undo a failed operation by an e2fsprogs program.

More information: https://manned.org/e2undo.

- Display information about a specific undo file:

```
e2undo -h {{path/to/undo_file}} {{/dev/sdXN}}
```

- Perform a dry-run and display the candidate blocks for replaying:

```
e2undo -nv {{path/to/undo_file}} {{/dev/sdXN}}
```

- Perform an undo operation:

```
e2undo {{path/to/undo_file}} {{/dev/sdXN}}
```

- Perform an undo operation and display [v]erbose information:

```
e2undo -v {{path/to/undo_file}} {{/dev/sdXN}}
```

- Write the old contents of the block to an undo file before overwriting a file system block:

```
e2undo -z {{path/to/file.e2undo}} {{path/to/undo_file}} {{/dev/sdXN}}
```

e4defrag

Defragment an ext4 filesystem.

More information: https://manned.org/e4defrag.

- Defragment the filesystem:

```
e4defrag {{/dev/sdXN}}
```

- See how fragmented a filesystem is:

```
e4defrag -c {{/dev/sdXN}}
```

- Print errors and the fragmentation count before and after each file:

```
e4defrag -v {{/dev/sdXN}}
```

ebuild

A low level interface to the Gentoo Portage system.

More information: https://wiki.gentoo.org/wiki/Ebuild.

- Create or update the package manifest:

```
ebuild {{path/to/file.ebuild}} manifest
```

- Clean the temporary build directories for the build file:

```
ebuild {{path/to/file.ebuild}} clean
```

- Fetch sources if they do not exist:

```
ebuild {{path/to/file.ebuild}} fetch
```

- Extract the sources to a temporary build directory:

```
ebuild {{path/to/file.ebuild}} unpack
```

- Compile the extracted sources:

```
ebuild {{path/to/file.ebuild}} compile
```

- Install the package to a temporary install directory:

```
ebuild {{path/to/file.ebuild}} install
```

- Install the temporary files to the live filesystem:

```
ebuild {{path/to/file.ebuild}} qmerge
```

- Fetch, unpack, compile, install and qmerge the specified ebuild file:

```
ebuild {{path/to/file.ebuild}} merge
```

eclean-kernel

Remove old kernels in Gentoo.

More information: https://wiki.gentoo.org/wiki/Kernel/Removal#Using_eclean-kernel.

- List all kernel files:

```
sudo eclean-kernel {{[-l|--list-kernels]}}
```

- Remove all kernels except for the two newest ones:

```
sudo eclean-kernel {{[-n|--num]}} 2
```

- Remove all kernels except for the two newest ones and ask before removal:

```
sudo eclean-kernel {{[-a|--all]}} {{[-n|--num]}} 2
```

eclean

Clean repository source files and binary packages.

More information: https://wiki.gentoo.org/wiki/Eclean.

- Clean the source file directory:

```
sudo eclean distfiles
```

- Clean the binary package directory:

```
sudo eclean packages
```

- Clean the distfiles of all uninstalled packages, but keep the distfiles of installed packages:

```
sudo eclean {{[-d|--deep]}} {{[-n|--package-names]}} distfiles
```

- Clean the binary packages of all uninstalled packages, but keep the binaries of installed packages:

```
sudo eclean {{[-d|--deep]}} {{[-n|--package-names]}} packages
```

edit

An alias to a **`run-mailcap`**'s action edit.

Originally **`run-mailcap`** is used to process/edit mime-type/file.

More information: https://www.computerhope.com/unix/uedit.htm.

- Edit action can be used to view any file on default mailcap explorer:

```
edit {{filename}}
```

- With `run-mailcap`:

```
run-mailcap --action=edit {{filename}}
```

edquota

Edit quotas for a user or group. By default it operates on all filesystems with quotas.

Quota information is stored permanently in the **`quota.user`** and **`quota.group`** files in the root of the filesystem.

More information: https://manned.org/edquota.

- Edit quota of the current user:

```
edquota {{[-u|--user]}} $(whoami)
```

- Edit quota of a specific user:

```
sudo edquota {{[-u|--user]}} {{username}}
```

- Edit quota for a group:

```
sudo edquota {{[-g|--group]}} {{group}}
```

- Restrict operations to a given filesystem (by default edquota operates on all filesystems with quotas):

```
sudo edquota {{[-f|--file-system]}} {{filesystem}}
```

- Edit the default grace period:

```
sudo edquota {{[-t|--edit-period]}}
```

- Duplicate a quota to other users:

```
sudo edquota {{[-p|--prototype]}} {{reference_user}}
{{destination_user1 destination_user2 ...}}
```

efibootdump

Dump boot entries from a variable or a file.

More information: https://manned.org/efibootdump.

- Dump boot entries from a file:

```
efibootdump {{[-f|--file]}} {{/sys/firmware/efi/efivars/
Boot0000-8be4df61-93ca-11d2-aa0d-00e098032b8c}}
```

efibootmgr

Manipulate the UEFI Boot Manager.

More information: https://manned.org/efibootmgr.

- List all boot options with their numbers:

```
efibootmgr {{[-u|--unicode]}}
```

- Add UEFI Shell v2 as a boot option:

```
sudo efibootmgr {{[-c|--create]}} {{[-d|--disk]}} {{/dev/
sda}} {{[-p|--part]}} {{1}} {{[-l|--loader]}}
"{{\path\to\shell.efi}}" {{[-L|--label]}} "{{UEFI Shell}}"
```

- Add Linux as a boot option:

```
sudo efibootmgr {{[-c|--create]}} {{[-d|--disk]}} {{/dev/
sda}} {{[-p|--part]}} {{1}} {{[-l|--loader]}}
"{{\vmlinuz}}" {{[-u|--unicode]}} "{{kernel_cmdline}}" {{[-
L|--label]}} "{{Linux}}"
```

- Change the current boot order:

```
sudo efibootmgr {{[-o|--bootorder]}}
{{0002,0008,0001,001A,...}}
```

- Delete a boot option:

```
sudo efibootmgr {{[-b|--bootnum]}} {{0008}} {{[-B|--delete-
bootnum]}}
```

efivar

Manage UEFI variables.

More information: https://manned.org/efivar.

- List all UEFI variables:

```
efivar {{[-l|--list]}}
```

- Print the contents of a variable:

```
efivar {{[-n|--name]}} {{146b234d-4052-4e07-
b326-11220f8e1fe8-lBoot0000}} {{[-p|--print]}}
```

eglinfo

Display platform EGL information.

More information: https://github.com/dv1/eglinfo.

- Display full platform information:

```
eglinfo
```

- Display a brief version of platform information:

```
eglinfo -B
```

- Display help:

```
eglinfo -h
```

ego

Funtoo's official system personality management tool.

More information: https://funtoo-ego.readthedocs.io/en/develop/.

- Synchronize the Portage tree:

  ```
  ego sync
  ```

- Update the bootloader configuration:

  ```
  ego boot update
  ```

- Read a Funtoo wiki page by name:

  ```
  ego doc {{wiki_page}}
  ```

- Print current profile:

  ```
  ego profile show
  ```

- Enable/Disable mix-ins:

  ```
  ego profile mix-in +{{gnome}} -{{kde-plasma-5}}
  ```

- Query Funtoo bugs, related to a specified package:

  ```
  ego query bug {{package}}
  ```

einfo

Provides the number of records indexed in each database field, the last update date of the database, and the available links from the database to other Entrez databases.

More information: https://www.ncbi.nlm.nih.gov/books/NBK179288/.

- Print all database names:

```
einfo -dbs
```

- Print all information of the protein database in XML format:

```
einfo -db {{protein}}
```

- Print all fields of the nuccore database:

```
einfo -db {{nuccore}} -fields
```

- Print all links of the protein database:

```
einfo -db {{protein}} -links
```

eix

Utilities for searching local Gentoo packages.

Update local package cache using **`eix-update`**.

More information: https://wiki.gentoo.org/wiki/Eix.

- Search for a package:

```
eix {{query}}
```

- Search for installed packages:

```
eix --installed {{query}}
```

- Search in package descriptions:

```
eix --description "{{description}}"
```

- Search by package license:

```
eix --license {{license}}
```

- Exclude results from search:

```
eix --not --license {{license}}
```

eject

Eject CDs, floppy disks, tape drives, and USB sticks.

More information: https://manned.org/eject.

- Display the default device:

```
eject {{[-d|--default]}}
```

- Eject the default device:

```
eject
```

- Eject a specific device (the default order is cd-rom, scsi, floppy and tape):

```
eject {{/dev/cdrom}}
```

- Toggle whether a device's tray is open or closed:

```
eject {{[-T|--traytoggle]}} {{/dev/cdrom}}
```

- Eject a cd drive:

```
eject {{[-r|--cdrom]}} {{/dev/cdrom}}
```

- Eject a floppy drive:

```
eject {{[-f|--floppy]}} {{/mnt/floppy}}
```

- Eject a tape drive:

```
eject {{[-q|--tape]}} {{/mnt/tape}}
```

- Set whether the physical eject button is [i]gnored (`on` prevents ejecting):

```
eject {{[-i|--manualeject]}} {{on|off}}
```

elink

Look up precomputed neighbors within a database, or find associated records in other databases.

It is part of the **edirect** package.

More information: https://www.ncbi.nlm.nih.gov/books/NBK179288/.

- Search pubmed then find related sequences:

```
esearch -db pubmed -query "{{selective serotonin reuptake inhibitor}}" | elink -target nuccore
```

- Search nucleotide then find related biosamples:

```
esearch -db nuccore -query "{{insulin [PROT] AND rodents [ORGN]}}" | elink -target biosample
```

emaint

Perform Portage maintenance tasks.

More information: https://wiki.gentoo.org/wiki/Portage#emaint.

- Synchronize repositories that are set to auto-sync (default for most repositories):

```
sudo emaint sync {{[-a|--auto]}}
```

- Synchronize a specific repository:

```
sudo emaint sync {{[-r|--repo]}} {{repository}}
```

- Synchronize all repositories:

```
sudo emaint sync {{[-A|--allrepos]}}
```

- Clear the Portage resume list:

```
sudo emaint cleanresume {{[-f|--fix]}}
```

- Clean Portage logs:

```
sudo emaint logs {{[-C|--clean]}}
```

emerge

Gentoo Linux package manager utility.

For equivalent commands in other package managers, see https://wiki.archlinux.org/title/Pacman/Rosetta.

More information: https://wiki.gentoo.org/wiki/Portage#emerge.

- Synchronize all packages:

```
sudo emerge --sync
```

- Update all packages, including dependencies:

```
sudo emerge {{[-avuDN|--ask --verbose --update --deep --newuse]}} @world
```

- Resume a failed update, skipping the failing package:

```
sudo emerge --resume --skipfirst
```

- Install a new package, with confirmation:

```
sudo emerge {{[-av|--ask --verbose]}} {{package}}
```

- Remove a package and its dependencies with confirmation:

```
sudo emerge {{[-avc|--ask --verbose --depclean]}} {{package}}
```

- Remove orphaned packages (installed as dependencies but no longer required by any package):

```
sudo emerge {{[-avc|--ask --verbose --depclean]}}
```

- Search the package database for a keyword:

```
emerge {{[-S|--searchdesc]}} {{keyword}}
```

engrampa

Package files into zip/tar file in MATE desktop environment.

See also: **zip**, **tar**.

More information: https://github.com/mate-desktop/engrampa.

- Start Engrampa:

```
engrampa
```

- Open specific archives:

```
engrampa {{path/to/archive1.tar path/to/archive2.tar ...}}
```

- Archive specific files and/or directories recursively:

```
engrampa --add-to={{path/to/compressed.tar}} {{path/to/
file_or_directory1 path/to/file_or_directory2 ...}}
```

- Extract files and/or directories from archives to a specific path:

```
engrampa --extract-to={{path/to/directory}} {{path/to/
archive1.tar path/to/archive2.tar ...}}
```

enum4linux

Enumerate Windows and Samba information from remote systems.

More information: https://labs.portcullis.co.uk/tools/enum4linux/.

- Try to enumerate using all methods:

```
enum4linux -a {{remote_host}}
```

- Enumerate using given login credentials:

```
enum4linux -u {{user_name}} -p {{password}} {{remote_host}}
```

- List usernames from a given host:

```
enum4linux -U {{remote_host}}
```

- List shares:

```
enum4linux -S {{remote_host}}
```

- Get OS information:

```
enum4linux -o {{remote_host}}
```

envycontrol

GPU switching utility for Nvidia Optimus laptops.

More information: https://github.com/bayasdev/envycontrol.

- Switch between different GPU modes:

```
sudo envycontrol {{[-s|--switch]}} {{nvidia|integrated|hybrid}}
```

- Specify your display manager manually:

```
envycontrol --dm
```

- Check current GPU mode:

```
sudo envycontrol {{[-q|--query]}}
```

- Reset settings:

```
sudo envycontrol --reset
```

- Display help:

```
envycontrol {{[-h|--help]}}
```

- Display version:

```
envycontrol {{[-v|--version]}}
```

eog

View and manage images.

More information: https://manned.org/eog.

- Open an image file:

```
eog {{path/to/image.png}}
```

- Open multiple images:

```
eog {{image1.jpg image2.png ...}}
```

- Open images in fullscreen mode:

```
eog --fullscreen {{path/to/image.jpg}}
```

- Start a slideshow:

```
eog --slide-show {{path/to/directory}}
```

- Open a single image without image collection:

```
eog --single-window {{path/to/image.jpg}}
```

- Open an image in a new window instance:

```
eog --new-instance {{path/to/image.jpg}}
```

eopkg

Package manager for Solus.

More information: https://help.getsol.us/docs/user/package-management/basics.

- Install a specific package:

```
sudo eopkg install {{package}}
```

- Update all packages:

```
sudo eopkg upgrade
```

- Search for packages:

```
sudo eopkg search {{search_term}}
```

equery

View information about Portage packages.

More information: https://wiki.gentoo.org/wiki/Equery.

- List all installed packages:

```
equery list '*'
```

- Search for installed packages in the Portage tree and in overlays:

```
equery list -po {{package1 package2 ...}}
```

- List all packages that depend on a given package:

```
equery depends {{package}}
```

- List all packages that a given package depends on:

```
equery depgraph {{package}}
```

- List all files installed by a package:

```
equery files --tree {{package}}
```

esearch

Perform a new Entrez search using terms in indexed fields.

It is part of the **edirect** package.

More information: https://www.ncbi.nlm.nih.gov/books/NBK179288/.

- Search the pubmed database for selective serotonin reuptake inhibitor:

```
esearch -db pubmed -query "{{selective serotonin reuptake inhibitor}}"
```

- Search the protein database using a query and regex:

```
esearch -db {{protein}} -query '{{Escherichia*}}'
```

- Search the nucleotide database for sequences whose metadata contain insulin and rodents:

```
esearch -db nuccore -query "{{insulin [PROT] AND rodents [ORGN]}}"
```

- Display help:

```
esearch {{[-h|-help]}}
```

eselect kernel

An **`eselect`** module for managing the **`/usr/src/linux`** symlink.

More information: https://wiki.gentoo.org/wiki/Eselect#Kernel.

- List available kernel symlink targets with their numbers:

  ```
  eselect kernel list
  ```

- Set the `/usr/src/linux` symlink by name or number from the `list` command:

  ```
  eselect kernel set {{name|number}}
  ```

- Show what the current kernel symlink points to:

  ```
  eselect kernel show
  ```

- Set the kernel symlink to the currently running kernel:

  ```
  eselect kernel update
  ```

eselect locale

An **`eselect`** module for managing the **`LANG`** environment variable, which sets the system language.

More information: https://wiki.gentoo.org/wiki/Eselect#Locale.

- List available locales:

  ```
  eselect locale list
  ```

- Set the `LANG` environment variable in `/etc/profile.env` by name or index from the `list` command:

  ```
  eselect locale set {{name|index}}
  ```

- Display the value of `LANG` in `/etc/profile.env`:

  ```
  eselect locale show
  ```

eselect news

An **eselect** module for reading Gentoo news items.

Note: Portage will print a notice when a repository is synchronized and there are unread news items.

More information: https://wiki.gentoo.org/wiki/Eselect#News.

- List available news items with their numbers (all by default):

  ```
  eselect news list {{all|new}}
  ```

- Print the specified news items:

  ```
  eselect news read {{number1 number2 ...}}
  ```

- Print all unread news items:

  ```
  eselect news read
  ```

- Mark the specified news items as unread:

  ```
  eselect news unread {{number1 number2 ...}}
  ```

- Delete all read news items:

  ```
  eselect news purge
  ```

- Print the number of available news items (new by default):

  ```
  eselect news count {{all|new}}
  ```

eselect profile

An **`eselect`** module for managing the **`/etc/portage/make.profile`** symlink, which sets the system profile.

More information: https://wiki.gentoo.org/wiki/Eselect#Profile.

- List available profile symlink targets with their numbers:

  ```
  eselect profile list
  ```

- Set the `/etc/portage/make.profile` symlink by name or number from the `list` command:

  ```
  eselect profile set {{name|number}}
  ```

- Show the current system profile:

  ```
  eselect profile show
  ```

eselect repository

An **`eselect`** module for configuring ebuild repositories for Portage.

After enabling a repository, you have to run **`emerge --sync repo_name`** to download ebuilds.

More information: https://wiki.gentoo.org/wiki/Eselect/Repository.

- List all ebuild repositories registered on https://repos.gentoo.org:

```
eselect repository list
```

- List enabled repositories:

```
eselect repository list -i
```

- Enable a repository from the list by its name or index from the `list` command:

```
eselect repository enable {{name|index}}
```

- Enable an unregistered repository:

```
eselect repository add {{name}} {{rsync|git|mercurial|
svn|...}} {{sync_uri}}
```

- Disable repositories without removing their contents:

```
eselect repository disable {{repo1 repo2 ...}}
```

- Disable repositories and remove their contents:

```
eselect repository remove {{repo1 repo2 ...}}
```

- Create a local repository and enable it:

```
eselect repository create {{name}} {{path/to/repo}}
```

eselect

Gentoo's multi-purpose configuration and management tool.

It consists of various modules that take care of individual administrative tasks.

Some subcommands such as **`kernel`**, **`locale`**, **`profile`**, etc. have their own usage documentation.

More information: https://wiki.gentoo.org/wiki/Eselect.

- Display a list of installed modules:

```
eselect
```

- View documentation for a specific module:

```
tldr eselect {{module}}
```

- Display a help message for a specific module:

```
eselect {{module}} help
```

etckeeper

Track system configuration files in Git.

More information: https://manned.org/etckeeper.

- Set up a Git repo and perform various setup tasks (run from `/etc`):

```
sudo etckeeper init
```

- Commit all changes in `/etc`:

```
sudo etckeeper commit {{message}}
```

- Run arbitrary Git commands:

```
sudo etckeeper vcs {{status}}
```

- Check if there are uncommitted changes (only returns an exit code):

```
sudo etckeeper unclean
```

- Destroy existing repo and stop tracking changes:

```
sudo etckeeper uninit
```

ethtool

Display and modify Network Interface Controller (NIC) parameters.

More information: https://manned.org/ethtool.

- Display the current settings for an interface:

```
ethtool {{eth0}}
```

- Display the driver information for an interface:

```
ethtool {{[-i|--driver]}} {{eth0}}
```

- Display all supported features for an interface:

```
ethtool {{[-k|--show-features]}} {{eth0}}
```

- Display the network usage statistics for an interface:

```
ethtool {{[-S|--statistics]}} {{eth0}}
```

- Blink one or more LEDs on an interface for 10 seconds:

```
ethtool {{[-p|--identify]}} {{eth0}} {{10}}
```

- Set the link speed, duplex mode, and parameter auto-negotiation for a given interface:

```
ethtool {{[-s|--change]}} {{eth0}} speed {{10|100|1000}}
duplex {{half|full}} autoneg {{on|off}}
```

eu-readelf

Display information about ELF files.

More information: https://manned.org/eu-readelf.

- Display all extractable information contained in the ELF file:

```
eu-readelf {{[-a|--all]}} {{path/to/file}}
```

- Display the contents of all NOTE segments/sections, or of a particular segment/section:

```
eu-readelf {{[-n|--notes]}} {{.note.ABI-tag}} {{path/to/file}}
```

euse

Enable, disable, and obtain information about Gentoo USE flags.

More information: https://wiki.gentoo.org/wiki/Euse.

- List active global USE flags:

```
euse {{[-a|--active]}} {{[-g|--global]}}
```

- List active local USE flags:

```
euse {{[-a|--active]}} {{[-l|--local]}}
```

- Enable a global USE flag:

```
sudo euse {{[-E|--enable]}} {{use_flag}}
```

- Disable a global USE flag (put a '-' sign in front of the USE flag):

```
sudo euse {{[-D|--disable]}} {{use_flag}}
```

- Remove a global USE flag:

```
sudo euse {{[-P|--prune]}} {{use_flag}}
```

evince

View documents.

More information: https://manned.org/evince.

- Open a PDF file:

```
evince {{path/to/file.pdf}}
```

- Open multiple documents:

```
evince {{file1.pdf file2.pdf ...}}
```

- Open in fullscreen mode:

```
evince {{[-f|--fullscreen]}} {{path/to/file.pdf}}
```

- Open in presentation mode:

```
evince {{[-s|--presentation]}} {{path/to/file.pdf}}
```

- Open a specific page number:

```
evince {{[-i|--page-index]}} {{5}} {{path/to/file.pdf}}
```

evtest

Display information from input device drivers.

More information: https://manned.org/evtest.

- List all detected input devices:

```
sudo evtest
```

- Display events from a specific input device:

```
sudo evtest /dev/input/event{{number}}
```

- Grab the device exclusively, preventing other clients from receiving events:

```
sudo evtest --grab /dev/input/event{{number}}
```

- Query the state of a specific key or button on an input device:

```
sudo evtest --query /dev/input/event{{number}} {{event_type}} {{event_code}}
```

eww

Implement your own custom widgets in any window manager.

More information: https://elkowar.github.io/eww.

- Start the daemon:

```
eww daemon
```

- Open a widget:

```
eww {{[-c|--config]}} {{path/to/source_code_directory}} open {{window_name}}
```

- Close a widget:

```
eww {{[-c|--config]}} {{path/to/source_code_directory}} close {{window_name}}
```

- Reload the configuration:

```
eww reload
```

- Kill the daemon:

```
eww kill
```

- Print and watch logs:

```
eww logs
```

exch

Swap the filepaths and filenames of two files.

More information: https://manned.org/exch.

- Swap filepaths:

```
exch {{path/to/file1}} {{path/to/file2}}
```

exec

Execute a command without creating a child process.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-exec.

- Execute a specific command:

```
exec {{command -with -flags}}
```

- Execute a command with a (mostly) empty environment:

```
exec -c {{command -with -flags}}
```

- Execute a command as a login shell:

```
exec -l {{command -with -flags}}
```

- Execute a command with a different name:

```
exec -a {{name}} {{command -with -flags}}
```

exif

Show and change EXIF information in JPEG files.

More information: https://manned.org/exif.

- Show all recognized EXIF information in an image:

```
exif {{path/to/image.jpg}}
```

- Show a table listing known EXIF tags and whether each one exists in an image:

```
exif {{[-l|--list-tags]}} --no-fixup {{path/to/image.jpg}}
```

- Extract the image thumbnail into a separate file:

```
exif {{[-e|--extract-thumbnail]}} {{[-o|--output]}} {{path/to/thumbnail.jpg}} {{path/to/image.jpg}}
```

- Show the raw contents of the "Model" tag in the given image:

```
exif --ifd {{0}} {{[-t|--tag]}} "Model" {{[-m|--machine-readable]}} {{path/to/image.jpg}}
```

- Change the value of the "Artist" tag to John Smith and save to `new.jpg`:

```
exif {{[-o|--output]}} {{path/to/new.jpg}} --ifd {{0}} {{[-t|--tag]}} "Artist" --set-value "John Smith" --no-fixup {{path/to/image.jpg}}
```

exiqgrep

Perl script offering possibilities to **grep** in the Exim queue output.

More information: https://www.exim.org/exim-html-current/doc/html/spec_html/ch-exim_utilities.html.

- Match the sender address using a case-insensitive search:

```
exiqgrep -f '<{{email@example.com}}>'
```

- Match the sender address and display message IDs only:

```
exiqgrep -i -f '<{{email@example.com}}>'
```

- Match the [r]ecipient address:

```
exiqgrep -r '{{email@example.com}}'
```

- Remove all messages matching the sender address from the queue:

```
exiqgrep -i -f '<{{email@example.com}}>' | xargs exim -Mrm
```

- Test for bounced messages:

```
exiqgrep -f '^<>$'
```

- Display the [c]ount of bounced messages:

```
exiqgrep -c -f '^<>$'
```

expac

A data extraction tool for alpm databases, offering printf-like flexibility for pacman-based utilities.

See also: **pacman**.

More information: https://github.com/falconindy/expac.

- List the dependencies of a package:

```
expac {{[-S|--sync]}} '%D' {{package}}
```

- List the optional dependencies of a package:

```
expac {{[-S|--sync]}} "%o" {{package}}
```

- List the download size of packages in MiB:

```
expac {{[-S|--sync]}} {{[-H|--humansize]}} M
'%k\t%n' {{package1 package2 ...}}
```

- List packages marked for upgrade with their download size:

```
expac {{[-S|--sync]}} {{[-H|--humansize]}} M '%k\t%n' $
(pacman -Qqu) | sort {{[-sh|--sort --human-numeric-sort]}}
```

- List explicitly-installed packages with their optional dependencies:

```
expac {{[-d|--delim]}} '\n\n' {{[-l|--listdelim]}}
'\n\t' {{[-Q|--query]}} '%n\n\t%O' $(pacman -Qeq)
```

expect

Script executor that interacts with other programs that require user input.

More information: https://manned.org/expect.

- Execute an expect script from a file:

```
expect {{path/to/file}}
```

- Execute a specified expect script:

```
expect -c "{{commands}}"
```

- Enter an [i]nteractive REPL (use `exit` or `<Ctrl d>` to exit):

```
expect -i
```

export

Export shell variables to child processes.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-export.

- Set an environment variable:

```
export {{VARIABLE}}={{value}}
```

- Unset an environment variable:

```
export -n {{VARIABLE}}
```

- Export a [f]unction to child processes:

```
export -f {{FUNCTION_NAME}}
```

- Append a pathname to the environment variable `PATH`:

```
export PATH=$PATH:{{path/to/append}}
```

- [p]rint a list of active exported variables in shell command form:

```
export -p
```

extrace

Trace exec() calls.

More information: https://github.com/chneukirchen/extrace.

- Trace all program executions occurring on the system:

```
sudo extrace
```

- Run a command and only trace descendants of this command:

```
sudo extrace {{command}}
```

- Print the current working [d]irectory of each process:

```
sudo extrace -d
```

- Resolve the full path of each executable:

```
sudo extrace -l
```

- Display the [u]ser running each process:

```
sudo extrace -u
```

extrepo

Manage external Debian repositories.

It is used to manage external repositories in Debian.

More information: https://manned.org/extrepo.1p.

- Search for a given package:

```
extrepo search {{package}}
```

- Enable the repository:

```
sudo extrepo enable {{repository_name}}
```

- Disable the repository:

```
sudo extrepo disable {{repository_name}}
```

- Update the repository:

```
sudo extrepo update {{repository_name}}
```

extundelete

Recover deleted files from ext3 or ext4 partitions by parsing the journal.

See also: **`date`** for Unix time information and **`umount`** for unmounting partitions.

More information: https://extundelete.sourceforge.net/options.html.

- Restore all deleted files inside partition `N` on device `X`:

```
sudo extundelete {{/dev/sdXN}} --restore-all
```

- Restore a file from a path relative to root (Do not start the path with `/`):

```
extundelete {{/dev/sdXN}} --restore-file {{path/to/file}}
```

- Restore a directory from a path relative to root (Do not start the path with `/`):

```
extundelete {{/dev/sdXN}} --restore-directory {{path/to/
directory}}
```

- Restore all files deleted after January 1st, 2020 (in Unix time):

```
extundelete {{/dev/sdXN}} --restore-all --after
{{1577840400}}
```

eyeD3

Read and manipulate metadata of MP3 files.

More information: https://manned.org/eyeD3.

- View information about an MP3 file:

```
eyeD3 {{filename.mp3}}
```

- Set the title of an MP3 file:

```
eyeD3 {{[-t|--title]}} "{{A Title}}" {{filename.mp3}}
```

- Set the album of all the MP3 files in a directory:

```
eyeD3 {{[-A|--album]}} "{{Album Name}}" {{*.mp3}}
```

- Set the front cover art for an MP3 file:

```
eyeD3 --add-image {{front_cover.jpeg}}:FRONT_COVER:
{{filename.mp3}}
```

f5fpc

A proprietary commercial SSL VPN client by BIG-IP Edge.

More information: https://my.f5.com/manage/s/article/K47922841.

- Open a new VPN connection:

```
sudo f5fpc --start
```

- Open a new VPN connection to a specific host:

```
sudo f5fpc --start --host {{host.example.com}}
```

- Specify a username (user will be prompted for a password):

```
sudo f5fpc --start --host {{host.example.com}} --username
{{user}}
```

- Show the current VPN status:

```
sudo f5fpc --info
```

- Shutdown the VPN connection:

```
sudo f5fpc --stop
```

Factorio

Create and start a headless Factorio server.

More information: https://wiki.factorio.com/Multiplayer.

- Create a new save file:

```
{{path/to/factorio}} --create {{path/to/save_file.zip}}
```

- Start a Factorio server:

```
{{path/to/factorio}} --start-server {{path/to/save_file.zip}}
```

- Create a new save file with specific settings:

```
{{path/to/factorio}} --create {{path/to/save.zip}} --map-gen-settings {{path/to/map-gen-settings.json}} --map-settings {{path/to/map-settings.json}}
```

- Shut the server gracefully:

```
/quit
```

- Display help:

```
{{path/to/factorio}} {{[-h|--help]}}
```

fadvise

Control Linux file caching behavior.

See also: **`fincore`**.

More information: https://manned.org/fadvise.

- Preload a file or directory into cache:

```
fadvise {{[-a|--advice]}} willneeded {{path/to/file_or_directory}}
```

- Suggest dropping a file from cache:

```
fadvise {{path/to/file}}
```

- Display help:

```
fadvise {{[-h|--help]}}
```

fail2ban-client

Configure and control fail2ban server.

More information: https://manned.org/fail2ban-client.

- Retrieve current status of the jail service:

```
fail2ban-client status {{jail}}
```

- Remove the specified IP from the jail service's ban list:

```
fail2ban-client set {{jail}} unbanip {{ip}}
```

- Verify fail2ban server is alive:

```
fail2ban-client ping
```

faillock

Display and modify authentication failure record files.

More information: https://manned.org/faillock.

- List login failures of the current user:

```
faillock
```

- Reset the failure records of the current user:

```
faillock --reset
```

- List login failures of all users:

```
sudo faillock
```

- List login failures of the specified user:

```
sudo faillock --user {{user}}
```

- Reset the failure records of the specified user:

```
sudo faillock --user {{user}} --reset
```

fakeroot

Run a command in an environment faking root privileges for file manipulation.

More information: https://manned.org/fakeroot.1.

- Start the default shell as fakeroot:

```
fakeroot
```

- Run a command as fakeroot:

```
fakeroot -- {{command}} {{command_arguments}}
```

- Run a command as fakeroot and [s]ave the environment to a file on exit:

```
fakeroot -s {{path/to/file}} -- {{command}}
{{command_arguments}}
```

- Load a fakeroot environment and run a command as fakeroot:

```
fakeroot -i {{path/to/file}} -- {{command}}
{{command_arguments}}
```

- Run a command keeping the real ownership of files instead of pretending they are owned by root:

```
fakeroot {{[-u|--unknown-is-real]}} -- {{command}}
{{command_arguments}}
```

- Display help:

```
fakeroot {{[-h|--help]}}
```

faketime

Fake the system time for a command.

More information: https://manned.org/faketime.

- Fake the time to this evening, before printing the result of `date`:

```
faketime '{{today 23:30}}' {{date}}
```

- Open a new Bash shell, which uses yesterday as the current date:

```
faketime '{{yesterday}}' {{bash}}
```

- Simulate how a program would act next Friday night:

```
faketime '{{next Friday 1 am}}' {{path/to/program}}
```

fallocate

Reserve or deallocate disk space to files.

The utility allocates space without zeroing.

More information: https://manned.org/fallocate.

- Reserve a file taking up 700 MiB of disk space:

```
fallocate {{[-l|--length]}} {{700M}} {{path/to/file}}
```

- Shrink an already allocated file by 200 MiB:

```
fallocate {{[-c|--collapse-range]}} {{[-l|--length]}}
{{200M}} {{path/to/file}}
```

- Shrink 20 MB of space after 100 MiB in a file:

```
fallocate {{[-c|--collapse-range]}} {{[-o|--offset]}}
{{100M}} {{[-l|--length]}} {{20M}} {{path/to/file}}
```

fancontrol

Automated fan speed management.

Uses a configuration file to control a wide range of variables.

More information: https://manned.org/fancontrol.

- Start `fancontrol` using default `/etc/fancontrol` configuration file:

  `fancontrol`

- Start `fancontrol` using custom configuration file path:

  `fancontrol {{path/to/config_file}}`

farge

Display the color of a specific pixel on the screen in either hexadecimal or RGB formats.

More information: https://github.com/sdushantha/farge.

- Display a small preview window of a pixel's color with it's hexadecimal value, and copy this value to the clipboard:

```
farge
```

- Copy a pixel's hexadecimal value to the clipboard without displaying a preview window:

```
farge --no-preview
```

- Output a pixel's hexadecimal value to `stdout`, and copy this value to the clipboard:

```
farge --stdout
```

- Output a pixel's RGB value to `stdout`, and copy this value to the clipboard:

```
farge --rgb --stdout
```

- Display a pixel's hexadecimal value as a notification which expires in 5000 milliseconds, and copy this value to the clipboard:

```
farge --notify --expire-time 5000
```

fatlabel

Get or set the label of a FAT32 partition.

More information: https://manned.org/fatlabel.

- Get the label of a FAT32 partition:

```
fatlabel {{/dev/sda1}}
```

- Set the label of a FAT32 partition:

```
fatlabel {{/dev/sdc3}} "{{new_label}}"
```

fatrace

Report file access events.

More information: https://manned.org/fatrace.

- Print file access events in all mounted filesystems to `stdout`:

```
sudo fatrace
```

- Print file access events on the mount of the current directory, with timestamps, to `stdout`:

```
sudo fatrace {{[-c|--current-mount]}} {{[-t|--timestamp]}}
```

fbi

Display images in a terminal framebuffer.

More information: https://manned.org/fbi.

- Display an image:

```
sudo fbi {{path/to/file}}
```

- Display multiple images:

```
sudo fbi {{path/to/file1 path/to/file2 ...}}
```

- Start a slideshow with a custom delay (in seconds):

```
sudo fbi {{[-t|--timeout]}} {{delay}} {{path/to/files/*}}
```

- Scale the image to fit the screen:

```
sudo fbi {{[-a|--autozoom]}} {{path/to/file}}
```

- Display help:

```
fbi {{[-h|--help]}}
```

fbset

Show and modify frame buffer device settings.

More information: https://manned.org/man/fbset.1.

- Show current framebuffer settings:

```
sudo fbset {{[-i|--info]}}
```

- Set a framebuffer mode defined in `/etc/fb.modes`:

```
sudo fbset "{{800}}x{{600}}-{{60}}"
```

- Set an arbitrary framebuffer mode:

```
sudo fbset {{[-g|--geometry]}} {{TTY_horizontal}}
{{TTY_vertical}} {{monitor_horizontal}} {{monitor_vertical}}
{{color_depth}}
```

fcrackzip

ZIP archive password cracking utility.

More information: https://manned.org/fcrackzip.

- Brute-force a password with a length of 4 to 8 characters, and contains only alphanumeric characters (order matters):

```
fcrackzip {{[-b|--brute-force]}} {{[-l|--length]}} 4-8 {{[-c|--charset]}} aA1 {{archive}}
```

- Brute-force a password in verbose mode with a length of 3 characters that only contains lowercase characters, `$` and `%`:

```
fcrackzip {{[-v|--verbose]}} {{[-b|--brute-force]}} {{[-l|--length]}} 3 {{[-c|--charset]}} a:$% {{archive}}
```

- Brute-force a password that contains only lowercase and special characters:

```
fcrackzip {{[-b|--brute-force]}} {{[-l|--length]}} 4 {{[-c|--charset]}} a! {{archive}}
```

- Brute-force a password containing only digits, starting from the password `12345`:

```
fcrackzip {{[-b|--brute-force]}} {{[-l|--length]}} 5 {{[-c|--charset]}} 1 {{[-p|--init-password]}} 12345 {{archive}}
```

- Crack a password using a wordlist:

```
fcrackzip {{[-u|--use-unzip]}} {{[-D|--dictionary]}} {{[-p|--init-password]}} {{wordlist}} {{archive}}
```

- Benchmark cracking performance:

```
fcrackzip {{[-B|--benchmark]}}
```

fdisk

Manage partition tables and partitions on a storage drive.

See also: **`partprobe`**, **`parted`**, **`cfdisk`**.

More information: https://manned.org/fdisk.

- List partitions:

```
sudo fdisk {{[-l|--list]}}
```

- Start the interactive partition manipulator:

```
sudo fdisk {{/dev/sdX}}
```

- Open a help [m]enu:

```
<m>
```

- View the [p]artition table:

```
<p>
```

- Create a [n]ew partition:

```
<n>
```

- Select a partition to [d]elete:

```
<d>
```

- [w]rite the changes made:

```
<w>
```

- Discard the changes made and [q]uit:

```
<q>
```

feedreader

A GUI desktop RSS client.

Note: FeedReader is no longer being maintained.

More information: https://github.com/jangernert/FeedReader.

- Print the count of unread articles:

  ```
  feedreader --unreadCount
  ```

- Add a URL for a feed to follow:

  ```
  feedreader --addFeed={{feed_url}}
  ```

- Grab a specific article using its URL:

  ```
  feedreader --grabArticle={{article_url}}
  ```

- Download all images from a specific article:

  ```
  feedreader --url={{feed_url}} --grabImages={{article_path}}
  ```

- Play media from a URL:

  ```
  feedreader --playMedia={{article_url}}
  ```

fftest

Test force-feedback devices (e.g., controllers with rumble functionality).

More information: https://manned.org/fftest.

- Open `fftest` on an event:

```
fftest {{/dev/input/eventX}}
```

fgconsole

Print the number of the foreground virtual terminal.

More information: https://manned.org/fgconsole.

- Print current terminal number or `serial` if the terminal is serial:

  `fgconsole`

- Print the next unallocated virtual terminal:

  `fgconsole {{[-n|--next-available]}}`

filefrag

Report how badly fragmented a particular file might be.

More information: https://manned.org/filefrag.

- Display a report for one or more files:

```
filefrag {{path/to/file1 path/to/file2 ...}}
```

- Display a report using a 1024 byte blocksize:

```
filefrag -k {{path/to/file}}
```

- Display a report using a certain blocksize:

```
filefrag -b{{1024|1K|1M|1G|...}} {{path/to/file}}
```

- Sync the file before requesting the mapping:

```
filefrag -s {{path/to/file1 path/to/file2 ...}}
```

- Display mapping of extended attributes:

```
filefrag -x {{path/to/file1 path/to/file2 ...}}
```

- Display a report with verbose information:

```
filefrag -v {{path/to/file1 path/to/file2 ...}}
```

finch

Console-based modular messaging client.

More information: https://developer.pidgin.im/wiki/Using%20Finch.

- Launch finch:

  `finch`

- Quit:

  `{{<Alt q>|<Ctrl c>}}`

- Show actions menu:

  `<Alt a>`

- Jump to n-th window:

  `<Alt {{Number}}>`

- Close current window:

  `<Alt c>`

- Start moving a window, use `<ArrowKeys>` to move, press `<Esc>` when done:

  `<Alt m>`

- Start resizing a window, use `<ArrowKeys>` to resize, press `<Esc>` when done:

  `<Alt r>`

fincore

Display how much cache memory a file is taking.

See also: **`fadvise`**.

More information: https://manned.org/fincore.

- Display cache details for a file:

```
fincore {{path/to/file}}
```

- Display all possible data columns:

```
fincore --output-all {{path/to/file}}
```

- Display help:

```
fincore {{[-h|--help]}}
```

findfs

Finds a filesystem by label or UUID.

More information: https://manned.org/findfs.

- Search block devices by filesystem label:

```
findfs LABEL={{label}}
```

- Search by filesystem UUID:

```
findfs UUID={{uuid}}
```

- Search by partition label (GPT or MAC partition table):

```
findfs PARTLABEL={{partition_label}}
```

- Search by partition UUID (GPT partition table only):

```
findfs PARTUUID={{partition_uuid}}
```

findmnt

Find your filesystem.

More information: https://manned.org/findmnt.

- List all mounted filesystems:

```
findmnt
```

- Search for a device:

```
findmnt {{/dev/sdb1}}
```

- Search for a mountpoint:

```
findmnt {{/}}
```

- Find filesystems in specific type:

```
findmnt {{[-t|--types]}} {{ext4,virtiofs,btrfs,...}}
```

- Find filesystems with specific label:

```
findmnt LABEL={{BigStorage}}
```

- Check mount table content in detail and verify `/etc/fstab`:

```
findmnt {{[-x|--verify]}} --verbose
```

firejail

Securely sandboxes processes to containers using built-in Linux capabilities.

More information: https://manned.org/firejail.

- Integrate firejail with your desktop environment:

```
sudo firecfg
```

- Open a restricted Mozilla Firefox:

```
firejail {{firefox}}
```

- Start a restricted Apache server on a known interface and address:

```
firejail --net={{eth0}} --ip={{192.168.1.244}} {{/etc/init.d/apache2}} {{start}}
```

- List running sandboxes:

```
firejail --list
```

- List network activity from running sandboxes:

```
firejail --netstats
```

- Shutdown a running sandbox:

```
firejail --shutdown={{7777}}
```

- Run a restricted Firefox session to browse the internet:

```
firejail --seccomp --private --private-dev --private-tmp --protocol=inet firefox --new-instance --no-remote --safe-mode --private-window
```

- Use custom hosts file (overriding `/etc/hosts` file):

```
firejail --hosts-file={{~/myhosts}} {{curl http://mysite.arpa}}
```

firewall-cmd

The firewalld command-line client.

View and adapt the runtime or permanent firewall configuration state.

More information: https://firewalld.org/documentation/man-pages/firewall-cmd.

- View all available firewall zones and rules in their runtime configuration state:

```
firewall-cmd --list-all-zones
```

- Permanently move the interface into the block zone, effectively blocking all communication:

```
firewall-cmd --permanent --zone {{block}} --change-interface {{enp1s0}}
```

- Permanently open the port for a service in the specified zone (like port 443 when in the `public` zone):

```
firewall-cmd --permanent --zone {{public}} --add-service {{https}}
```

- Permanently close the port for a service in the specified zone (like port 80 when in the `public` zone):

```
firewall-cmd --permanent --zone {{public}} --remove-service {{http}}
```

- Permanently forward a port for incoming packets in the specified zone (like port 443 to 8443 when entering the `public` zone):

```
firewall-cmd --permanent --zone {{public}} --add-rich-rule 'rule family "{{ipv4|ipv6}}" forward-port port "{{443}}" protocol "{{udp|tcp}}" to-port "{{8443}}"'
```

- Reload firewalld to lose any runtime changes and force the permanent configuration to take effect immediately:

```
firewall-cmd --reload
```

- Save the runtime configuration state to the permanent configuration:

```
firewall-cmd --runtime-to-permanent
```

- Enable panic mode in case of Emergency. All traffic is dropped, any active connection will be terminated:

```
firewall-cmd --panic-on
```

fixfiles

Fix file SELinux security contexts.

More information: https://manned.org/fixfiles.

- If specified with onboot, this fixfiles will record the current date in the `/.autorelabel` file, so that it can be used later to speed up labeling. If used with restore, the restore will only affect files that were modified today:

```
fixfiles -B
```

- [F]orce reset of context to match `file_context` for customizable files:

```
fixfiles -F
```

- Clear `/tmp` directory without confirmation:

```
fixfiles -f
```

- Use the [R]pm database to discover all files within specific packages and restore the file contexts:

```
fixfiles -R {{rpm_package1,rpm_package2 ...}}
```

- Run a diff on the `PREVIOUS_FILECONTEXT` file to the [C]urrently installed one, and restore the context of all affected files:

```
fixfiles -C PREVIOUS_FILECONTEXT
```

- Only act on files created after a specific date which will be passed to find `--newermt` command:

```
fixfiles -N {{YYYY-MM-DD HH:MM}}
```

- Bind [M]ount filesystems before relabeling them, this allows fixing the context of files or directories that have been mounted over:

```
fixfiles -M
```

- Modify [v]erbosity from progress to verbose and run `restorecon` with `-v` instead of `-p`:

```
fixfiles -v
```

flameshot

Screenshot utility with a GUI.

Supports basic image editing, such as text, shapes, colors, and imgur.

More information: https://flameshot.org/docs/advanced/commandline-options/.

- Create a fullscreen screenshot:

```
flameshot full
```

- Create a screenshot interactively:

```
flameshot gui
```

- Create a screenshot and save it to a specific path:

```
flameshot gui {{[-p|--path]}} {{path/to/directory}}
```

- Create a screenshot interactively in a simplified mode:

```
flameshot launcher
```

- Create a screenshot from a specific monitor:

```
flameshot screen {{[-n|--number]}} {{2}}
```

- Create a screenshot and print it to `stdout`:

```
flameshot gui {{[-r|--raw]}}
```

- Create a screenshot and copy it to the clipboard:

```
flameshot gui {{[-c|--clipboard]}}
```

- Create a screenshot with a specific delay in milliseconds:

```
flameshot full {{[-d|--delay]}} {{5000}}
```

flash

Flash cards in the terminal.

More information: https://github.com/tallguyjenks/fla.sh.

- Open a menu of available flashcard decks for selection:

```
flash
```

- Display information about the flashcard system:

```
flash -i
```

- Change the previewer from default `bat` to `cat`:

```
flash -p {{cat}}
```

- Display help:

```
flash -h
```

- Display version:

```
flash -v
```

flashrom

Read, write, verify and erase flash chips.

More information: https://manned.org/flashrom.

- Probe the chip, ensuring the wiring is correct:

```
flashrom {{[-p|--programmer]}} {{programmer}}
```

- Read flash and save it to a file:

```
flashrom {{[-p|--programmer]}} {{programmer}} {{[-r|--read]}}
{{path/to/file}}
```

- Write a file to the flash:

```
flashrom {{[-p|--programmer]}} {{programmer}} {{[-w|--
write]}} {{path/to/file}}
```

- Verify the flash against a file:

```
flashrom {{[-p|--programmer]}} {{programmer}} {{[-v|--
verify]}} {{path/to/file}}
```

- Probe the chip using Raspberry Pi:

```
flashrom {{[-p|--programmer]}} {{linux_spi:dev=/dev/
spidev0.0}}
```

flatpak-builder

Help build application dependencies.

More information: https://docs.flatpak.org/en/latest/flatpak-builder-command-reference.html.

- Build a Flatpak and export it to a new repository:

```
flatpak-builder {{path/to/build_directory}} {{path/to/manifest}}
```

- Build a Flatpak and export it to the specified repository:

```
flatpak-builder --repo {{repository_name}} {{path/to/build_directory}} {{path/to/manifest}}
```

- Build a Flatpak and install it locally:

```
flatpak-builder --install {{path/to/build_directory}} {{path/to/manifest}}
```

- Build and sign a Flatpak and export it to the specified repository:

```
flatpak-builder --gpg-sign {{key_id}} --repo {{repository_name}} {{path/to/manifest}}
```

- Run a shell inside of an application sandbox without installing it:

```
flatpak-builder --run {{path/to/build_directory}} {{path/to/manifest}} {{sh}}
```

flatpak mask

Mask out updates and automatic installation.

More information: https://docs.flatpak.org/en/latest/flatpak-command-reference.html#flatpak-mask.

- Ignore updates for a specific flatpak:

```
flatpak mask {{com.example.app}}
```

- Undo ignore updates:

```
flatpak mask --remove {{com.example.app}}
```

- List all currently masked patterns:

```
flatpak mask {{--system|--user}}
```

flatpak remote-info

Show information about an application or runtime in a remote.

More information: https://docs.flatpak.org/en/latest/flatpak-command-reference.html#flatpak-remote-info.

- Show information about a flatpak:

```
flatpak remote-info {{remote_name}} {{com.example.app}}
```

- Show a log of previous versions in a remote:

```
flatpak remote-info --log {{remote_name}} {{com.example.app}}
```

- Show information about the specific commit, rather than the latest version:

```
flatpak remote-info --commit {{COMMIT}} {{remote_name}}
{{com.example.app}}
```

flatpak run

Run flatpak applications and runtimes.

More information: https://docs.flatpak.org/en/latest/flatpak-command-reference.html#flatpak-run.

- Run an installed application:

```
flatpak run {{com.example.app}}
```

- Run an installed application from a specific branch e.g. stable, beta, master:

```
flatpak run --branch={{stable|beta|master|...}}
{{com.example.app}}
```

- Run an interactive shell inside a flatpak:

```
flatpak run --command={{sh}} {{com.example.app}}
```

- Run an installed application with a specific runtime version:

```
flatpak run --runtime-version={{24.08|master|stable|...}}
{{com.example.app}}
```

- Run an installed application with a different runtime (but same version number):

```
flatpak run --runtime={{org.freedesktop.Sdk}}
{{com.example.app}}
```

flatpak update

Update flatpak applications and runtimes.

More information: https://docs.flatpak.org/en/latest/flatpak-command-reference.html#flatpak-update.

- Update all installed applications and runtimes (use `-y` to confirm all prompts automatically):

```
flatpak update
```

- Update only a specific app:

```
flatpak update {{com.example.app}}
```

- Update/Downgrade to a specific commit (also see flatpak remote-info and flatpak mask):

```
flatpak update --commit {{COMMIT}} {{com.example.app}}
```

flatpak

Build, install and run flatpak applications and runtimes.

More information: https://docs.flatpak.org/en/latest/flatpak-command-reference.html#flatpak.

- Run an installed application:

```
flatpak run {{com.example.app}}
```

- Install an application from a remote source:

```
flatpak install {{remote_name}} {{com.example.app}}
```

- List installed applications, ignoring runtimes:

```
flatpak list --app
```

- Update all installed applications and runtimes:

```
flatpak update
```

- Add a remote source:

```
flatpak remote-add --if-not-exists {{remote_name}}
{{remote_url}}
```

- Remove an installed application:

```
flatpak remove {{com.example.app}}
```

- Remove all unused applications:

```
flatpak remove --unused
```

- Show information about an installed application:

```
flatpak info {{com.example.app}}
```

flex

Lexical analyzer generator.

Given the specification for a lexical analyzer, generates C code implementing it.

More information: https://manned.org/lex.1.

- Generate an analyzer from a Lex file, storing it to the file `lex.yy.c`:

```
flex {{analyzer.l}}
```

- Write analyzer to `stdout`:

```
flex {{[-t|--stdout]}} {{analyzer.l}}
```

- Specify the output file:

```
flex {{analyzer.l}} {{[-o|--outfile]}} {{analyzer.c}}
```

- Generate a batch scanner instead of an interactive scanner:

```
flex {{[-B|--batch]}} {{analyzer.l}}
```

- Compile a C file generated by Lex:

```
cc {{path/to/lex.yy.c}} -o {{executable}}
```

flock

Manage file locks from shell scripts.

It can be used to ensure that only one instance of a command is running.

More information: https://manned.org/flock.

- Run a command with a file lock as soon as the lock is available:

```
flock {{path/to/lock.lock}} {{command}}
```

- Run a command with a file lock, or exit if the lock is currently being held (with exit code 1):

```
flock {{path/to/lock.lock}} {{[-n|--nonblock]}} {{command}}
```

- Run a command with a file lock, or exit with a specific error code if the lock is currently being held:

```
flock {{path/to/lock.lock}} {{[-n|--nonblock]}} {{[-E|--conflict-exit-code]}} {{123}} {{command}}
```

- Run a command with a file lock, waiting up to 10 seconds for the lock to be available before giving up:

```
flock {{path/to/lock.lock}} {{[-w|--timeout]}} 10 {{command}}
```

- Backup a bunch of files, waiting for the previous `tar` command to finish if it's still running elsewhere and holding the same lock file (can be used in a `cron` job that runs often):

```
flock {{path/to/backup.lock}} {{tar -cvf path/to/backup.tar path/to/data/}}
```

fluidsynth

Synthesize audio from MIDI files.

More information: https://github.com/FluidSynth/fluidsynth/wiki/UserManual.

- Play a MIDI file:

```
fluidsynth {{path/to/soundfont.sf2}} {{path/to/file.midi}}
```

- Specify the audio driver:

```
fluidsynth {{[-a|--audio-driver]}} {{pipewire|pulseaudio}}
{{path/to/soundfont.sf2}} {{path/to/file.midi}}
```

- Display help:

```
fluidsynth {{[-h|--help]}}
```

fold

Folds long lines for fixed-width output devices.

More information: https://www.gnu.org/software/coreutils/manual/html_node/fold-invocation.html.

- Fold lines in a fixed width:

```
fold {{[-w|--width]}} {{width}} {{path/to/file}}
```

- Count width in bytes (the default is to count in columns):

```
fold {{[-b|--bytes]}} {{[-w|--width]}} {{width_in_bytes}} {{path/to/file}}
```

- Break the line after the rightmost blank within the width limit:

```
fold {{[-s|--spaces]}} {{[-w|--width]}} {{width}} {{path/to/file}}
```

fonttools subset

Generate subsets of fonts or optimize file sizes.

More information: https://fonttools.readthedocs.io/en/latest/subset/index.html.

- Subset a TTF font file to the Basic Latin Unicode block:

```
fonttools subset {{path/to/font.ttf}} --unicodes=U+0000-007F
```

- Change the file type to WOFF2:

```
fonttools subset {{path/to/font.ttf}} --unicodes=U+0000-007F
--flavor=woff2
```

- Keep only the onum (oldstyle figures) and kern (kerning) OpenType font features:

```
fonttools subset {{path/to/font.ttf}} --unicodes=U+0000-007F
--layout-features=onum,kern
```

- Set the output file's name:

```
fonttools subset {{path/to/font.ttf}} --unicodes=U+0000-007F
--output-file={{path/to/subset.ttf}}
```

fonttools

Manipulate fonts in Python.

More information: https://fonttools.readthedocs.io/en/latest/.

- Subset a TTF font file to the Basic Latin Unicode block:

```
fonttools subset {{path/to/font.ttf}} --unicodes=U+0000-007F
```

- Display help:

```
fonttools --help
```

foreman

Manage Procfile-based applications.

More information: https://manned.org/foreman.

- Start an application with the Procfile in the current directory:

```
foreman start
```

- Start an application with a specified Procfile:

```
foreman start {{[-f|--procfile]}} {{Procfile}}
```

- Start a specific application:

```
foreman start {{process}}
```

- Validate Procfile format:

```
foreman check
```

- Run one-off commands with the process's environment:

```
foreman run {{command}}
```

- Start all processes except the one named "worker":

```
foreman start {{[-m|--formation]}} all=1,{{worker}}=0
```

fprintd-delete

Remove fingerprints from the database.

More information: https://manned.org/fprintd-delete.

- Remove all fingerprints for a specific user:

```
fprintd-delete {{username}}
```

- Remove a specific fingerprints for a specific user:

```
fprintd-delete {{username}} {{[-f|--finger]}} {{left-thumb|left-index-finger|left-middle-finger|left-ring-finger|left-little-finger|right-thumb|...}}
```

- Display help:

```
fprintd-delete
```

fprintd-enroll

Enroll fingerprints into the database.

More information: https://manned.org/fprintd-enroll.

- Enroll the right index finger for the current user:

```
fprintd-enroll
```

- Enroll a specific finger for the current user:

```
fprintd-enroll {{[-f|--finger]}} {{left-thumb|left-index-finger|left-middle-finger|left-ring-finger|left-little-finger|right-thumb|...}}
```

- Enroll the right index finger for a specific user:

```
fprintd-enroll {{username}}
```

- Enroll a specific finger for a specific user:

```
fprintd-enroll {{[-f|--finger]}} {{finger_name}} {{username}}
```

- Display help:

```
fprintd-enroll --help
```

fprintd-list

List enrolled fingerprints.

More information: https://manned.org/fprintd-list.

- List enrolled fingerprints for a specific user:

```
fprintd-list {{username}}
```

- List enrolled fingerprints for one or more users:

```
fprintd-list {{username1 username2 ...}}
```

- Display help:

```
fprintd-list
```

fprintd-verify

Verify fingerprints against the database.

More information: https://manned.org/fprintd-verify.

- Verify all stored fingerprints for the current user:

```
fprintd-verify
```

- Verify a specific fingerprint for the current user:

```
fprintd-verify {{[-f|--finger]}} {{left-thumb|left-index-finger|left-middle-finger|left-ring-finger|left-little-finger|right-thumb|...}}
```

- Verify fingerprints for a specific user:

```
fprintd-verify {{username}}
```

- Verify a specific fingerprint for a specific user:

```
fprintd-verify {{[-f|--finger]}} {{finger_name}} {{username}}
```

- Fail the process if a fingerprint doesn't match with ones stored in the database for the current user:

```
fprintd-verify --g-fatal-warnings
```

- Display help:

```
fprintd-verify {{[-h|--help]}}
```

fprintd

Fingerprint management daemon.

fprintd isn't a command, but a set of commands.

More information: https://fprint.freedesktop.org/.

- View documentation for listing enrolled fingerprints:

```
tldr fprintd-list
```

- View documentation for enrolling new fingerprints:

```
tldr fprintd-enroll
```

- View documentation for verifying a fingerprint against a database:

```
tldr fprintd-verify
```

- View documentation for deleting fingerprints from a database:

```
tldr fprintd-delete
```

free

Display amount of free and used memory in the system.

More information: https://manned.org/free.

- Display system memory:

```
free
```

- Display memory in Bytes/KB/MB/GB:

```
free -{{b|k|m|g}}
```

- Display memory in human-readable units:

```
free {{[-h|--human]}}
```

- Refresh the output every 2 seconds:

```
free {{[-s|--seconds]}} 2
```

fsadm

Check or resize a filesystem on a device.

More information: https://manned.org/fsadm.

- Check a filesystem for errors:

```
fsadm check {{/dev/vg_name/lv_name}}
```

- Perform a dry-run resize to a specific size (no changes made):

```
fsadm {{[-n|--dry-run]}} resize {{/dev/vg_name/lv_name}} {{10G}}
```

- Grow a filesystem to fill the entire device (omit the size):

```
fsadm resize {{/dev/vg_name/lv_name}}
```

- Resize the filesystem and the underlying logical volume together:

```
fsadm {{[-l|--lvresize]}} resize {{/dev/vg_name/lv_name}} {{100G}}
```

- For ext2/3/4, unmount and resize offline:

```
fsadm {{[-e|--ext-offline]}} resize {{/dev/vg_name/lv_name}} {{20G}}
```

fsck

Check the integrity of a filesystem or repair it. The filesystem should be unmounted at the time the command is run.

More information: https://manned.org/fsck.

- Check filesystem `/dev/sdXN`, reporting any damaged blocks:

  ```
  sudo fsck {{/dev/sdXN}}
  ```

- Check filesystem `/dev/sdXN`, reporting any damaged blocks and interactively letting the user choose to repair each one:

  ```
  sudo fsck -r {{/dev/sdXN}}
  ```

- Check filesystem `/dev/sdXN`, reporting any damaged blocks and [a]utomatically repairing them:

  ```
  sudo fsck -a {{/dev/sdXN}}
  ```

fscrypt

Go tool for managing Linux filesystem encryption.

More information: https://github.com/google/fscrypt.

- Prepare the root filesystem for use with `fscrypt`:

```
sudo fscrypt setup
```

- Prepare a specific mountpoint for use with `fscrypt`:

```
fscrypt setup {{path/to/directory}}
```

- Enable filesystem encryption for a directory:

```
fscrypt encrypt {{path/to/directory}}
```

- Unlock an encrypted directory:

```
fscrypt unlock {{path/to/encrypted_directory}}
```

- Lock an encrypted directory:

```
fscrypt lock {{path/to/encrypted_directory}}
```

fstrim

Discard unused blocks on a mounted filesystem.

Only supported by flash memory devices such as SSDs and microSD cards.

More information: https://manned.org/fstrim.

- Trim unused blocks on all mounted partitions that support it:

```
sudo fstrim {{[-a|--all]}}
```

- Trim unused blocks on a specified partition:

```
sudo fstrim {{/}}
```

- Display statistics after trimming:

```
sudo fstrim {{[-v|--verbose]}} {{/}}
```

fuser

Display process IDs currently using files or sockets.

More information: https://manned.org/fuser.

- Find which processes are accessing a file or directory:

```
fuser {{path/to/file_or_directory}}
```

- Show more fields (`USER`, `PID`, `ACCESS` and `COMMAND`):

```
fuser {{[-v|--verbose]}} {{path/to/file_or_directory}}
```

- Identify processes using a TCP socket:

```
fuser {{[-n|--namespace]}} tcp {{port}}
```

- Kill all processes accessing a file or directory (sends the `SIGKILL` signal):

```
fuser {{[-k|--kill]}} {{path/to/file_or_directory}}
```

- Find which processes are accessing the filesystem containing a specific file or directory:

```
fuser {{[-m|--mount]}} {{path/to/file_or_directory}}
```

- Kill all processes with a TCP connection on a specific port:

```
fuser {{[-k|--kill]}} {{port}}/tcp
```

fuzzel

A Wayland-native application launcher and fuzzy finder, inspired by **`rofi`** and **`dmenu`**.

More information: https://codeberg.org/dnkl/fuzzel.

- Run applications:

```
fuzzel
```

- Run `fuzzel` in dmenu mode:

```
fuzzel {{[-d|--dmenu]}}
```

- Display a menu of the output of the `ls` command:

```
{{ls}} | fuzzel {{[-d|--dmenu]}}
```

- Display a menu with custom items separated by a new line (`\n`):

```
echo -e "{{red}}\n{{green}}\n{{blue}}" | fuzzel {{[-d|--dmenu]}}
```

- Let the user choose between multiple items and save the selected one to a file:

```
echo -e "{{red}}\n{{green}}\n{{blue}}" | fuzzel {{[-d|--dmenu]}} > {{color.txt}}
```

- Reset apps usage count (default cache directory: `$XDG_CACHE_HOME/fuzzel`):

```
rm {{[-v|--verbose]}} $HOME/.cache/fuzzel
```

- Launch `fuzzel` on a specific monitor, see `wlr-randr` or `swaymsg --type get_outputs`:

```
fuzzel {{[-o|--output]}} "{{DP-1}}"
```

- Use `fuzzel` to do an online search:

```
fuzzel {{[-d|--dmenu]}} {{[-l|--lines]}} 0 --placeholder "{{Type your search}}" | sed 's/^/\"/g;s/$/\"/g' | xargs firefox --search
```

fwconsole

Manage and configure your FreePBX system (PBX server).

More information: https://sangomakb.atlassian.net/wiki/spaces/PG/pages/41779247/fwconsole+commands+13.

- Reload FreePBX configurations:

  ```
  fwconsole reload
  ```

- Start Asterisk and other commands needed by FreePBX:

  ```
  fwconsole start
  ```

- Stop Asterisk and other commands needed by FreePBX:

  ```
  fwconsole stop
  ```

- View and update settings:

  ```
  fwconsole setting {{keyword}} {{new_value}}
  ```

- List available backups:

  ```
  fwconsole backup --list
  ```

- List available FreePBX commands:

  ```
  fwconsole list
  ```

- Change ownership of all files and directories that FreePBX needs to be owned by the apache user:

  ```
  fwconsole chown
  ```

fwupdmgr

Update device firmware, including UEFI, using **`fwupd`**.

See also: **`fwupdtool`**.

More information: https://github.com/fwupd/fwupd/blob/main/src/fwupdmgr.md.

- Display all devices detected by `fwupd`:

```
fwupdmgr get-devices
```

- Download the latest firmware metadata from LVFS:

```
fwupdmgr refresh
```

- List the updates available for devices on your system:

```
fwupdmgr get-updates
```

- Install firmware updates:

```
fwupdmgr update
```

- Remount `/boot` with more privileges if update complains about a read-only filesystem:

```
sudo mount {{[-o|--options]}} uid=1000,gid=1000,umask=0022 {{/dev/sdX}} /boot
```

- Show firmware update history:

```
fwupdmgr get-history
```

fwupdtool

Update device firmware manually or manipulate firmware files.

See also: **`fwupdmgr`**.

More information: https://github.com/fwupd/fwupd/blob/main/src/fwupdtool.md.

- Display all devices detected by `fwupd`:

```
fwupdtool get-devices
```

- Install firmware from a file:

```
fwupdtool install {{path/to/firmware}}
```

- Display help:

```
fwupdtool {{[-h|--help]}}
```

gamescope

A micro-compositor used as a game layer.

See also: **cage**.

More information: https://github.com/ValveSoftware/gamescope.

- Run a program with gamescope on the terminal:

```
gamescope -- {{program}}
```

- Run a game with gamescope through Steam:

```
gamescope -- %command%
```

- Upscale a 720p game to 1440p with integer scaling:

```
gamescope {{[-h|--nested-height]}} 720 {{[-H|--output-height]}} 1440 {{[-S|--scaler]}} integer -- %command%
```

- Limit a vsynced game to 30 FPS:

```
gamescope {{[-r|--nested-refresh]}} 30 -- %command%
```

- Launch Steam in Big Picture Mode and integrate with gamescope:

```
gamescope {{[-e|--steam]}} -- /usr/bin/steam -tenfoot
```

- Specify which display to prefer:

```
gamescope {{[-O|--prefer-output]}} {{HDMI-A-1,DP-3,...}} -- {{program}}
```

- Toggle fullscreen:

```
<Super f>
```

- Display help:

```
gamescope --help
```

gbp

A system to integrate the Debian package build system with Git.

More information: https://honk.sigxcpu.org/projects/git-buildpackage/manual-html/gbp.html.

- Convert an existing Debian package to gbp:

```
gbp import-dsc {{path/to/package.dsc}}
```

- Build the package in the current directory using the default builder (`debuild`):

```
gbp buildpackage -jauto -us -uc
```

- Build a package in a `pbuilder` environment for Debian Bullseye:

```
DIST={{bullseye}} ARCH={{amd64}} gbp buildpackage -jauto -us -uc --git-builder={{git-pbuilder}}
```

- Specify a package to be a source-only upload in the `.changes` file (see https://wiki.debian.org/SourceOnlyUpload):

```
gbp buildpackage -jauto -us -uc --changes-options={{-S}}
```

- Import a new upstream release:

```
gbp import-orig --pristine-tar {{path/to/package.tar.gz}}
```

gcov

Code coverage analysis and profiling tool that discovers untested parts of a program.

Also displays a copy of source code annotated with execution frequencies of code segments.

More information: https://gcc.gnu.org/onlinedocs/gcc/Invoking-Gcov.html.

- Generate a coverage report named `file.cpp.gcov`:

```
gcov {{path/to/file.cpp}}
```

- Write individual execution counts for every basic block:

```
gcov {{[-a|--all-blocks]}} {{path/to/file.cpp}}
```

- Write branch frequencies to the output file and print summary information to `stdout` as a percentage:

```
gcov {{[-b|--branch-probabilities]}} {{path/to/file.cpp}}
```

- Write branch frequencies as the number of branches taken, rather than the percentage:

```
gcov {{[-c|--branch-counts]}} {{path/to/file.cpp}}
```

- Do not create a `gcov` output file:

```
gcov {{[-n|--no-output]}} {{path/to/file.cpp}}
```

- Write file level as well as function level summaries:

```
gcov {{[-f|--function-summaries]}} {{path/to/file.cpp}}
```

gcrane completion

Generate the autocompletion script for gcrane for the specified shell.

The available shells are Bash, fish, PowerShell, and Zsh.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/gcrane/README.md.

- Generate the autocompletion script for your shell:

```
gcrane completion {{shell_name}}
```

- Disable completion descriptions:

```
gcrane completion {{shell_name}} --no-descriptions
```

- Load completions in your current shell session (Bash/Zsh):

```
source <(gcrane completion bash/zsh)
```

- Load completions in your current shell session (fish):

```
gcrane completion fish | source
```

- Load completions for every new session (Bash):

```
gcrane completion bash > /etc/bash_completion.d/gcrane
```

- Load completions for every new session (Zsh):

```
gcrane completion zsh > "${fpath[1]}/_gcrane"
```

- Load completions for every new session (fish):

```
gcrane completion fish > ~/.config/fish/completions/gcrane.fish
```

- Display help:

```
gcrane completion {{shell_name}} {{[-h|--help]}}
```

gdbus

Interact with D-Bus objects.

Part of GLib.

More information: https://manned.org/gdbus.

- List all names on the session bus:

```
gdbus list-names --session
```

- List all names on the system bus:

```
gdbus list-names --system
```

- Introspect an object to see its interfaces and methods:

```
gdbus introspect --session --dest {{destination_bus_name}} --
object-path /{{path/to/object}}
```

- Call a method on an object with arguments:

```
gdbus call --session --dest {{destination_bus_name}} --
object-path /{{path/to/object}} --method
{{interface.method_name}} {{argument1 argument2 ...}}
```

- Emit a signal from an object with arguments:

```
gdbus emit --session --object-path /{{path/to/object}} --
signal {{interface.signal_name}} {{argument1 argument2 ...}}
```

- Monitor all messages on the session bus:

```
gdbus monitor --session
```

gdebi

Easily install **`.deb`** files.

More information: https://manned.org/gdebi.

- Install local `.deb` packages resolving and installing its dependencies:

```
gdebi {{path/to/package.deb}}
```

- Do not show progress information:

```
gdebi {{path/to/package.deb}} {{[-q|--quiet]}}
```

- Set an APT configuration option:

```
gdebi {{path/to/package.deb}} {{[-o|--option]}} {{APT_OPTS}}
```

- Use alternative root dir:

```
gdebi {{path/to/package.deb}} --root {{path/to/
root_directory}}
```

- Display version:

```
gdebi --version
```

gedit

Text editor of the GNOME Desktop project.

More information: https://help.gnome.org/users/gedit/stable/.

- Open a text file:

```
gedit {{path/to/file}}
```

- Open multiple text files:

```
gedit {{file1 file2 ...}}
```

- Open a text file with a specific encoding:

```
gedit --encoding {{UTF-8}} {{path/to/file}}
```

- Display a list of supported encodings:

```
gedit --list-encodings
```

genfstab

Generate output suitable for addition to the **`/etc/fstab`** file.

More information: https://manned.org/genfstab.

- Generate the `/etc/fstab` file using volume UUIDs during an Arch Linux installation (requires root permissions):

```
genfstab -U {{/mnt}} >> {{/mnt/etc/fstab}}
```

- Display fstab-compatible output based on volume labels:

```
genfstab -L {{path/to/mount_point}}
```

- Display fstab-compatible output based on volume UUIDs:

```
genfstab -U {{path/to/mount_point}}
```

- Display fstab-compatible output based on the specified identifier:

```
genfstab -t {{LABEL|UUID|PARTLABEL|PARTUUID}}
```

- Append a volume into the `/etc/fstab` file to mount it automatically:

```
genfstab -U {{path/to/mount_point}} | sudo tee -a /etc/fstab
```

genie

Set up and use a "bottle" namespace to run systemd under WSL (Windows Subsystem for Linux).

To run these from Windows rather than an already-running distribution, precede them with **`wsl`**.

More information: https://github.com/arkane-systems/genie.

- Initialize the bottle (run once, at start):

```
genie {{[-i|--initialize]}}
```

- Run a login shell inside the bottle:

```
genie {{[-s|--shell]}}
```

- Run a specified command inside the bottle:

```
genie {{[-c|--command]}} {{command}}
```

genisoimage

Pre-mastering program to generate ISO9660/Joliet/HFS hybrid filesystems.

More information: https://manned.org/genisoimage.1.

- Create an ISO image from the given source directory:

```
genisoimage -o {{myimage.iso}} {{path/to/source_directory}}
```

- Create an ISO image with files larger than 2GiB by reporting a smaller apparent size for ISO9660 filesystems:

```
genisoimage -o -allow-limited-size {{myimage.iso}} {{path/to/source_directory}}
```

genkernel

Gentoo Linux utility to compile and install kernels.

More information: https://wiki.gentoo.org/wiki/Genkernel.

- Automatically compile and install a generic kernel:

```
sudo genkernel all
```

- Build and install the bzImage|initramfs|kernel|ramdisk only:

```
sudo genkernel {{bzImage|initramfs|kernel|ramdisk}}
```

- Apply changes to the kernel configuration before compiling and installing:

```
sudo genkernel --menuconfig all
```

- Generate a kernel with a custom name:

```
sudo genkernel --kernname={{custom_name}} all
```

- Use a kernel source outside the default directory `/usr/src/linux`:

```
sudo genkernel --kerneldir={{path/to/directory}} all
```

getcap

Display the name and capabilities of each specified file.

More information: https://manned.org/getcap.

- Get capabilities for the given files:

```
getcap {{path/to/file1 path/to/file2 ...}}
```

- Get capabilities for all the files recursively under the given directories:

```
getcap -r {{path/to/directory1 path/to/directory2 ...}}
```

- Display all searched entries even if no capabilities are set:

```
getcap -v {{path/to/file1 path/to/file2 ...}}
```

getconf

Get configuration values from your Linux system.

More information: https://manned.org/getconf.1.

- List [a]ll configuration values available:

```
getconf -a
```

- List the configuration values for a specific directory:

```
getconf -a {{path/to/directory}}
```

- Check if the system is 32-bit or 64-bit:

```
getconf LONG_BIT
```

- Check how many processes the current user can run at once:

```
getconf CHILD_MAX
```

- List every configuration value and then find patterns with the `grep` command (i.e every value with MAX in it):

```
getconf -a | grep MAX
```

getenforce

Get the current mode of SELinux (i.e. enforcing, permissive, or disabled).

See also: **`setenforce`**, **`semanage-permissive`**.

More information: https://manned.org/getenforce.

- Display the current mode of SELinux:

```
getenforce
```

getent

Get entries from Name Service Switch libraries.

More information: https://manned.org/getent.

- Get list of all groups:

```
getent group
```

- See the members of a group:

```
getent group {{group_name}}
```

- Get list of all services:

```
getent services
```

- Find a username by UID:

```
getent passwd 1000
```

- Perform a reverse DNS lookup:

```
getent hosts {{host}}
```

getfacl

Get file access control lists (ACL).

More information: https://manned.org/getfacl.

- Display the file access control list:

```
getfacl {{path/to/file_or_directory}}
```

- Display the file access control list with numeric user and group IDs:

```
getfacl {{[-n|--numeric]}} {{path/to/file_or_directory}}
```

- Display the file access control list with tabular output format:

```
getfacl {{[-t|--tabular]}} {{path/to/file_or_directory}}
```

getfattr

Display file names and extended attributes.

More information: https://manned.org/getfattr.

- Retrieve all extended attributes of a file and display them in a detailed format:

```
getfattr {{[-d|--dump]}} {{path/to/file}}
```

- Get a specific attribute of a file:

```
getfattr {{[-n|--name]}} user.{{attribute_name}} {{path/to/file}}
```

getopt

Parse command-line arguments.

More information: https://manned.org/getopt.

- Parse optional `verbose`/`version` flags with shorthands:

```
getopt {{[-o|--options]}} vV {{[-l|--longoptions]}} verbose,version -- --version --verbose
```

- Add a `--file` option with a required argument with shorthand `-f`:

```
getopt {{[-o|--options]}} f: {{[-l|--longoptions]}} file: -- --file=somefile
```

- Add a `--verbose` option with an optional argument with shorthand `-v`, and pass a non-option parameter `arg`:

```
getopt {{[-o|--options]}} v:: {{[-l|--longoptions]}} verbose:: -- --verbose arg
```

- Accept a `-r` and `--verbose` flag, a `--accept` option with an optional argument and add a `--target` with a required argument option with shorthands:

```
getopt {{[-o|--options]}} rv::s::t: {{[-l|--longoptions]}} verbose,source::,target: -- -v --target target
```

getsebool

Get SELinux boolean value.

See also: **`semanage-boolean`**, **`setsebool`**.

More information: https://manned.org/getsebool.

- Show the current setting of a boolean:

```
getsebool {{httpd_can_connect_ftp}}
```

- Show the current setting of [a]ll booleans:

```
getsebool -a
```

- Show the current setting of all booleans with explanations:

```
sudo semanage boolean {{[-l|--list]}}
```

gif2webp

Convert a GIF image to WebP.

More information: https://developers.google.com/speed/webp/docs/gif2webp.

- Convert a GIF image to WebP:

```
gif2webp {{path/to/image.gif}} -o {{path/to/image.webp}}
```

gio trash

Move files to the trash bin.

Used by GNOME to handle trash.

More information: https://manned.org/gio.1.

- Move specific files to the trash bin:

```
gio trash {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}}
```

- List trash bin items:

```
gio trash --list
```

- Restore a specific item from trash using its ID:

```
gio trash trash://{{id}}
```

gio

Handle local and virtual files (GVfs).

Part of GLib used in GNOME-based systems.

More information: https://manned.org/gio.

- Open a file with the default application (e.g. PDF, image):

```
gio open {{path/to/file}}
```

- List files in a directory:

```
gio list {{path/to/directory}}
```

- Show information about a file:

```
gio info {{path/to/file}}
```

- Copy a file:

```
gio copy {{path/to/source}} {{path/to/destination}}
```

- Send a file to the trash (reversible):

```
gio trash {{path/to/file}}
```

gksu

Frontend for **su**.

Allows you to run graphical commands which need root access without having to run an X terminal emulator.

Note: This command is deprecated in favor of commands like **pkexec**, and is no longer maintained.

See also: **gksudo**.

More information: https://manned.org/man/gksudo.

- Run a command as a specific user:

```
gksu {{[-u|--user]}} {{userid}} {{command}}
```

- Run the command while preserving the current environments:

```
gksu {{[-u|--user]}} {{userid}} {{[-k|--preserve-env]}}
{{command}}
```

- Force `gksu` to use `su` to run the command:

```
gksu {{[-u|--user]}} {{userid}} {{[-w|--su-mode]}}
{{command}}
```

- Force `gksu` to use `sudo` to run the command:

```
gksu {{[-u|--user]}} {{userid}} {{[-S|--sudo-mode]}}
{{command}}
```

- Output debug info for the given command:

```
gksu {{[-u|--user]}} {{userid}} {{[-d|--debug]}} {{command}}
```

gksudo

Frontend for **sudo**.

Allows you to run graphical commands which need root access without having to run an X terminal emulator.

Note: This command is deprecated in favor of commands like **pkexec**, and is no longer maintained.

See also: **gksu**.

More information: https://manned.org/man/gksudo.

- Run a command as a specific user:

```
gksudo {{[-u|--user]}} {{userid}} {{command}}
```

- Run the command while preserving the current environments:

```
gksudo {{[-u|--user]}} {{userid}} {{[-k|--preserve-env]}}
{{command}}
```

- Force `gksudo` to use `su` to run the command:

```
gksu {{[-u|--user]}} {{userid}} {{[-w|--su-mode]}}
{{command}}
```

- Force `gksudo` to use `sudo` to run the command:

```
gksudo {{[-u|--user]}} {{userid}} {{[-S|--sudo-mode]}}
{{command}}
```

- Output debug info for the given command:

```
gksudo {{[-u|--user]}} {{userid}} {{[-d|--debug]}}
{{command}}
```

gnome-calculator

The official calculator for the GNOME desktop environment.

More information: https://wiki.gnome.org/Apps/Calculator.

- Launch the GNOME Calculator GUI:

```
gnome-calculator
```

- Solve the specified equation without launching the desktop application:

```
gnome-calculator --solve {{2^5 * 2 + 5}}
```

- Display version:

```
gnome-calculator --version
```

gnome-extensions

Manage gnome extensions from the terminal.

More information: https://wiki.gnome.org/Projects/GnomeShell/Extensions.

- List all the installed extensions:

```
gnome-extensions list
```

- Display information about a specific extension:

```
gnome-extensions info "{{extension_id}}"
```

- Enable a specific extension:

```
gnome-extensions enable "{{extension_id}}"
```

- Disable a specific extension:

```
gnome-extensions disable "{{extension_id}}"
```

- Uninstall a specific extension:

```
gnome-extensions uninstall "{{extension_id}}"
```

- Display help for a specific subcommand (like `list`):

```
gnome-extensions help {{subcommand}}
```

- Display version:

```
gnome-extensions version
```

gnome-screenshot

Capture the screen, a window, or a user-defined area and save the image to a file.

More information: https://manned.org/gnome-screenshot.

- Take a screenshot and save it to the default location, normally `~/Pictures`:

```
gnome-screenshot
```

- Take a screenshot and save it to the named file location:

```
gnome-screenshot {{[-f|--file]}} {{path/to/file}}
```

- Take a screenshot and save it to the clipboard:

```
gnome-screenshot {{[-c|--clipboard]}}
```

- Take a screenshot after the specified number of seconds:

```
gnome-screenshot {{[-d|--delay]}} {{5}}
```

- Launch the GNOME Screenshot GUI:

```
gnome-screenshot {{[-i|--interactive]}}
```

- Take a screenshot of the current window and save it to the specified file location:

```
gnome-screenshot {{[-w|--window]}} {{[-f|--file]}} {{path/to/file}}
```

- Take a screenshot after the specified number of seconds and save it to the clipboard:

```
gnome-screenshot {{[-d|--delay]}} {{10}} {{[-c|--clipboard]}}
```

- Display the version:

```
gnome-screenshot --version
```

gnome-software

Add and remove applications and update your system.

More information: https://manned.org/gnome-software.

- Launch the GNOME Software GUI if it's not already running:

```
gnome-software
```

- Launch the GNOME Software GUI if it's not open, and navigate to the specified page:

```
gnome-software --mode {{updates|updated|installed|overview}}
```

- Launch the GNOME Software GUI if it's not open and view the details of the specified package:

```
gnome-software --details {{package}}
```

- Display the version:

```
gnome-software --version
```

gnome-terminal

The GNOME Terminal emulator.

More information: https://help.gnome.org/users/gnome-terminal/stable/.

- Open a new GNOME terminal window:

```
gnome-terminal
```

- Run a specific command in a new terminal window:

```
gnome-terminal -- {{command}}
```

- Open a new tab in the last opened window instead:

```
gnome-terminal --tab
```

- Set the title of the new tab:

```
gnome-terminal --tab {{[-t|--title]}} "{{title}}"
```

goaccess

An open source real-time web log analyzer.

More information: https://goaccess.io.

- Analyze one or more log files in interactive mode:

```
goaccess {{path/to/logfile1 path/to/file2 ...}}
```

- Use a specific log-format (or pre-defined formats like "combined"):

```
goaccess {{path/to/logfile}} --log-format={{format}}
```

- Analyze a log from `stdin`:

```
tail {{[-f|--follow]}} {{path/to/logfile}} | goaccess -
```

- Analyze a log and write it to an HTML file in real-time:

```
goaccess {{path/to/logfile}} {{[-o|--output]}} {{path/to/
file.html}} --real-time-html
```

goldeneye.py

A HTTP DoS test tool.

More information: https://github.com/jseidl/GoldenEye.

- Test a specific website:

```
./goldeneye.py {{url}}
```

- Test a specific website with 100 user agents and 200 concurrent sockets:

```
./goldeneye.py {{url}} --useragents 100 --sockets 200
```

- Test a specific website without verifying the SSL certificate:

```
./goldeneye.py {{url}} --nosslcheck
```

- Test a specific website in debug mode:

```
./goldeneye.py {{url}} --debug
```

- Display help:

```
./goldeneye.py --help
```

goobook

Access Google contacts from **`mutt`** or the command-line.

More information: https://manned.org/goobook.

- Allow `goobook` to access Google contacts using OAuth2:

```
goobook authenticate
```

- Dump all contacts to XML (`stdout`):

```
goobook dump_contacts
```

google-chrome-stable

This command is an alias of **`chromium`**.

More information: https://chrome.google.com.

- View documentation for the original command:

```
tldr chromium
```

gpasswd

Administer **`/etc/group`** and **`/etc/gshadow`**.

More information: https://manned.org/gpasswd.

- Define group administrators:

```
sudo gpasswd {{[-A|--administrators]}} {{user1,user2}} {{group}}
```

- Set the list of group members:

```
sudo gpasswd {{[-M|--members]}} {{user1,user2}} {{group}}
```

- Create a password for the named group:

```
gpasswd {{group}}
```

- Add a user to the named group:

```
gpasswd {{[-a|--add]}} {{user}} {{group}}
```

- Remove a user from the named group:

```
gpasswd {{[-d|--delete]}} {{user}} {{group}}
```

gpclient

Connect to a GlobalProtect VPN on Linux via OpenConnect.

More information: https://github.com/yuezk/GlobalProtect-openconnect.

- Connect to a GlobalProtect VPN using a portal server:

```
gpclient connect {{vpn_gateway_url}}
```

- Disconnect from the currently connected VPN server:

```
gpclient disconnect
```

- Launch the graphical user interface (GUI) for VPN management:

```
gpclient launch-gui
```

- Use OpenSSL workaround to bypass legacy renegotiation errors:

```
gpclient connect --fix-openssl {{vpn_gateway_url}}
```

- Ignore TLS errors during connection:

```
gpclient connect --ignore-tls-errors {{vpn_gateway_url}}
```

- Display version:

```
gpclient --version
```

- Display help for any command:

```
gpclient help {{command}}
```

gpu-screen-recorder

Record the screen and encode the video with a GPU.

More information: https://git.dec05eba.com/gpu-screen-recorder/about/.

- Select a source using a desktop portal and record it:

```
gpu-screen-recorder -w portal -o {{path/to/video.mp4}}
```

- Specify a specific video source:

```
gpu-screen-recorder -w {{screen|DP-1|HDMI-A1|...}} -o {{path/to/video.mp4}}
```

- List video capture sources:

```
gpu-screen-recorder --list-capture-options
```

- List audio capture sources:

```
gpu-screen-recorder {{--list-audio-devices|--list-application-audio}}
```

- Record using the replay buffer:

```
gpu-screen-recorder -w {{screen}} -r {{30}} -c {{mp4}} -ro {{path/to/directory}} -o {{whatever}}
```

- Capture a video from the replay buffer:

```
pkill -SIGUSR1 -f gpu-screen-recorder
```

- Run `gpu-screen-recorder` in the background:

```
systemctl start --user gpu-screen-recorder
```

grim

Grab images (Screenshots) from a Wayland compositor.

More information: https://sr.ht/~emersion/grim.

- Screenshot all outputs:

```
grim
```

- Screenshot a specific output:

```
grim -o {{path/to/output_file}}
```

- Screenshot a specific region:

```
grim -g "{{x_position}},{{y_position}} {{width}}x{{height}}"
```

- Select a specific region and screenshot it, (using slurp):

```
grim -g "{{$(slurp)}}"
```

- Use a custom filename:

```
grim "{{path/to/file.png}}"
```

- Screenshot and copy to clipboard:

```
grim - | {{clipboard_manager}}
```

groupadd

Add user groups to the system.

See also: **`groups`**, **`groupdel`**, **`groupmod`**.

More information: https://manned.org/groupadd.

- Create a new group:

```
sudo groupadd {{group_name}}
```

- Create a new system group:

```
sudo groupadd {{[-r|--system]}} {{group_name}}
```

- Create a new group with the specific groupid:

```
sudo groupadd {{[-g|--gid]}} {{id}} {{group_name}}
```

groupdel

Delete existing user groups from the system.

See also: **`groups`**, **`groupadd`**, **`groupmod`**.

More information: https://manned.org/groupdel.

- Delete an existing group:

```
sudo groupdel {{group_name}}
```

groupmod

Modify existing user groups in the system.

See also: **`groups`**, **`groupadd`**, **`groupdel`**.

More information: https://manned.org/groupmod.

- Change the group name:

```
sudo groupmod {{[-n|--new-name]}} {{new_group}} {{group_name}}
```

- Change the group ID:

```
sudo groupmod {{[-g|--gid]}} {{new_id}} {{group_name}}
```

growpart

Extend a partition in a disk or disk image to fill available space.

More information: https://github.com/canonical/cloud-utils.

- Extend partition n from sdX to fill empty space until end of disk or beginning of next partition:

```
growpart {{/dev/sdX}} {{n}}
```

- Show what modifications would be made when growing partition n in a disk image:

```
growpart {{[-N|--dry-run]}} /{{path/to/disk.img}} {{n}}
```

grub-bios-setup

Set up a device to use GRUB with a BIOS configuration.

You should use **`grub-install`** instead of **`grub-bios-setup`** in most cases.

More information: https://manned.org/grub-bios-setup.8.

- Set up a device to boot with GRUB:

```
grub-bios-setup {{/dev/sdX}}
```

- Install even if problems are detected:

```
grub-bios-setup {{[-f|--force]}} {{/dev/sdX}}
```

- Install GRUB in a specific directory:

```
grub-bios-setup {{[-d|--directory]}} {{/boot/grub}} {{/dev/sdX}}
```

grub-editenv

Edit GRUB environment variables.

More information: https://www.gnu.org/software/grub/manual/grub/grub.html.

- Set a default boot entry (Assuming the boot entry already exists):

```
grub-editenv /boot/grub/grubenv set default={{Ubuntu}}
```

- Display all GRUB environment variables:

```
grub-editenv /boot/grub/grubenv list
```

- Reset the `saved_entry` variable to the default:

```
grub-editenv /boot/grub/grubenv unset saved_entry
```

grub-file

Check if a file is of a bootable image type.

More information: https://manned.org/grub-file.

- Check if a file is an ARM EFI image:

```
grub-file --is-arm-efi {{path/to/file}}
```

- Check if a file is an i386 EFI image:

```
grub-file --is-i386-efi {{path/to/file}}
```

- Check if a file is an x86_64 EFI image:

```
grub-file --is-x86_64-efi {{path/to/file}}
```

- Check if a file is an ARM image (Linux kernel):

```
grub-file --is-arm-linux {{path/to/file}}
```

- Check if a file is an x86 image (Linux kernel):

```
grub-file --is-x86-linux {{path/to/file}}
```

- Check if a file is an x86_64 XNU image (macOS kernel):

```
grub-file --is-x86_64-xnu {{path/to/file}}
```

grub-install

Install GRUB to a device.

More information: https://www.gnu.org/software/grub/manual/grub/grub.html#Installing-GRUB-using-grub_002dinstall.

- Install GRUB on a BIOS system:

```
sudo grub-install {{path/to/device}}
```

- Install GRUB on a BIOS system while specifying architecture:

```
sudo grub-install --target {{i386-pc}} {{path/to/device}}
```

- Install GRUB on an UEFI system:

```
sudo grub-install --efi-directory {{path/to/efi_directory}}
```

- Install GRUB on an UEFI system while specifying architecture and boot menu text:

```
sudo grub-install --target {{x86_64-efi}} --efi-directory {{path/to/efi_directory}} --bootloader-id {{GRUB}}
```

- Install GRUB pre-loading specific modules:

```
sudo grub-install --target {{x86_64-efi}} --efi-directory {{path/to/efi_directory}} --modules "{{part_gpt part_msdos}}"
```

- Display help:

```
grub-install {{[-?|--help]}}
```

grub-mkconfig

Generate a GRUB configuration file.

More information: https://www.gnu.org/software/grub/manual/grub/grub.html#Invoking-grub_002dmkconfig.

- Do a dry run and print the configuration to `stdout`:

```
sudo grub-mkconfig
```

- Generate the configuration file:

```
sudo grub-mkconfig {{[-o|--output]}} {{/boot/grub/grub.cfg}}
```

- Display help:

```
grub-mkconfig {{[-h|--help]}}
```

grub-mkrescue

Make a GRUB CD/USB/floppy bootable image.

More information: https://www.gnu.org/software/grub/manual/grub/grub.html#Invoking-grub_002dmkrescue.

- Create a bootable ISO from the current directory and save it as `grub.iso`:

```
grub-mkrescue --output {{grub.iso}} .
```

- Create an ISO using GRUB files from a custom directory:

```
grub-mkrescue --directory {{/usr/lib/grub/i386-pc}} --output {{grub.iso}} {{path/to/source}}
```

- Use compression for GRUB files when building the image, setting `no` disables compression:

```
grub-mkrescue --compress {{no|xz|gz|lzo}} --output {{grub.iso}} {{path/to/source}}
```

- Disable the GRUB command-line interface in the generated image:

```
grub-mkrescue --disable-cli --output {{grub.iso}} {{path/to/source}}
```

- Preload specific GRUB modules into the image:

```
grub-mkrescue --modules "{{part_gpt iso9660}}" --output {{grub.iso}} {{path/to/source}}
```

- Pass additional options directly to `xorriso`:

```
grub-mkrescue --output {{grub.iso}} -- {{-volid}} {{volume_name}} {{path/to/source}}
```

- Display help:

```
grub-mkrescue {{[-?|--help]}}
```

- Display version:

```
grub-mkrescue --version
```

grub-mount

Mount a file system or file system image read-only using GRUB's file system drivers.

More information: https://www.gnu.org/software/grub/manual/grub/grub.html#Invoking-grub_002dmount.

- Mount a block device or file system image to a mount point:

```
grub-mount {{/dev/sdXY}} {{/mnt}}
```

- Mount an entire disk image's second partition, `-r` specifies the partition number in the image:

```
grub-mount {{[-r|--root]}} {{2}} {{disk.img}} {{/mnt}}
```

- Mount an encrypted device and prompt for a passphrase:

```
grub-mount {{[-C|--crypto]}} {{/dev/sdXY}} {{/mnt}}
```

- Load a ZFS encryption key from a file:

```
grub-mount {{[-K|--zfs-key]}} /{{path/to/zfs.key}} {{/dev/sdX}} {{/mnt}}
```

- Show debugging output for a matching category:

```
grub-mount {{[-d|--debug]}} {{string}} {{image}} {{/mnt}}
```

- Enable verbose output:

```
grub-mount {{[-v|--verbose]}} {{image}} {{/mnt}}
```

- Display help:

```
grub-mount {{[-?|--help]}}
```

- Display version:

```
grub-mount --version
```

grub-probe

Probe device information for a particular path or device.

More information: https://www.gnu.org/software/grub/manual/grub/html_node/Invoking-grub_002dprobe.html.

- Get GRUB filesystem module for a path:

```
sudo grub-probe {{[-t|--target]}} fs {{/boot/grub}}
```

- Get the system device containing a path:

```
sudo grub-probe {{[-t|--target]}} device {{/boot/grub}}
```

- Get GRUB disk name for a system device:

```
sudo grub-probe {{[-t|--target]}} drive {{/dev/sdX}} {{[-d|--device]}}
```

- Get filesystem UUID:

```
sudo grub-probe {{[-t|--target]}} fs_uuid {{/boot/grub}}
```

- Get filesystem label:

```
sudo grub-probe {{[-t|--target]}} fs_label {{/boot/grub}}
```

- Get MBR partition type code(two hexadecimal digits):

```
sudo grub-probe {{[-t|--target]}} msdos_parttype {{/dev/sdX}}
```

- Probe using a custom device map:

```
sudo grub-probe {{[-t|--target]}} drive {{/boot/grub}} {{[-m|--device-map]}} {{path/to/custom_device.map}}
```

grub-reboot

Set the default boot entry for GRUB, for the next boot only.

More information: https://manned.org/grub-reboot.

- Set the default boot entry to an entry number, name or identifier for the next boot:

```
sudo grub-reboot {{entry_number}}
```

- Set the default boot entry to an entry number, name or identifier for an alternative boot directory for the next boot:

```
sudo grub-reboot --boot-directory /{{path/to/boot_directory}} {{entry_number}}
```

grub-script-check

The program **grub-script-check** takes a GRUB script file and checks it for syntax errors.

It may take a path as a non-option argument. If none is supplied, it will read from **stdin**.

More information: https://www.gnu.org/software/grub/manual/grub/grub.html#Invoking-grub_002dscript_002dcheck.

- Check a specific script file for syntax errors:

```
grub-script-check {{path/to/grub_config_file}}
```

- Display each line of input after reading it:

```
grub-script-check {{[-v|--verbose]}}
```

- Display help:

```
grub-script-check --help
```

- Display version:

```
grub-script-check --version
```

grub-set-default

Set the default boot entry for GRUB.

More information: https://manned.org/grub-set-default.

- Set the default boot entry to an entry number, name or identifier:

```
sudo grub-set-default {{entry_number}}
```

- Set the default boot entry to an entry number, name or identifier for an alternative boot directory:

```
sudo grub-set-default --boot-directory /{{path/to/boot_directory}} {{entry_number}}
```

grub2-mkpasswd-pbkdf2

Generate a hashed password for GRUB.

More information: https://manned.org/grub2-mkpasswd-pbkdf2.

- Create a password hash for GRUB 2 using PBKDF2 and print it to `stdout`:

```
sudo grub2-mkpasswd-pbkdf2 {{[-c|--iteration-count]}}
{{number_of_pbkdf2_iterations}} {{[-s|--salt]}}
{{salt_length}}
```

grubby

Tool for configuring **`grub`** and **`zipl`** bootloaders.

More information: https://manned.org/grubby.8.

- Add kernel boot arguments to all kernel menu entries:

```
sudo grubby --update-kernel=ALL --args '{{quiet console=ttyS0}}'
```

- Remove existing arguments from the entry for the default kernel:

```
sudo grubby --update-kernel=DEFAULT --remove-args {{quiet}}
```

- List all kernel menu entries:

```
sudo grubby --info=ALL
```

gs

GhostScript, a PDF and PostScript interpreter.

More information: https://manned.org/gs.

- To view a file:

```
gs -dQUIET -dBATCH {{file.pdf}}
```

- Reduce PDF file size to 150 dpi images for reading on a e-book device:

```
gs -dNOPAUSE -dQUIET -dBATCH -sDEVICE=pdfwrite -
dPDFSETTINGS=/ebook -sOutputFile={{output.pdf}} {{input.pdf}}
```

- Convert PDF file (pages 1 through 3) to an image with 150 dpi resolution:

```
gs -dQUIET -dBATCH -dNOPAUSE -sDEVICE=jpeg -r150 -
dFirstPage={{1}} -dLastPage={{3}} -
sOutputFile={{output_%d.jpg}} {{input.pdf}}
```

- Extract pages from a PDF file:

```
gs -dQUIET -dBATCH -dNOPAUSE -sDEVICE=pdfwrite -
sOutputFile={{output.pdf}} {{input.pdf}}
```

- Merge PDF files:

```
gs -dQUIET -dBATCH -dNOPAUSE -sDEVICE=pdfwrite -
sOutputFile={{output.pdf}} {{input1.pdf}} {{input2.pdf}}
```

- Convert from PostScript file to PDF file:

```
gs -dQUIET -dBATCH -dNOPAUSE -sDEVICE=pdfwrite -
sOutputFile={{output.pdf}} {{input.ps}}
```

gsettings

Query and modify dconf settings with schema validation.

More information: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/using_the_desktop_environment_in_rhel_8/configuring-gnome-at-low-level_using-the-desktop-environment-in-rhel-8#using-gsettings-command_configuring-gnome-at-low-level.

- Set the value of a key. Fails if the key doesn't exist or the value is out of range:

```
gsettings set {{org.example.schema}} {{example-key}} {{value}}
```

- Print the value of a key or the schema-provided default if the key has not been set in `dconf`:

```
gsettings get {{org.example.schema}} {{example-key}}
```

- Unset a key, so that its schema default value will be used:

```
gsettings reset {{org.example.schema}} {{example-key}}
```

- Display all (non-relocatable) schemas, keys, and values:

```
gsettings list-recursively
```

- Display all keys and values (default if not set) from one schema:

```
gsettings list-recursively {{org.example.schema}}
```

- Display schema-allowed values for a key (helpful with enum keys):

```
gsettings range {{org.example.schema}} {{example-key}}
```

- Display the human-readable description of a key:

```
gsettings describe {{org.example.schema}} {{example-key}}
```

gtk-launch

Launch applications from **`.desktop`** files that reside in standard locations.

More information: https://manned.org/gtk-launch.

- Launch an application:

```
gtk-launch {{Application_name}}
```

- Display help:

```
gtk-launch {{[-h|--help]}}
```

guake

A drop-down terminal for GNOME.

See also: **yakuake**.

More information: https://github.com/Guake/guake.

- Toggle Guake visibility:

  `<F12>`

- Toggle fullscreen mode:

  `<F11>`

- Open a new tab:

  `<Ctrl Shift t>`

- Close the terminal:

  `<Super x>`

- Go to the previous tab:

  `<Ctrl PageUp>`

- Search the selected text in the browser:

  `<Ctrl Shift l>`

guix package

Install, upgrade and remove Guix packages, or rollback to previous configurations.

More information: https://guix.gnu.org/manual/en/guix.html#Invoking-guix-package.

- Install a new package:

```
guix package {{[-i|--install]}} {{package}}
```

- Remove a package:

```
guix package {{[-r|--remove]}} {{package}}
```

- Search the package database for a `regex`:

```
guix package {{[-s|--search]}} "{{search_pattern}}"
```

- List installed packages:

```
guix package {{[-I|--list-installed]}}
```

- List generations:

```
guix package {{[-l|--list-generations]}}
```

- Roll back to the previous generation:

```
guix package --roll-back
```

gummy

Screen brightness/temperature manager for Linux/X11.

More information: https://github.com/Gitoffthelawn/gummy.

- Set the screen temperature to 3000K:

```
gummy {{[-t|--temperature]}} {{3000}}
```

- Set the screen backlight to 50%:

```
gummy --backlight {{50}}
```

- Set the screen pixel brightness to 45%:

```
gummy {{[-b|--brightness]}} {{45}}
```

- Increase current screen pixel brightness by 10%:

```
gummy {{[-b|--brightness]}} {{+10}}
```

- Decrease current screen pixel brightness by 10%:

```
gummy {{[-b|--brightness]}} {{-10}}
```

- Set the temperature and pixel brightness for the second screen:

```
gummy {{[-s|--screen]}} {{1}} {{[-t|--temperature]}} {{3800}} {{[-b|--brightness]}} {{65}}
```

gzexe

Compress executable files while keeping them executable.

Back up the original file, appending **~** to its name and create a shell script that uncompresses and executes the binary inside it.

More information: https://manned.org/gzexe.1.

- Compress an executable file in-place:

```
gzexe {{path/to/executable}}
```

- [d]ecompress a compressed executable in-place (i.e. convert the shell script back to an uncompressed binary):

```
gzexe -d {{path/to/compressed_executable}}
```

halt

Halt the system.

More information: https://manned.org/halt.8.

- Halt the system:

  `halt`

- Power off the system (same as `poweroff`):

  `halt {{[-p|--poweroff]}}`

- Reboot the system (same as `reboot`):

  `halt --reboot`

- Halt immediately without contacting the system manager:

  `halt {{[-f|--force]}}`

- Write the wtmp shutdown entry without halting the system:

  `halt {{[-w|--wtmp-only]}}`

handlr

Manage your default applications.

More information: https://github.com/chmln/handlr.

- Open a URL in the default application:

```
handlr open {{https://example.com}}
```

- Open a PDF in the default PDF viewer:

```
handlr open {{path/to/file.pdf}}
```

- Set `imv` as the default application for PNG files:

```
handlr set {{.png}} {{imv.desktop}}
```

- Set MPV as the default application for all audio files:

```
handlr set '{{audio/*}}' {{mpv.desktop}}
```

- List all default apps:

```
handlr list
```

- Print the default application for PNG files:

```
handlr get {{.png}}
```

hardinfo

Show hardware information in GUI window.

More information: https://github.com/lpereira/hardinfo.

- Start `hardinfo`:

```
hardinfo
```

- Print report to `stdout`:

```
hardinfo {{[-r|--generate-report]}}
```

- Save report to HTML file:

```
hardinfo {{[-r|--generate-report]}} {{[-f|--report-format]}}
html > hardinfo.html
```

haveged

Hardware-based random number generator.

More information: https://manned.org/haveged.

- Generate a random number:

```
sudo haveged
```

- Run `haveged` in foreground:

```
sudo haveged {{[-F|--Foreground]}}
```

- Set file path for output of `haveged`:

```
sudo haveged {{[-f|--file]}} {{path/to/file}}
```

- Set run level for daemon:

```
sudo haveged {{[-r|--run]}} {{runlevel}}
```

- Set collection buffer size in kibibyte words:

```
sudo haveged {{[-b|--buffer]}} {{buffersizeinKW}}
```

- Insert a command to an already running `haveged` process or daemon:

```
sudo haveged {{[-c|--command]}} {{command}}
```

- Set cache size in kibibyte words:

```
sudo haveged {{[-d|--data]}} {{cachesizeinKW}}
```

- Set number of bytes to write to the output file:

```
sudo haveged {{[-n|--number]}} {{byteamount}}
```

hcitool

Monitor, configure connections, and send special commands to Bluetooth devices.

More information: https://manned.org/hcitool.

- Scan for Bluetooth devices:

```
hcitool scan
```

- Output the name of a device, returning its MAC address:

```
hcitool name {{bdaddr}}
```

- Fetch information about a remote Bluetooth device:

```
hcitool info {{bdaddr}}
```

- Check the link quality to a Bluetooth device:

```
hcitool lq {{bdaddr}}
```

- Modify the transmit power level:

```
hcitool tpl {{bdaddr}} {{0|1}}
```

- Display the link policy:

```
hcitool lp
```

- Request authentication with a specific device:

```
hcitool auth {{bdaddr}}
```

- Display local devices:

```
hcitool dev
```

hd-idle

Spin down external disks after a period of idle time.

More information: https://manned.org/hd-idle.

- Start a service to manage hard drive spin down. By default hard disks will spin down after 10 minutes of inactivity:

```
systemctl start hd-idle
```

- Spin down a disk immediately:

```
hd-idle -t {{/dev/sdX}}
```

- Set disks to never spin down, then set explicit idle times (in seconds) for disks which have "sda" or "sdb" in their device name:

```
hd-idle -i 0 -a /dev/sda -i {{300}} -a /dev/sdb -i {{1200}}
```

hddtemp

Display temperature of HDD via S.M.A.R.T.

More information: https://manned.org/hddtemp.

- Display temperature of a specific drive:

```
hddtemp {{type}}:{{/dev/sdX}}
```

- Display temperature of a SATA drive assigned to `sda`:

```
hddtemp SATA:/dev/sda
```

- Log temperatures to syslog every `n` seconds:

```
hddtemp {{[-S|--syslog]}} {{nseconds}} {{type}}:{{/dev/sdX}}
```

- Print only numeric value of temperature without unit:

```
hddtemp {{[-n|--numeric]}} {{type}}:{{/dev/sdX}}
```

- Define the unit used to denote temperature:

```
hddtemp {{[-u|--unit]}} {{C|F}} {{type}}:{{/dev/sdX}}
```

- Wake ATA drive before attempting to read temperature:

```
hddtemp {{[-w|--wake-up]}} {{type}}:{{/dev/sdX}}
```

- Enter debug mode to show S.M.A.R.T. fields and their values:

```
hddtemp {{[-D|--debug]}} {{type}}:{{/dev/sdX}}
```

- Suppress compatibility check for drive types:

```
hddtemp {{[-q|--quiet]}} {{type}}:{{/dev/sdX}}
```

hdparm

Get and set SATA and IDE hard drive parameters.

More information: https://manned.org/hdparm.

- Request the identification info of a given device:

```
sudo hdparm -I {{/dev/device}}
```

- Get the Advanced Power Management level:

```
sudo hdparm -B {{/dev/device}}
```

- Set the Advanced Power Management value (values 1-127 permit spin-down, and values 128-254 do not):

```
sudo hdparm -B {{1}} {{/dev/device}}
```

- Display the device's current power mode status:

```
sudo hdparm -C {{/dev/device}}
```

- Force a drive to immediately enter standby mode (usually causes a drive to spin down):

```
sudo hdparm -y {{/dev/device}}
```

- Put the drive into idle (low-power) mode, also setting its standby timeout:

```
sudo hdparm -S {{standby_timeout}} {{device}}
```

- Test the read speed of a specific device:

```
sudo hdparm -tT {{device}}
```

head

Output the first part of files.

More information: https://www.gnu.org/software/coreutils/manual/html_node/head-invocation.html.

- Show first 10 lines in a file:

```
head {{path/to/file}}
```

- Show first 10 lines of multiple files:

```
head {{path/to/file1 path/to/file2 ...}}
```

- Output the first 5 lines of a file:

```
head {{[-5|--lines 5]}} {{path/to/file}}
```

- Output the first few lines of a file:

```
head {{[-n|--lines]}} {{count}} {{path/to/file}}
```

- Output the first few bytes of a file:

```
head {{[-c|--bytes]}} {{count}} {{path/to/file}}
```

- Output everything but the last few lines of a file:

```
head {{[-n|--lines]}} -{{count}} {{path/to/file}}
```

- Output everything but the last few bytes of a file:

```
head {{[-c|--bytes]}} -{{count}} {{path/to/file}}
```

hexdump

An ASCII, decimal, hexadecimal, octal dump.

See also: **`hexyl`**, **`od`**, **`xxd`**.

More information: https://manned.org/hexdump.

- Print the hexadecimal representation of a file, replacing duplicate lines by '*':

```
hexdump {{path/to/file}}
```

- Display the input offset in hexadecimal and its ASCII representation in two columns:

```
hexdump {{[-C|--canonical]}} {{path/to/file}}
```

- Display the hexadecimal representation of a file, but interpret only n bytes of the input:

```
hexdump {{[-C|--canonical]}} {{[-n|--length]}}
{{number_of_bytes}} {{path/to/file}}
```

- Don't replace duplicate lines with '*':

```
hexdump {{[-v|--no-squeezing]}} {{path/to/file}}
```

hlint

Suggest improvements to Haskell code.

More information: https://hackage.haskell.org/package/hlint.

- Display suggestions for a given file:

```
hlint {{path/to/file}} options
```

- Check all Haskell files and generate a report:

```
hlint {{path/to/directory}} {{[-r|--report]}}
```

- Automatically apply most suggestions:

```
hlint {{path/to/file}} --refactor
```

- Display additional options:

```
hlint {{path/to/file}} --refactor-options
```

- Generate a settings file ignoring all outstanding hints:

```
hlint {{path/to/file}} --default > {{.hlint.yaml}}
```

hollywood

Fill your console with Hollywood melodrama technobabble.

More information: https://manned.org/hollywood.

- Fill the console:

  ```
  hollywood
  ```

- Exit `hollywood`:

  ```
  <Ctrl c><Ctrl c>
  ```

- Display help:

  ```
  hollywood {{[-h|--help]}}
  ```

homectl

Create, remove, change or inspect home directories using the systemd-homed service.

More information: https://www.freedesktop.org/software/systemd/man/homectl.html.

- List user accounts and their associated home directories:

```
homectl list
```

- Create a user account and their associated home directory:

```
sudo homectl create {{username}}
```

- Remove a specific user and the associated home directory:

```
sudo homectl remove {{username}}
```

- Change the password for a specific user:

```
sudo homectl passwd {{username}}
```

- Run a shell or a command with access to a specific home directory:

```
sudo homectl with {{username}} -- {{command}} {{command_arguments}}
```

- Lock or unlock a specific home directory:

```
sudo homectl {{lock|unlock}} {{username}}
```

- Change the disk space assigned to a specific home directory to 100 GiB:

```
sudo homectl resize {{username}} {{100G}}
```

- Display help:

```
homectl {{[-h|--help]}}
```

homeshick

Synchronize Git dotfiles.

See also: **`chezmoi`**, **`stow`**, **`tuckr`**, **`vcsh`**.

More information: https://github.com/andsens/homeshick/wiki.

- Create a new castle:

```
homeshick generate {{castle_name}}
```

- Add a file to your castle:

```
homeshick track {{castle_name}} {{path/to/file}}
```

- Go to a castle:

```
homeshick cd {{castle_name}}
```

- Clone a castle:

```
homeshick clone {{github_username}}/{{repository_name}}
```

- Symlink all files from a castle:

```
homeshick link {{castle_name}}
```

hostnamectl

Get or set the hostname of the computer.

More information: https://www.freedesktop.org/software/systemd/man/hostnamectl.html.

- Get the hostname of the computer:

```
hostnamectl
```

- Set the hostname of the computer:

```
sudo hostnamectl set-hostname "{{hostname}}"
```

- Set a pretty hostname for the computer:

```
sudo hostnamectl set-hostname --static
"{{hostname.example.com}}" && sudo hostnamectl set-hostname
--pretty "{{hostname}}"
```

- Reset hostname to its default value:

```
sudo hostnamectl set-hostname --pretty ""
```

htpdate

Synchronize local date and time via HTTP headers from web servers.

More information: https://www.vervest.org/htp/.

- Synchronize date and time:

```
sudo htpdate {{host}}
```

- Perform simulation of synchronization, without any action:

```
htpdate -q {{host}}
```

- Compensate the systematic clock drift:

```
sudo htpdate -x {{host}}
```

- Set time immediate after the synchronization:

```
sudo htpdate -s {{host}}
```

http-prompt

An interactive HTTP client featuring autocomplete and syntax highlighting.

More information: https://github.com/httpie/http-prompt.

- Launch a session targeting the default URL of http://localhost:8000 or the previous session:

```
http-prompt
```

- Launch a session with a given URL:

```
http-prompt {{http://example.com}}
```

- Launch a session with some initial options:

```
http-prompt {{localhost:8000/api}} --auth
{{username:password}}
```

http_load

An HTTP benchmarking tool.

Runs multiple HTTP fetches in parallel to test the throughput of a web server.

More information: https://www.acme.com/software/http_load/.

- Emulate 20 requests based on a given URL list file per second for 60 seconds:

```
http_load -rate {{20}} -seconds {{60}} {{path/to/urls.txt}}
```

- Emulate 5 concurrent requests based on a given URL list file for 60 seconds:

```
http_load -parallel {{5}} -seconds {{60}} {{path/to/
urls.txt}}
```

- Emulate 1000 requests at 20 requests per second, based on a given URL list file:

```
http_load -rate {{20}} -fetches {{1000}} {{path/to/urls.txt}}
```

- Emulate 1000 requests at 5 concurrent requests at a time, based on a given URL list file:

```
http_load -parallel {{5}} -fetches {{1000}} {{path/to/
urls.txt}}
```

hwclock

Read or change the hardware clock. Usually requires root.

More information: https://manned.org/hwclock.

- Display the current time as reported by the hardware clock:

```
sudo hwclock
```

- Write the current software clock time to the hardware clock (sometimes used during system setup):

```
sudo hwclock {{[-w|--systohc]}}
```

- Write the current hardware clock time to the software clock:

```
sudo hwclock {{[-s|--hctosys]}}
```

hwinfo

Probe for the hardware present in the system.

See also: **`inxi`**, **`lshw`**.

More information: https://manpages.opensuse.org/hwinfo/hwinfo.8.en.html.

- Display all available hardware information:

```
hwinfo
```

- Display information about a specific hardware component:

```
hwinfo --{{cpu|memory|disk|gfxcard|network|usb|pci|keyboard|
mouse|monitor|sound|fingerprint|...}}
```

- Display information about a specific hardware component succinctly:

```
hwinfo {{--component}} --short
```

- Write all hardware information to a file:

```
hwinfo --all --log {{path/to/file}}
```

- Display help:

```
hwinfo --help
```

hwloc-ls

This command is an alias of **`lstopo-no-graphics`**.

- View documentation for the original command:

  ```
  tldr lstopo-no-graphics
  ```

hyprctl

Control parts of the Hyprland Wayland compositor.

More information: https://wiki.hypr.land/Configuring/Using-hyprctl/.

- Reload Hyprland configuration:

```
hyprctl reload
```

- Return the active window name:

```
hyprctl activewindow
```

- List all connected input devices:

```
hyprctl devices
```

- List all outputs with respective properties:

```
hyprctl workspaces
```

- Call a dispatcher:

```
hyprctl dispatch {{dispatcher}}
```

- Set a configuration keyword dynamically:

```
hyprctl keyword {{keyword}} {{value}}
```

- Display version:

```
hyprctl version
```

hyprpicker

Minimal Wayland color picker for wlroots compositors (e.g., Hyprland).

Requires a Wayland session. For clipboard autocopy, **`wl-copy`** must be installed.

More information: https://wiki.hypr.land/Hypr-Ecosystem/hyprpicker/.

- Pick a color in the default (hex) format:

  ```
  hyprpicker
  ```

- Pick a color and output in a specific format:

  ```
  hyprpicker {{[-f|--format]}} {{hex|rgb|hsl|hsv|cmyk}}
  ```

- Copy the picked color to the clipboard:

  ```
  hyprpicker {{[-a|--autocopy]}}
  ```

- Disable colored output (print plain text only):

  ```
  hyprpicker {{[-n|--no-fancy]}}
  ```

- Store the picked color in a shell variable:

  ```
  {{color}}=$(hyprpicker {{[-f|--format]}} {{hex}})
  ```

- Display help:

  ```
  hyprpicker {{[-h|--help]}}
  ```

hyprpm

Control plugins for the Hyprland Wayland compositor.

More information: https://wiki.hypr.land/Plugins/Using-Plugins/#hyprpm.

- Add a plugin:

```
hyprpm add {{git_url}}
```

- Remove a plugin:

```
hyprpm remove {{git_url|plugin_name}}
```

- Enable a plugin:

```
hyprpm enable {{plugin_name}}
```

- Disable a plugin:

```
hyprpm disable {{plugin_name}}
```

- Update and check all plugins:

```
hyprpm update
```

- Force an operation:

```
hyprpm {{[-f|--force]}} {{operation}}
```

- List all installed plugins:

```
hyprpm list
```

hyprshot

Screenshot utility for the Hyprland Wayland compositor.

More information: https://github.com/Gustash/Hyprshot.

- Select and take a screenshot of a region:

```
hyprshot {{[-m|--mode]}} region
```

- Select and take a screenshot of a specific window:

```
hyprshot {{[-m|--mode]}} window
```

- Select and take a screenshot of a specific output:

```
hyprshot {{[-m|--mode]}} output
```

- Take a screenshot of the currently active window:

```
hyprshot {{[-m|--mode]}} active {{[-m|--mode]}} window
```

- Freeze the screen and take a screenshot of the selected region:

```
hyprshot {{[-z|--freeze]}} {{[-m|--mode]}} region
```

- Select and take a screenshot of a specific window, saving to the given output directory:

```
hyprshot {{[-o|--output-folder]}} {{path/to/directory}} {{[-m|--mode]}} window
```

- Select and take a screenshot of a specific output, saving the screenshot to the clipboard only:

```
hyprshot --clipboard {{[-m|--mode]}} output
```

i2cdetect

Scan I2C buses.

See also: **`i2cdump`**, **`i2cget`**, **`i2cset`**.

More information: https://manned.org/i2cdetect.

- List active I2C buses:

```
i2cdetect -l
```

- Scan devices on an I2C bus:

```
i2cdetect {{i2c_bus}}
```

- Scan devices on an I2C bus without asking for confirmation:

```
i2cdetect -y {{i2c_bus}}
```

i2cdump

Dump I2C device registers.

See also: **`i2cdetect`**, **`i2cget`**, **`i2cset`**.

Note: All addresses should be specified in hexadecimal.

More information: https://manned.org/i2cdump.

- Dump all registers of an I2C device:

```
i2cdump {{i2cbus}} {{device_address}}
```

- Dump all registers of an I2C device without asking for confirmation:

```
i2cdump -y {{i2cbus}} {{device_address}}
```

- Dump all registers of an I2C device using a specific mode:

```
i2cdump {{i2cbus}} {{device_address}} {{b|w|c|s|i}}
```

- Dump registers from `start` to `end` of an I2C device:

```
i2cdump -r {{start}}-{{end}} {{i2cbus}} {{device_address}}
```

i2cget

Read from an register of an I2C device.

See also: **`i2cdetect`**, **`i2cdump`**, **`i2cset`**.

Note: All addresses should be specified in hexadecimal.

More information: https://manned.org/i2cget.

- Read from a register of an I2C device:

```
i2cget {{i2cbus}} {{device_address}} {{register_address}}
```

- Read from a register of an I2C device without asking for confirmation:

```
i2cget -y {{i2cbus}} {{device_address}} {{register_address}}
```

- Read from a register of an I2C device using a specific mode:

```
i2cget {{i2cbus}} {{device_address}} {{register_address}}
{{b|w|c|s|i}}
```

i2cset

Set the value of a register of an I2C device.

See also: **`i2cdetect`**, **`i2cdump`**, **`i2cget`**.

Note: All addresses should be specified in hexadecimal.

More information: https://manned.org/i2cset.

- Write to a register of an I2C device:

```
i2cset {{i2cbus}} {{device_address}} {{register_address}} {{value}}
```

- Write to a register of an I2C device without asking for confirmation:

```
i2cset -y {{i2cbus}} {{device_address}} {{register_address}} {{value}}
```

- Write to a register of an I2C device using a specific mode:

```
i2cset {{i2cbus}} {{device_address}} {{register_address}} {{value}} {{b|w|c|s|i}}
```

i3-scrot

Wrapper script around the screenshot utility **`scrot`** for the i3 window manager.

The default save location is **`~/Pictures`** and can be changed in **`~/.config/i3-scrot.conf`**.

More information: https://gitlab.manjaro.org/packages/community/i3/i3-scrot.

- Capture a screenshot of the whole screen and save it to the default directory:

```
i3-scrot
```

- Capture a screenshot of the active window:

```
i3-scrot --window
```

- Capture a screenshot of a specific rectangular selection:

```
i3-scrot --select
```

- Capture a screenshot of the whole screen and copy it to the clipboard:

```
i3-scrot --desk-to-clipboard
```

- Capture a screenshot of the active window and copy it to the clipboard:

```
i3-scrot --window-to-clipboard
```

- Capture a screenshot of a specific selection and copy it to the clipboard:

```
i3-scrot --select-to-clipboard
```

- Capture a screenshot of the active window after a delay of 5 seconds:

```
i3-scrot --window {{5}}
```

i386

This command is an alias of **`setarch i386`**.

- View documentation for the original command:

```
tldr setarch
```

i3exit

Exit the i3 window manager.

More information: https://gitlab.manjaro.org/packages/extra/i3exit/-/blob/master/i3exit.

- Log out of i3:

```
i3exit logout
```

- Lock i3:

```
i3exit lock
```

- Shut down the system:

```
i3exit shutdown
```

- Suspend the system:

```
i3exit suspend
```

- Switch to the login screen to log in as a different user:

```
i3exit switch_user
```

- Hibernate the system:

```
i3exit hibernate
```

- Reboot the system:

```
i3exit reboot
```

i3status

Status line for the i3 window manager.

This command is usually called from the i3 configuration file.

More information: https://i3wm.org/i3status/manpage.html.

- Print the status line to `stdout` periodically, using the default configuration:

```
i3status
```

- Print the status line to `stdout` periodically, using a specific configuration:

```
i3status -c {{path/to/i3status.conf}}
```

- Display help and version:

```
i3status -h
```

i7z

An Intel CPU (only i3, i5 and i7) realtime reporting tool.

More information: https://manned.org/i7z.

- Start i7z (needs to be run in superuser mode):

```
sudo i7z
```

ico

Display an animation of a polyhedron.

More information: https://manned.org/ico.1.

- Display the wireframe of an icosahedron that changes its position every 0.1 seconds:

```
ico -sleep {{0.1}}
```

- Display a solid icosahedron with red faces on a blue background:

```
ico -faces -noedges -colors {{red}} -bg {{blue}}
```

- Display the wireframe of a cube with size 100x100 that moves by +1+2 per frame:

```
ico -obj {{cube}} -size {{100x100}} -delta {{+1+2}}
```

- Display the inverted wireframe of an icosahedron with line width 10 using 5 threads:

```
ico -i -lw {{10}} -threads {{5}}
```

id3v2

Manage id3v2 tags, converts and lists id3v1.

More information: https://manned.org/id3v2.1.

- List all genres:

```
id3v2 {{[-L|--list-genres]}}
```

- List all tags of specific files:

```
id3v2 {{[-l|--list]}} {{path/to/file1 path/to/file2 ...}}
```

- Delete all `id3v2` or `id3v1` tags of specific files:

```
id3v2 {{--delete-v2|--delete-v1}} {{path/to/file1 path/to/
file2 ...}}
```

- Display help:

```
id3v2 {{[-h|--help]}}
```

- Display version:

```
id3v2 {{[-v|--version]}}
```

ifdown

Disable network interfaces.

More information: https://manned.org/ifdown.

- Disable interface eth0:

```
ifdown {{eth0}}
```

- Disable all interfaces which are enabled:

```
ifdown {{[-a|--all]}}
```

ifmetric

An IPv4 route metrics manipulation tool.

More information: https://0pointer.de/lennart/projects/ifmetric/.

- Set the priority of the specified network interface (a higher number indicates lower priority):

```
sudo ifmetric {{interface}} {{value}}
```

- Reset the priority of the specified network interface:

```
sudo ifmetric {{interface}} {{0}}
```

ifstat

View network interface statistics.

More information: https://manned.org/ifstat.

- View network interface statistics since last query:

```
ifstat
```

- View network interface statistics since last boot:

```
ifstat {{[-a|--ignore]}}
```

- View error rate:

```
ifstat {{[-e|--errors]}}
```

ifstatus

Display OpenWRT interface status in JSON format.

More information: https://openwrt.org/docs/techref/netifd.

- Display interface status:

```
ifstatus {{interface_name}}
```

- Display help:

```
ifstatus
```

iftop

Show bandwidth usage on an interface by host.

More information: https://manned.org/iftop.

- Show the bandwidth usage:

```
sudo iftop
```

- Show the bandwidth usage of a given interface:

```
sudo iftop -i {{interface}}
```

- Show the bandwidth usage with port information:

```
sudo iftop -P
```

- Do not show bar graphs of traffic:

```
sudo iftop -b
```

- Do not look up hostnames:

```
sudo iftop -n
```

- Display help:

```
<?>
```

ifup

Enable network interfaces.

More information: https://manned.org/ifup.8.

- Enable interface eth0:

```
ifup {{eth0}}
```

- Enable all the interfaces defined with "auto" in `/etc/network/interfaces`:

```
ifup {{[-a|--all]}}
```

ikaros

Vanilla OS Tool for managing drivers for your device.

More information: https://github.com/Vanilla-OS/Ikaros.

- Interactively install drivers for your device:

```
ikaros install {{device}}
```

- Automatically install the recommended drivers for your device:

```
ikaros auto-install {{device}}
```

- List devices:

```
ikaros list-devices
```

img2txt

Convert images to colour ASCII characters and output them to text-based coloured files.

More information: https://manned.org/img2txt.

- Set output column count to a specific value:

```
img2txt {{[-W|--width]}} {{10}} {{path/to/image}}
```

- Set output line count to a specific value:

```
img2txt {{[-H|--height]}} {{5}} {{path/to/image}}
```

- Set output font width to a specific value:

```
img2txt {{[-x|--font-width]}} {{12}} {{path/to/image}}
```

- Set output font height to a specific value:

```
img2txt {{[-y|--font-height]}} {{14}} {{path/to/image}}
```

- Set image brightness to a specific value:

```
img2txt {{[-b|--brightness]}} {{2}} {{path/to/image}}
```

img2webp

Convert an image to WebP.

More information: https://developers.google.com/speed/webp/docs/img2webp.

- Convert an image to WebP:

```
img2webp {{path/to/image}} -o {{path/to/image.webp}}
```

imgp

Resize and rotate JPEG and PNG images.

More information: https://github.com/jarun/imgp.

- Convert single images and/or whole directories containing valid image formats:

```
imgp {{[-x|--res]}} {{1366x1000}} {{path/to/directory}} {{path/to/file}}
```

- Scale an image by 75% and overwrite the source image to a target resolution:

```
imgp {{[-x|--res]}} {{75}} z-w {{path/to/file}}
```

- Rotate an image clockwise by 90 degrees:

```
imgp {{[-o|--rotate]}} {{90}} {{path/to/file}}
```

impala

Manage Wi-Fi networks through a TUI.

More information: https://github.com/pythops/impala.

- Launch `impala` in station mode:

  `impala`

- Launch `impala` in Access Point mode:

  `impala {{[-m|--mode]}} ap`

- Switch between different sections:

  `{{<Tab>|<Shift Tab>}}`

- Select a network to connect to:

  `<Space>`

- Display hotkeys:

  `<?>`

importctl

Download, Import, or Export disk images.

More information: https://www.freedesktop.org/software/systemd/man/importctl.html.

- Download image in tarball format from a url via pull:

```
sudo importctl pull-tar {{URL}} {{path/to/directory}}
```

- Pull or download from a remote source that is either raw or qcow2 file, and stores it as a raw file:

```
sudo importctl pull-raw {{https://example.com/source.ext}} {{name}} --class={{machine|portable|sysext|confext}}
```

- Import a raw disk image into the image directory that is possibly compressed with xz, gzip, or bzip2:

```
importctl import-raw {{path/to/file.ext}} {{name}} --class={{machine|portable|sysext|confext}}
```

- Export a container image as tarball into current working directory:

```
importctl export-tar --class={{machine|portable|sysext|confext}} {{name}} {{path/to/file.ext}}
```

imv

CLI image viewer for wayland and X11 aimed at tiling window managers.

Handles multiple formats including Photoshop (PSD).

More information: https://sr.ht/~exec64/imv.

- View multiple images:

```
imv {{path/to/image1.ext path/to/image2.ext ...}}
```

- View in fullscreen mode:

```
imv -f {{path/to/image.ext}}
```

- View images [r]ecursively from a path:

```
imv -r --slideshow {{path/to/directory}}
```

- Open multiple images via `stdin`:

```
find . -type f -name "{{*.svg}}" | imv
```

- Make a slideshow from a directory showing each image for 10 seconds:

```
imv -t 10 {{path/to/directory}}
```

- View multiple images from the web:

```
curl {{[-Osw|--remote-name --silent --write-out]}} '%
{filename_effective}\n' '{{http://www.example.com/
[1-10].jpg}}' | imv
```

informant

Read Arch Linux package news.

More information: https://github.com/bradford-smith94/informant.

- Read all unread news:

```
sudo informant read
```

- Check for news:

```
informant check
```

- List latest news:

```
informant list
```

- Display help:

```
informant {{[-h|--help]}}
```

init

Linux run level manager.

Requires the SYSVINIT compile-time option to be enabled if using systemd.

More information: https://manned.org/init.8.

- Set the system to run a graphical environment:

```
sudo init 5
```

- Set the system to run multiuser terminal:

```
sudo init 3
```

- Shut down the system:

```
init 0
```

- Reboot the system:

```
init 6
```

- Set the system to run on terminal with only root user allowed and no networking:

```
sudo init 1
```

inotifywait

Waits for changes to files.

More information: https://manned.org/inotifywait.

- Watch a specific file for events, exiting after the first one:

```
inotifywait {{path/to/file}}
```

- Continuously watch a specific file for events without exiting:

```
inotifywait {{[-m|--monitor]}} {{path/to/file}}
```

- Watch a directory recursively for events:

```
inotifywait {{[-m|--monitor]}} {{[-r|--recursive]}} {{path/to/directory}}
```

- Watch a directory for changes, excluding files, whose names match a `regex`:

```
inotifywait {{[-m|--monitor]}} {{[-r|--recursive]}} --exclude "{{regex}}" {{path/to/directory}}
```

- Watch a file for changes, exiting when no event occurs for 30 seconds:

```
inotifywait {{[-m|--monitor]}} {{[-t|--timeout]}} {{30}} {{path/to/file}}
```

- Only watch a file for file modification events:

```
inotifywait {{[-e|--event]}} {{modify}} {{path/to/file}}
```

- Watch a file printing only events, and no status messages:

```
inotifywait {{[-q|--quiet]}} {{path/to/file}}
```

- Run a command when a file is accessed:

```
inotifywait {{[-e|--event]}} {{access}} {{path/to/file}} && {{command}}
```

inputattach

Attach a device to the Linux input system.

The input devices are presented as files in **`/dev/input/`**.

More information: https://manned.org/inputattach.

- Attach a Pulse8 CEC device to the input system:

```
inputattach --pulse8-cec {{/dev/ttyACM0}}
```

- Display help:

```
inputattach --help
```

insmod

Dynamically load modules into the Linux Kernel.

See also: **kmod** for other module management commands.

More information: https://manned.org/insmod.

- Insert a kernel module into the Linux kernel:

```
sudo insmod {{path/to/module.ko}}
```

installpkg

Install a Slackware package.

More information: http://www.slackware.com/config/packages.php.

- Install a package:

```
sudo installpkg {{path/to/package.tgz}}
```

- Generate a report of a simulated installation to `stdout`:

```
installpkg -warn {{path/to/package.tgz}}
```

- Create a package from current directory and its subdirectories:

```
installpkg -m {{package_name.tgz}}
```

- Install the contents of the current directory and subdirectories as a package with a specified name:

```
sudo installpkg -r {{package_name.tgz}}
```

instaloader

Download pictures, videos, captions, and other metadata from Instagram.

Note: You will need to provide Instagram login information if you want high-quality media downloads.

More information: https://instaloader.github.io/cli-options.html.

- Download a profile:

```
instaloader {{profile_name}}
```

- Download highlights:

```
instaloader --highlights {{profile_name}}
```

- Download posts with geotags (if available), suppressing any user interaction:

```
instaloader {{[-q|--quiet]}} {{[-G|--geotags]}}
{{profile_name}}
```

- Specify a user agent for HTTP requests:

```
instaloader --user-agent {{user_agent}} {{profile_name}}
```

- Specify login info and download posts (useful for private profiles):

```
instaloader {{[-l|--login]}} {{username}} {{[-p|--password]}}
{{password}} {{profile_name}}
```

- Skip a target if the first downloaded file has been found (useful for updating Instagram archives):

```
instaloader {{[-F|--fast-update]}} {{profile_name}}
```

- Download stories and IGTV videos (login required):

```
instaloader {{[-l|--login]}} {{username}} {{[-p|--password]}}
{{password}} {{[-s|--stories]}} --igtv {{profile_name}}
```

- Download all types of posts (login required):

```
instaloader {{[-l|--login]}} {{username}} {{[-p|--password]}}
{{password}} {{[-s|--stories]}} --igtv --highlights
{{profile_name}}
```

intercept

Read raw input events from a specified input event device and redirect it to **`stdout`**.

More information: https://gitlab.com/interception/linux/tools/-/tree/master#intercept.

- Read and output raw input events from a given input device file (the system will not see any key presses):

```
sudo intercept -g {{/dev/input/eventX}}
```

- Read and output raw input events from a given input device file (the system can see key presses and does not block other programs from reading them):

```
sudo intercept {{/dev/input/eventX}}
```

inxi

Print a summary of system information and resources for debugging purposes.

See also: **`lshw`**, **`hwinfo`**.

More information: https://manned.org/inxi.

- Print a summary of CPU, memory, hard drive and kernel information:

```
inxi
```

- Print a full description of CPU, memory, disk, network, and process information and filter sensitive information:

```
inxi {{[-ez|--expanded --filter]}}
```

- Print a summary of CPU information:

```
inxi {{[-C|--cpu]}}
```

- Print a summary of graphics information:

```
inxi {{[-G|--graphics]}}
```

- Print a summary of system RAM:

```
inxi {{[-m|--memory]}}
```

- Print a summary of system audio:

```
inxi {{[-A|--audio]}}
```

- Print available sensor data:

```
inxi {{[-s|--sensors]}}
```

- Print information about the distribution's repositories:

```
inxi {{[-r|--repos]}}
```

ionice

Get or set program I/O scheduling class and priority.

Scheduling classes: 1 (realtime), 2 (best-effort), 3 (idle).

Priority levels: 0 (the highest) - 7 (the lowest).

More information: https://manned.org/ionice.

- Run a command with the given scheduling class and priority:

```
ionice {{[-c|--class]}} {{scheduling_class}} {{[-n|--classdata]}} {{priority}} {{command}}
```

- Set I/O scheduling class of a running process with a specific [p]id, [P]gid or [u]id:

```
ionice {{[-c|--class]}} {{scheduling_class}} -{{p|P|u}} {{id}}
```

- Run a command with custom I/O scheduling class and priority:

```
ionice {{[-c|--class]}} {{scheduling_class}} {{[-n|--classdata]}} {{priority}} {{command}}
```

- Ignore failure to set the requested priority:

```
ionice {{[-t|--ignore]}} {{[-n|--classdata]}} {{priority}} {{[-p|--pid]}} {{pid}}
```

- Run the command even in case it was not possible to set the desired priority (this can happen due to insufficient privileges or an old kernel version):

```
ionice {{[-t|--ignore]}} {{[-n|--classdata]}} {{priority}} {{[-p|--pid]}} {{pid}}
```

- Print the I/O scheduling class and priority of a running process:

```
ionice {{[-p|--pid]}} {{pid}}
```

iostat

Report statistics for devices and partitions.

More information: https://manned.org/iostat.

- Display a report of CPU and disk statistics since system startup:

```
iostat
```

- Display a report of CPU and disk statistics with units converted to megabytes:

```
iostat -m
```

- Display CPU statistics:

```
iostat {{[-c|--compact]}}
```

- Display disk statistics with disk names (including LVM):

```
iostat -N
```

- Display extended disk statistics with disk names for device "sda":

```
iostat -xN {{sda}}
```

- Display incremental reports of CPU and disk statistics every 2 seconds:

```
iostat {{2}}
```

ip address

IP Address management subcommand.

More information: https://manned.org/ip-address.

- List network interfaces and their associated IP addresses:

```
ip {{[a|address]}}
```

- Filter to show only active network interfaces:

```
ip {{[a|address]}} {{[s|show]}} up
```

- Display information about a specific network interface:

```
ip {{[a|address]}} {{[s|show]}} {{ethX}}
```

- Add an IP address to a network interface:

```
sudo ip {{[a|address]}} {{[a|add]}} {{ip_address}} dev
{{ethX}}
```

- Remove an IP address from a network interface:

```
sudo ip {{[a|address]}} {{[d|delete]}} {{ip_address}} dev
{{ethX}}
```

- Delete all IP addresses in a given scope from a network interface:

```
sudo ip {{[a|address]}} {{[f|flush]}} {{ethX}} scope
{{global|host|link}}
```

ip link

Manage network interfaces.

More information: https://manned.org/ip-link.

- Show information about all network interfaces:

```
ip {{[l|link]}}
```

- Show information about a specific network interface:

```
ip {{[l|link]}} {{[sh|show]}} {{ethX}}
```

- Bring a network interface up or down:

```
sudo ip {{[l|link]}} {{[s|set]}} {{ethX}} {{up|down}}
```

- Give a meaningful name to a network interface:

```
sudo ip {{[l|link]}} {{[s|set]}} {{ethX}} {{[al|alias]}}
"{{LAN Interface}}"
```

- Change the MAC address of a network interface:

```
sudo ip {{[l|link]}} {{[s|set]}} {{ethX}} {{[a|address]}}
{{ff:ff:ff:ff:ff:ff}}
```

- Change the MTU size for a network interface to use jumbo frames:

```
sudo ip {{[l|link]}} {{[s|set]}} {{ethX}} mtu {{9000}}
```

- Set the promisc mode status of a device:

```
sudo ip {{[l|link]}} {{[s|set]}} {{ethX}} promisc {{on|off}}
```

ip maddress

Manage multicast addresses.

More information: https://manned.org/ip-maddress.

- List multicast addresses and how many programs are subscribed to them:

```
ip {{[m|maddress]}}
```

- List device specific addresses:

```
ip {{[m|maddress]}} {{[s|show]}} dev {{ethX}}
```

- Join a multicast group statically:

```
sudo ip {{[m|maddress]}} {{[a|add]}} {{33:33:00:00:00:02}}
dev {{ethX}}
```

- Leave a static multicast group:

```
sudo ip {{[m|maddress]}} {{[d|delete]}} {{33:33:00:00:00:02}}
dev {{ethX}}
```

- Display help:

```
ip {{[m|maddress]}} {{[h|help]}}
```

ip monitor

Monitor network for state changes.

More information: https://manned.org/ip-monitor.

- Monitor the whole network for state changes:

```
ip {{[mo|monitor]}}
```

- Specify the type to monitor:

```
ip {{[mo|monitor]}} {{link|address|route|neigh|rule|
maddress|...}}
```

- Replay an event file (can be generated with `rtmon`):

```
ip {{[mo|monitor]}} {{[f|file]}} {{path/to/file}}
```

ip neighbour

Neighbour/ARP tables management IP subcommand.

More information: https://manned.org/ip-neighbour.8.

- Display the neighbour/ARP table entries:

```
ip {{[n|neighbour]}}
```

- Remove entries in the neighbour table on device `ethX`:

```
sudo ip {{[n|neighbour]}} {{[f|flush]}} dev {{ethX}}
```

- Perform a neighbour lookup and return a neighbour entry:

```
ip {{[n|neighbour]}} {{[g|get]}} {{lookup_ip}} dev {{ethX}}
```

- Add or delete an ARP entry for the neighbour IP address to `ethX`:

```
sudo ip {{[n|neighbour]}} {{add|delete}} {{ip_address}}
lladdr {{mac_address}} dev {{ethX}} nud reachable
```

- Change or replace an ARP entry for the neighbour IP address to `ethX`:

```
sudo ip {{[n|neighbour]}} {{change|replace}} {{ip_address}}
lladdr {{new_mac_address}} dev {{ethX}}
```

ip netconf

Display configuration parameters.

More information: https://manned.org/ip-netconf.

- Show network configuration for all interfaces:

```
ip {{[netc|netconf]}}
```

- Show network configuration for a specific interface:

```
ip {{[netc|netconf]}} {{[s|show]}} dev {{network_interface}}
```

- Show only IPv4 network configuration:

```
ip -4 {{[netc|netconf]}}
```

- Show only IPv6 network configuration:

```
ip -6 {{[netc|netconf]}}
```

ip route add

Add a new networking route.

More information: https://manned.org/ip-route.

- Add a default route using gateway forwarding:

```
sudo ip {{[r|route]}} {{[a|add]}} default via {{gateway_ip}}
```

- Add a default route using `ethX`:

```
sudo ip {{[r|route]}} {{[a|add]}} default dev {{ethX}}
```

- Add a static route:

```
sudo ip {{[r|route]}} {{[a|add]}} {{destination_ip}} via {{gateway_ip}} dev {{ethX}}
```

- Add a route to a specific routing table:

```
sudo ip {{[r|route]}} {{[a|add]}} {{destination_ip}} dev {{ethX}} {{[t|table]}} {{ip}}
```

ip route get

Get a single route to a destination and print its contents exactly as the kernel sees it.

More information: https://manned.org/ip-route.

- Print route to a destination:

```
ip {{[r|route]}} {{[g|get]}} {{1.1.1.1}}
```

- Print route to a destination from a specific source address:

```
ip {{[r|route]}} {{[g|get]}} {{destination}} from {{source}}
```

- Print route to a destination for packets arriving on a specific interface:

```
ip {{[r|route]}} {{[g|get]}} {{destination}} iif {{ethX}}
```

- Print route to a destination, forcing output through a specific interface:

```
ip {{[r|route]}} {{[g|get]}} {{destination}} oif {{ethX}}
```

- Print route to a destination with a specified Type of Service (ToS):

```
ip {{[r|route]}} {{[g|get]}} {{destination}} tos {{0x10}}
```

- Print route to a destination using a specific VRF (Virtual Routing and Forwarding) instance:

```
ip {{[r|route]}} {{[g|get]}} {{destination}} vrf {{myvrf}}
```

ip route list

Display subcommand for IP Routing table management.

More information: https://manned.org/ip-route.

- Display the `main` routing table:

```
ip {{[r|route]}} {{[l|list]}}
```

- Display the main routing table (same as first example):

```
ip {{[r|route]}} {{[l|list]}} {{[t|table]}} {{main|254}}
```

- Display the local routing table:

```
ip {{[r|route]}} {{[l|list]}} {{[t|table]}} {{local|255}}
```

- Display all routing tables:

```
ip {{[r|route]}} {{[l|list]}} {{[t|table]}} {{all|unspec|0}}
```

- List routes from a given device only:

```
ip {{[r|route]}} {{[l|list]}} dev {{ethX}}
```

- List routes within a given scope:

```
ip {{[r|route]}} {{[l|list]}} {{[s|scope]}} link
```

- Display the routing cache:

```
ip {{[r|route]}} {{[l|list]}} {{[c|cache]}}
```

- Display only IPv6 or IPv4 routes:

```
ip {{-6|-4}} {{[r|route]}}
```

ip route show

This command is an alias of **`ip route list`**.

- View documentation for the original command:

```
tldr ip route list
```

ip route

IP Routing table management subcommand.

See also: **`routel`**.

More information: https://manned.org/ip-route.

- Display the `main` routing table:

```
ip {{[r|route]}}
```

- Add a default route using gateway forwarding:

```
sudo ip {{[r|route]}} {{[a|add]}} default via {{gateway_ip}}
```

- Add a default route using `ethX`:

```
sudo ip {{[r|route]}} {{[a|add]}} default dev {{ethX}}
```

- Add a static route:

```
sudo ip {{[r|route]}} {{[a|add]}} {{destination_ip}} via
{{gateway_ip}} dev {{ethX}}
```

- Delete a static route:

```
sudo ip {{[r|route]}} {{[d|delete]}} {{destination_ip}} dev
{{ethX}}
```

- Change or replace a static route:

```
sudo ip {{[r|route]}} {{change|replace}} {{destination_ip}}
via {{gateway_ip}} dev {{ethX}}
```

- Show which route will be used by the kernel to reach an IP address:

```
ip {{[r|route]}} {{[g|get]}} {{destination_ip}}
```

- Display a specific routing table:

```
ip {{[r|route]}} {{[l|list]}} {{[t|table]}} {{table_number}}
```

ip rule

IP routing policy database management.

More information: https://manned.org/ip-rule.

- Display the routing policy:

```
ip {{[ru|rule]}}
```

- Create a new generic routing rule with a higher priority than `main`:

```
sudo ip {{[ru|rule]}} {{[a|add]}} from all lookup {{100}}
```

- Add a new rule based on packet source addresses:

```
sudo ip {{[ru|rule]}} {{[a|add]}} from {{192.168.178.2/32}}
```

- Add a new rule based on packet destination addresses:

```
sudo ip {{[ru|rule]}} {{[a|add]}} to {{192.168.178.2/32}}
```

- Delete a rule based on packet source addresses:

```
sudo ip {{[ru|rule]}} {{[d|delete]}} from
{{192.168.178.2/32}}
```

- Remove all routing rules:

```
sudo ip {{[ru|rule]}} {{[f|flush]}}
```

- Save all rules to a file:

```
ip {{[ru|rule]}} {{[s|save]}} > {{path/to/ip_rules.dat}}
```

- Restore all rules from a file:

```
sudo ip < {{path/to/ip_rules.dat}} {{[ru|rule]}} {{[r|
restore]}}
```

ip stats

Manage and show interface statistics.

More information: https://manned.org/ip-stats.

- Show all interface statistics across all network devices:

```
ip {{[st|stats]}}
```

- Show statistics for a specific network interface:

```
ip {{[st|stats]}} show dev {{network_interface}}
```

- Show link-layer statistics (same as `ip -statistics link show`):

```
ip {{[st|stats]}} show group link
```

- Show hardware offload statistics for all devices:

```
ip {{[st|stats]}} show group offload
```

- Show offload statistics for a specific interface:

```
ip {{[st|stats]}} show dev {{network_interface}} group
offload
```

- Show a specific offload subgroup:

```
ip {{[st|stats]}} show dev {{network_interface}} group
offload subgroup {{l3_stats|cpu_hit|hw_stats_info}}
```

- Show address-family specific statistics (e.g. MPLS):

```
ip {{[st|stats]}} show group afstats subgroup {{mpls}}
```

- Enable Layer 3 hardware statistics collection on a device:

```
ip {{[st|stats]}} set dev {{network_interface}} l3_stats on
```

ip tuntap

Manage TUN/TAP virtual network interfaces.

More information: https://baturin.org/docs/iproute2/#ip-tuntap.

- Show all existing TUN/TAP devices:

```
ip {{[tunt|tuntap]}}
```

- Create a TUN device with a specific name:

```
sudo ip {{[tunt|tuntap]}} {{[a|add]}} {{[d|dev]}} {{tun0}}
{{[m|mode]}} {{[t|tun]}}
```

- Create a TAP device with a specific name:

```
sudo ip {{[tunt|tuntap]}} {{[a|add]}} {{[d|dev]}} {{tap0}}
{{[m|mode]}} {{[ta|tap]}}
```

- Delete a TUN or TAP device:

```
sudo ip {{[tunt|tuntap]}} {{[d|delete]}} {{[d|dev]}} {{tun0|
tap0}} {{[m|mode]}} {{tun|tap}}
```

- Set the owner (UID) of a TUN/TAP device:

```
sudo ip {{[tunt|tuntap]}} {{[a|add]}} {{[d|dev]}} {{tun0|
tap0}} {{[m|mode]}} {{tun|tap}} {{[u|user]}} {{username}}
```

- Set both owner (UID) and group (GID) for a TUN/TAP device:

```
sudo ip {{[tunt|tuntap]}} {{[a|add]}} {{[d|dev]}} {{tun0|
tap0}} {{[m|mode]}} {{tun|tap}} {{[u|user]}} {{username}}
{{[g|group]}} {{group_name}}
```

ip

Show/manipulate routing, devices, policy routing and tunnels.

Some subcommands such as **`address`** have their own usage documentation.

More information: https://manned.org/ip.8.

- List interfaces with detailed info:

```
ip {{[a|address]}}
```

- List interfaces with brief network layer info:

```
ip {{[-br|-brief]}} {{[a|address]}}
```

- List interfaces with brief link layer info:

```
ip {{[-br|-brief]}} {{[l|link]}}
```

- Display the routing table:

```
ip {{[r|route]}}
```

- Show neighbors (ARP table):

```
ip {{[n|neighbour]}}
```

- Make an interface up/down:

```
sudo ip {{[l|link]}} {{[s|set]}} {{ethX}} {{up|down}}
```

- Add/Delete an IP address to an interface:

```
sudo ip {{[a|address]}} {{add|delete}} {{ip}}/{{mask}} dev
{{ethX}}
```

- Add a default route:

```
sudo ip {{[r|route]}} {{[a|add]}} default via {{ip}} dev
{{ethX}}
```

ip6tables-restore

This command is an alias of **`iptables-restore`** for the IPv6 firewall.

- View documentation for the original command:

  ```
  tldr iptables-restore
  ```

ip6tables-save

This command is an alias of **`iptables-save`** for the IPv6 firewall.

- View documentation for the original command:

```
tldr iptables-save
```

ip6tables

This command is an alias of **`iptables`** for the IPv6 firewall.

- View documentation for the original command:

```
tldr iptables
```

ipcmk

Create IPC (Inter-process Communication) resources.

More information: https://manned.org/ipcmk.

- Create a shared memory segment:

```
ipcmk {{[-M|--shmem]}} {{segment_size_in_bytes}}
```

- Create a semaphore:

```
ipcmk {{[-S|--semaphore]}} {{element_size}}
```

- Create a message queue:

```
ipcmk {{[-Q|--queue]}}
```

- Create a shared memory segment with specific permissions (default is 0644):

```
ipcmk {{[-M|--shmem]}} {{segment_size_in_bytes}}
{{octal_permissions}}
```

ipcrm

Delete IPC (Inter-process Communication) resources.

More information: https://manned.org/ipcrm.

- Delete a shared memory segment by ID:

```
ipcrm {{[-m|--shmem-id]}} {{shmem_id}}
```

- Delete a shared memory segment by key:

```
ipcrm {{[-M|--shmem-key]}} {{shmem_key}}
```

- Delete an IPC queue by ID:

```
ipcrm {{[-q|--queue-id]}} {{ipc_queue_id}}
```

- Delete an IPC queue by key:

```
ipcrm {{[-Q|--queue-key]}} {{ipc_queue_key}}
```

- Delete a semaphore by ID:

```
ipcrm {{[-s|--semaphore-id]}} {{semaphore_id}}
```

- Delete a semaphore by key:

```
ipcrm {{[-S|--semaphore-key]}} {{semaphore_key}}
```

- Delete all IPC resources:

```
ipcrm {{[-a|--all]}}
```

ipcs

Show information about the usage of System V IPC facilities: shared memory segments, message queues, and semaphore arrays.

See also: **`lsipc`** for a more flexible tool, **`ipcmk`** for creating IPC facilities, and **`ipcrm`** for deleting them.

More information: https://manned.org/ipcs.

- Show information about all active IPC facilities:

```
ipcs
```

- Show information about active shared [m]emory segments, message [q]ueues or [s]empahore sets:

```
ipcs {{--shmems|--queues|--semaphores}}
```

- Show full details on the resource with a specific ID:

```
ipcs {{--shmems|--queues|--semaphores}} {{[-i|--id]}} {{resource_id}}
```

- Show limits in [b]ytes or in a human-readable format:

```
ipcs {{[-l|--limits]}} {{--bytes|--human}}
```

- Show summary about current usage:

```
ipcs {{[-u|--summary]}}
```

- Show creator's and owner's UIDs and PIDs for all IPC facilities:

```
ipcs {{[-c|--creator]}}
```

- Show the PID of the last operators for all IPC facilities:

```
ipcs {{[-p|--pid]}}
```

- Show last access times for all IPC facilities:

```
ipcs {{[-t|--time]}}
```

ipmitool

Interface with the Intelligent Platform Management Interface (IPMI).

More information: https://manned.org/ipmitool.

- Start the IPMI driver for local connections:

```
systemctl start ipmidrv
```

- Open IPMI shell on the local hardware:

```
sudo ipmitool shell
```

- Open IPMI shell on a remote host:

```
ipmitool -H {{ip_address}} -U {{user_name}} shell
```

ipset

Create IP sets for firewall rules.

More information: https://manned.org/ipset.

- Create an empty IP set which will contain IP addresses:

```
ipset create {{set_name}} hash:ip
```

- Destroy a specific IP set:

```
ipset destroy {{set_name}}
```

- Add an IP address to a specific set:

```
ipset add {{set_name}} {{192.168.1.25}}
```

- Delete a specific IP address from a set:

```
ipset del {{set_name}} {{192.168.1.25}}
```

- Save an IP set:

```
ipset save {{set_name}} > {{path/to/ip_set}}
```

iptables-restore

Restore the **`iptables`** IPv4 configuration.

Use **`ip6tables-restore`** to do the same for IPv6.

More information: https://manned.org/iptables-restore.

- Restore the `iptables` configuration from a file:

```
sudo iptables-restore {{path/to/file}}
```

iptables-save

Save the **`iptables`** IPv4 configuration.

Use **`ip6tables-save`** to do the same for IPv6.

More information: https://manned.org/iptables-save.

- Print the `iptables` configuration:

```
sudo iptables-save
```

- Print the `iptables` configuration of a specific table:

```
sudo iptables-save {{[-t|--table]}} {{table}}
```

- Save the `iptables` configuration to a file:

```
sudo iptables-save {{[-f|--file]}} {{path/to/file}}
```

iptables

Configure tables, chains and rules of the Linux kernel IPv4 firewall.

Use **`ip6tables`** to set rules for IPv6 traffic.

See also: **`iptables-save`**, **`iptables-restore`**.

More information: https://manned.org/iptables.

- View chains, rules, packet/byte counters and line numbers for the filter table:

```
sudo iptables {{[-vnL --line-numbers|--verbose --numeric --list --line-numbers]}}
```

- Set chain policy rule:

```
sudo iptables {{[-P|--policy]}} {{chain}} {{rule}}
```

- Append rule to chain policy for IP:

```
sudo iptables {{[-A|--append]}} {{chain}} {{[-s|--source]}} {{ip}} {{[-j|--jump]}} {{rule}}
```

- Append rule to chain policy for IP considering protocol and port:

```
sudo iptables {{[-A|--append]}} {{chain}} {{[-s|--source]}} {{ip}} {{[-p|--protocol]}} {{tcp|udp|icmp|...}} --dport {{port}} {{[-j|--jump]}} {{rule}}
```

- Add a NAT rule to translate all traffic from the `192.168.0.0/24` subnet to the host's public IP:

```
sudo iptables {{[-t|--table]}} {{nat}} {{[-A|--append]}} {{POSTROUTING}} {{[-s|--source]}} {{192.168.0.0/24}} {{[-j|--jump]}} {{MASQUERADE}}
```

- Delete chain rule:

```
sudo iptables {{[-D|--delete]}} {{chain}} {{rule_line_number}}
```

isoinfo

Utility programs for dumping and verifying ISO disk images.

More information: https://manned.org/isoinfo.

- List all the files included in an ISO image:

```
isoinfo -f -i {{path/to/image.iso}}
```

- E[x]tract a specific file from an ISO image and send it out `stdout`:

```
isoinfo -i {{path/to/image.iso}} -x {{/PATH/TO/FILE/INSIDE/ISO.EXT}}
```

- Show header information for an ISO disk image:

```
isoinfo -d -i {{path/to/image.iso}}
```

isosize

Display the size of an ISO file.

More information: https://manned.org/isosize.

- Display the size of an ISO file:

```
isosize {{path/to/file.iso}}
```

- Display the block count and block size of an ISO file:

```
isosize {{[-x|--sectors]}} {{path/to/file.iso}}
```

- Display the size of an ISO file divided by a given number (only usable when --sectors is not given):

```
isosize {{[-d|--divisor]}} {{number}} {{path/to/file.iso}}
```

iw dev

Show and manipulate wireless devices.

For a list of channels, frequencies and reg information: https://wireless.docs.kernel.org/en/latest/en/developers/documentation/channellist.html.

More information: https://wireless.docs.kernel.org/en/latest/en/users/documentation/iw.html.

- Set device to monitor mode (interface must be down first. See also: `ip link`):

```
sudo iw dev {{wlp}} set type monitor
```

- Set device to managed mode (interface must be down first):

```
sudo iw dev {{wlp}} set type managed
```

- Set device Wi-Fi channel (device must first be in monitor mode with the interface up):

```
sudo iw dev {{wlp}} set channel {{channel_number}}
```

- Set device Wi-Fi frequency in Mhz (device must first be in monitor mode with the interface up):

```
sudo iw dev {{wlp}} set freq {{freq_in_mhz}}
```

- Show all known station info:

```
iw dev {{wlp}} station dump
```

- Create a virtual interface in monitor mode with a specific MAC address:

```
sudo iw dev {{wlp}} interface add "{{vif_name}}" type monitor addr {{12:34:56:aa:bb:cc}}
```

- Delete virtual interface:

```
sudo iw dev "{{vif_name}}" del
```

iw

Show and manipulate wireless devices.

See also: **`iw dev`**, **`nmcli`**, **`iwctl`**.

More information: https://wireless.docs.kernel.org/en/latest/en/users/documentation/iw.html.

- Scan for available wireless networks:

```
iw dev {{wlp}} scan
```

- Join an open wireless network:

```
iw dev {{wlp}} connect {{SSID}}
```

- Close the current connection:

```
iw dev {{wlp}} disconnect
```

- Show information about the current connection:

```
iw dev {{wlp}} link
```

- List all physical and logical wireless network interfaces:

```
iw dev
```

- List all wireless capabilities for all physical hardware interfaces:

```
iw phy
```

- List the kernel's current wireless regulatory domain information:

```
iw reg get
```

- Display help:

```
iw help
```

iwconfig

Configure and show the parameters of a wireless network interface.

More information: https://manned.org/iwconfig.

- Show the parameters and statistics of all the interfaces:

```
iwconfig
```

- Show the parameters and statistics of the specified interface:

```
iwconfig {{interface}}
```

- Set the ESSID (network name) of the specified interface (e.g. eth0 or wlp2s0):

```
iwconfig {{interface}} {{new_network_name}}
```

- Set the operating mode of the specified interface:

```
iwconfig {{interface}} mode {{Ad-Hoc|Managed|Master|Repeater|Secondary|Monitor|Auto}}
```

iwctl

Control the **`iwd`** network supplicant.

See also: **`nmcli`**, **`iw`**.

More information: https://manned.org/iwctl.

- Run `iwctl` in interactive mode:

```
iwctl
```

- Display Wi-Fi stations:

```
iwctl station list
```

- Look for networks with a station:

```
iwctl station {{station}} scan
```

- Display the networks found by a station:

```
iwctl station {{station}} get-networks
```

- Connect to a network with a station, if credentials are needed they will be asked:

```
iwctl station {{station}} connect {{network_name}}
```

- Display help:

```
iwctl {{[-h|--help]}}
```

iwinfo

Retrieve information about wireless interfaces on OpenWrt.

More information: https://openwrt.org/docs/guide-developer/ubus/iwinfo.

- List all available wireless interfaces:

```
iwinfo
```

- Display detailed information on a specific wireless interface:

```
iwinfo {{interface}} info
```

- Scan for nearby wireless networks visible to the interface:

```
iwinfo {{interface}} scan
```

- List connected devices:

```
iwinfo {{interface}} assoclist
```

- List channels supported by the interface:

```
iwinfo {{interface}} freqlist
```

- List available transmit power levels for the interface:

```
iwinfo {{interface}} txpowerlist
```

- Display help:

```
iwinfo h
```

iwlist

Get detailed information from a wireless interface.

More information: https://manned.org/iwlist.8.

- Display the list of access points and ad-hoc cells in range:

```
iwlist {{wireless_interface}} scan
```

- Display available frequencies in the device:

```
iwlist {{wireless_interface}} frequency
```

- List the bit-rates supported by the device:

```
iwlist {{wireless_interface}} rate
```

- List the WPA authentication parameters currently set:

```
iwlist {{wireless_interface}} auth
```

- List all the WPA encryption keys set in the device:

```
iwlist {{wireless_interface}} wpakeys
```

- List the encryption key sizes supported and list all the encryption keys set in the device:

```
iwlist {{wireless_interface}} keys
```

- List the various power management attributes and modes of the device:

```
iwlist {{wireless_interface}} power
```

- List generic information elements set in the device (used for WPA support):

```
iwlist {{wireless_interface}} genie
```

jhead

Image timestamp and EXIF data manipulation.

More information: https://www.sentex.net/~mwandel/jhead/usage.html.

- Show all EXIF data:

```
jhead {{path/to/image.jpg}}
```

- Set the file's date and time to the EXIF create date (file creation date will be changed):

```
jhead -ft {{path/to/image.jpg}}
```

- Set the EXIF time to the file's date and time (EXIF data will be changed):

```
jhead -dsft {{path/to/image.jpg}}
```

- Rename all JPEG files based on the EXIF create date to `YYYY_MM_DD-HH_MM_SS.jpg`:

```
jhead -n%Y_%m_%d-%H_%M_%S *.jpg
```

- Rotate losslessly all JPEG images by 90, 180 or 270 based on the EXIF orientation tag:

```
jhead -autorot *.jpg
```

- Update all EXIF timestamps (Format: +- hour:minute:seconds) (example: forgot to change the camera's time zone - removing 1 hour from timestamps):

```
jhead -ta-1:00:00 *.jpg
```

- Remove all EXIF data (including thumbnails):

```
jhead -purejpg {{path/to/image.jpg}}
```

jobs

Shell builtin for viewing information about processes spawned by the current shell.

Options other than **`-l`** and **`-p`** are exclusive to Bash.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-jobs.

- View jobs spawned by the current shell:

```
jobs
```

- List jobs and their process IDs:

```
jobs -l
```

- Display information about jobs with changed status:

```
jobs -n
```

- Display only process IDs:

```
jobs -p
```

- Display running processes:

```
jobs -r
```

- Display stopped processes:

```
jobs -s
```

journalctl

Query the systemd journal.

See also: **dmesg**.

More information: https://www.freedesktop.org/software/systemd/man/journalctl.html.

- Show all messages with priority level 3 (errors) from this boot:

```
journalctl {{[-b|--boot]}} {{[-p|--priority]}} 3
```

- Delete journal logs which are older than 2 days:

```
journalctl --vacuum-time 2d
```

- Show only the last n lines and follow new messages (like `tail -f` for traditional syslog):

```
journalctl {{[-n|--lines]}} {{n}} {{[-f|--follow]}}
```

- Show all messages by a specific unit:

```
journalctl {{[-u|--unit]}} {{unit}}
```

- Show logs for a given unit since the last time it started:

```
journalctl _SYSTEMD_INVOCATION_ID=$(systemctl show --value --property=InvocationID {{unit}})
```

- Filter messages within a time range (either timestamp or placeholders like "yesterday"):

```
journalctl {{[-S|--since]}} {{now|today|yesterday|tomorrow}} {{[-U|--until]}} "{{YYYY-MM-DD HH:MM:SS}}"
```

- Show all messages by a specific process:

```
journalctl _PID={{pid}}
```

- Show all messages by a specific executable:

```
journalctl {{path/to/executable}}
```

jpegtran

Perform lossless transformation of JPEG files.

More information: https://manned.org/jpegtran.

- Mirror an image horizontally or vertically:

```
jpegtran {{[-f|-flip]}} {{horizontal|vertical}} {{path/to/image.jpg}} > {{path/to/output.jpg}}
```

- Rotate an image 90, 180 or 270 degrees clockwise:

```
jpegtran {{[-ro|-rotate]}} {{90|180|270}} {{path/to/image.jpg}} > {{path/to/output.jpg}}
```

- Transpose the image across the upper-left to lower right axis:

```
jpegtran -transpose {{path/to/image.jpg}} > {{path/to/output.jpg}}
```

- Transverse the image across the upper right to lower left axis:

```
jpegtran -transverse {{path/to/image.jpg}} > {{path/to/output.jpg}}
```

- Convert the image to grayscale:

```
jpegtran {{[-g|-grayscale]}} {{path/to/image.jpg}} > {{path/to/output.jpg}}
```

- Crop the image to a rectangular region of width `W` and height `H` from the upper-left corner, saving the output to a specific file:

```
jpegtran -crop {{W}}x{{H}} -outfile {{path/to/output.jpg}} {{path/to/image.jpg}}
```

- Crop the image to a rectangular region of width `W` and height `H`, starting at point `X` and `Y` from the upper-left corner:

```
jpegtran -crop {{W}}x{{H}}+{{X}}+{{Y}} {{path/to/image.jpg}} > {{path/to/output.jpg}}
```

just

A V8 JavaScript runtime for Linux.

More information: https://github.com/just-js/just.

- Start a REPL (interactive shell):

```
just
```

- Run a JavaScript file:

```
just {{path/to/file.js}}
```

- Evaluate JavaScript code by passing it as an argument:

```
just eval "{{code}}"
```

- Initialize a new project in a directory of the same name:

```
just init {{project_name}}
```

- Build a JavaScript application into an executable:

```
just build {{path/to/file.js}} --static
```

kde-builder

Easily build KDE components from its source repositories.

Drop-in replacement for **`kdesrc-build`**.

More information: https://kde-builder.kde.org/en/cmdline/supported-cmdline-params.html.

- Initialize `kde-builder`:

  ```
  kde-builder --initial-setup
  ```

- Compile a KDE component and its dependencies from the source (use `workspace` to compile Plasma desktop):

  ```
  kde-builder {{component_name1 component_name2 ...}}
  ```

- Compile a component without updating its local code and without compiling its dependencies:

  ```
  kde-builder {{[-SD|--no-src --no-include-dependencies]}} {{component_name}}
  ```

- Refresh the build directories before compiling:

  ```
  kde-builder {{[-r|--refresh-build]}} {{component_name}}
  ```

- Resume compilation from a specific dependency:

  ```
  kde-builder {{[-f|--resume-from]}} {{dependency_component}} {{component_name}}
  ```

- Run a component with a specified executable name:

  ```
  kde-builder --run {{executable_name}}
  ```

- Install login session:

  ```
  kde-builder --install-login-session-only
  ```

- Use system libraries in place of a component if it fails to build:

  ```
  kde-builder --no-stop-on-failure {{component_name}}
  ```

kde-inhibit

Inhibit various desktop functions while a command runs.

More information: https://invent.kde.org/plasma/kde-cli-tools/-/blob/master/kdeinhibit/main.cpp.

- Inhibit power management:

```
kde-inhibit --power {{command}} {{command_arguments}}
```

- Inhibit screen saver:

```
kde-inhibit --screenSaver {{command}} {{command_arguments}}
```

- Launch VLC, and inhibit color correction (night mode) while it's running:

```
kde-inhibit --colorCorrect {{vlc}}
```

kdesrc-build

Easily build KDE components from its source repositories.

More information: https://docs.kde.org/trunk5/en/kdesrc-build/kdesrc-build/index.html.

- Initialize `kdesrc-build`:

```
kdesrc-build --initial-setup
```

- Compile a KDE component and its dependencies from source:

```
kdesrc-build {{component_name}}
```

- Compile a component without updating its local code and without compiling its dependencies:

```
kdesrc-build --no-src --no-include-dependencies {{component_name}}
```

- Refresh the build directories before compiling:

```
kdesrc-build --refresh-build {{component_name}}
```

- Resume compilation from a specific dependency:

```
kdesrc-build --resume-from {{dependency_component}} {{component_name}}
```

- Run a component with a specified executable name:

```
kdesrc-build --run --exec {{executable_name}} {{component_name}}
```

- Build all configured components:

```
kdesrc-build
```

- Use system libraries in place of a component if it fails to build:

```
kdesrc-build --no-stop-on-failure {{component_name}}
```

kdialog

Show KDE dialog boxes from within shell scripts.

More information: https://develop.kde.org/docs/administration/kdialog/.

- Open a dialog box displaying a specific message:

```
kdialog --msgbox "{{message}}"
"{{optional_detailed_message}}"
```

- Open a question dialog with a `yes` and `no` button, returning `0` and `1`, respectively:

```
kdialog --yesno "{{message}}"
```

- Open a warning dialog with a `yes`, `no`, and `cancel` button, returning `0`, `1`, or `2` respectively:

```
kdialog --warningyesnocancel "{{message}}"
```

- Open an input dialog box and print the input to `stdout` when `OK` is pressed:

```
kdialog --inputbox "{{message}}" "{{optional_default_text}}"
```

- Open a dialog to prompt for a specific password and print it to `stdout`:

```
kdialog --password "{{message}}"
```

- Open a dialog containing a specific dropdown menu and print the selected item to `stdout`:

```
kdialog --combobox "{{message}}" "{{item1}}" "{{item2}}"
"{{...}}"
```

- Open a file chooser dialog and print the selected file's path to `stdout`:

```
kdialog --getopenfilename
```

- Open a progressbar dialog and print a D-Bus reference for communication to `stdout`:

```
kdialog --progressbar "{{message}}"
```

kdocker

Easily dock applications to the system tray.

More information: https://github.com/user-none/KDocker.

- Display a cursor to send a window to the system tray when pressing the left mouse button (press any other mouse button to cancel):

```
kdocker
```

- Open an application and send it to the system tray:

```
kdocker {{application}}
```

- Send focused window to the system tray:

```
kdocker -f
```

- Display a cursor to send a window to the system tray with a custom icon when pressing the left mouse button:

```
kdocker -i /{{path/to/icon}}
```

- Open an application, send it to the system tray and if focus is lost, minimize it:

```
kdocker -l {{application}}
```

- Display version:

```
kdocker --version
```

kernel-install

Add and remove kernel and initrd images to and from **`/boot`**.

More information: https://www.freedesktop.org/software/systemd/man/kernel-install.html.

- Add kernel and initramfs images to bootloader partition:

```
sudo kernel-install add {{kernel-version}} {{kernel-image}} {{path/to/initrd-file ...}}
```

- Remove kernel from the bootloader partition:

```
sudo kernel-install remove {{kernel-version}}
```

- Show various paths and parameters that have been configured or auto-detected:

```
sudo kernel-install inspect {{kernel-image}}
```

kexec

Directly reboot into a new kernel.

More information: https://manned.org/kexec.

- Load a new kernel:

```
kexec {{[-l|--load]}} {{path/to/kernel}} --initrd={{path/to/initrd}} --command-line={{arguments}}
```

- Load a new kernel with current boot parameters:

```
kexec {{[-l|--load]}} {{path/to/kernel}} --initrd={{path/to/initrd}} --reuse-cmdline
```

- Execute a currently loaded kernel:

```
kexec {{[-e|--exec]}}
```

- Unload current kexec target kernel:

```
kexec {{[-u|--unload]}}
```

keyctl

Manipulate the Linux kernel keyring.

More information: https://manned.org/keyctl.

- List keys in a specific keyring:

```
keyctl list {{target_keyring}}
```

- List current keys in the user default session:

```
keyctl list {{@us}}
```

- Store a key in a specific keyring:

```
keyctl add {{type_keyring}} {{key_name}} {{key_value}}
{{target_keyring}}
```

- Store a key with its value from `stdin`:

```
echo -n {{key_value}} | keyctl padd {{type_keyring}}
{{key_name}} {{target_keyring}}
```

- Put a timeout on a key:

```
keyctl timeout {{key_name}} {{timeout_in_seconds}}
```

- Read a key and format it as a hex-dump if not printable:

```
keyctl read {{key_name}}
```

- Read a key and format as-is:

```
keyctl pipe {{key_name}}
```

- Revoke a key and prevent any further action on it:

```
keyctl revoke {{key_name}}
```

keyd

Remap keys.

More information: https://manned.org/keyd.

- Start and enable the `keyd` service:

```
systemctl enable keyd --now
```

- Display keypress information:

```
sudo keyd monitor
```

- Reset bindings and reload the configuration files in `/etc/keyd`:

```
sudo keyd reload
```

- List all valid key names:

```
keyd list-keys
```

- Create a temporary binding:

```
sudo keyd bind "{{pressed_key}} = {{output_key}}"
```

kill

Sends a signal to a process, usually related to stopping the process.

All signals except for SIGKILL and SIGSTOP can be intercepted by the process to perform a clean exit.

More information: https://manned.org/kill.

- Terminate a program using the default SIGTERM (terminate) signal:

```
kill {{process_id}}
```

- List signal values and their corresponding names (to be used without the `SIG` prefix). The available options may depend on the `kill` implementation:

```
kill {{-l|-L|--table}}
```

- Terminate a background job:

```
kill %{{job_id}}
```

- Terminate a program using the SIGHUP (hang up) signal. Many daemons will reload instead of terminating:

```
kill -{{1|HUP}} {{process_id}}
```

- Terminate a program using the SIGINT (interrupt) signal. This is typically initiated by the user pressing `<Ctrl c>`:

```
kill -{{2|INT}} {{process_id}}
```

- Signal the operating system to immediately terminate a program (which gets no chance to capture the signal):

```
kill -{{9|KILL}} {{process_id}}
```

- Signal the operating system to pause a program until a SIGCONT ("continue") signal is received:

```
kill -{{17|STOP}} {{process_id}}
```

- Send a `SIGUSR1` signal to all processes with the given GID (group id):

```
kill -{{SIGUSR1}} -{{group_id}}
```

kinfocenter

KDE information center to show system information.

More information: https://manned.org/kinfocenter.

- Open the GUI:

```
kinfocenter
```

- List all possible KCM modules for `kinfocenter`:

```
kinfocenter --list
```

- Display help:

```
kinfocenter {{[-h|--help]}}
```

kjv

The word of God available right on your desktop.

More information: https://github.com/bontibon/kjv.

- Display books:

```
kjv -l
```

- Open a specific book:

```
kjv {{Genesis}}
```

- Open a specific chapter of a book:

```
kjv {{Genesis}} {{2}}
```

- Open a specific verse of a specific chapter of a book:

```
kjv {{John}} {{3}}:{{16}}
```

- Open a specific range of verses of a book's chapter:

```
kjv {{Proverbs}} {{3}}:{{1-6}}
```

- Display a specific range of verses of a book from different chapters:

```
kjv {{Matthew}} {{1}}:{{7}}-{{2}}:{{6}}
```

- Display all verses that match a pattern:

```
kjv /{{Plagues}}
```

- Display all verses that match a pattern in a specific book:

```
kjv {{1Jn}}/{{antichrist}}
```

kmod

Manage Linux kernel modules.

This program is usually called via its symlinks: **`lsmod`**, **`rmmod`**, **`insmod`**, **`modinfo`**, **`modprobe`**, and **`depmod`**.

See their respective pages for more information.

More information: https://manned.org/kmod.

- List currently loaded kernel modules:

  ```
  kmod list
  ```

- Display the static device nodes information provided by the modules of the currently running kernel:

  ```
  kmod static-nodes
  ```

knock

Port knocking client to open specific ports on firewall.

More information: https://manned.org/knock.

- Knock on ports using different protocols:

```
knock {{hostname}} {{portnumber}}:{{protocol}}
```

- Knock on port using UDP:

```
knock {{[-u|--udp]}} {{hostname}} {{portnumber}}
```

- Force usage of IPv4/IPv6:

```
knock {{-4|-6}} {{hostname}} {{portnumber}}
```

- Display errors and details of connection:

```
knock {{[-v|--verbose]}} {{hostname}} {{portnumber}}
```

knockd

Port knocking daemon to listen for port knocking and execute scripts.

More information: https://manned.org/knockd.

- Start knockd system daemon:

```
knockd {{[-d|--daemon]}}
```

- Use specified configuration file for knockd:

```
knockd {{[-c|--config]}} {{path/to/file}}.configuration
```

koji build

Build an RPM package.

More information: https://docs.pagure.org/koji.

- Build a package from `src.rpm`:

```
koji build {{target}} {{path/to/src.rpm}}
```

- Build a package from a SCM (Source Code Management) URL:

```
koji build {{target}} {{git+https://src.fedoraproject.org/
rpms/vim.git#e847a50297a216229050bf4db3d06a139104e7cf}}
```

- Perform a scratch build:

```
koji build {{target}} {{path/to/src.rpm}} --scratch
```

- Wait on the build, even if it's running in the background:

```
koji build {{target}} {{path/to/src.rpm}} --wait
```

- Don't wait on build:

```
koji build {{target}} {{path/to/src.rpm}} --nowait
```

- Display help:

```
koji build --help
```

koji buildinfo

Print basic information about a build.

More information: https://docs.pagure.org/koji.

- Print basic information:

```
koji buildinfo {{BuildID_or_NVR1 BuildID_or_NRV2 ...}}
```

- Print basic information with changelog:

```
koji buildinfo {{BuildID_or_NVR1 BuildID_or_NRV2 ...}} --changelog
```

- Display help:

```
koji buildinfo --help
```

koji download-build

Download a built package.

More information: https://docs.pagure.org/koji.

- Download all RPMs from a specific build:

```
koji download-build {{BuildID|RPM|NVR}}
```

- Download RPMs signed with the given key:

```
koji download-build {{BuildID|RPM|NVR}} --key {{key}}
```

- Only download RPMs for given arch:

```
koji download-build {{BuildID|RPM|NVR}} --arch
{{x86_64,aarch64,noarch,...}}
```

- Download the given RPM:

```
koji download-build {{RPM}} --rpm
```

- Display help:

```
koji download-build --help
```

koji tag-build

Apply a tag to one or more builds.

More information: https://docs.pagure.org/koji.

- Apply a tag to one or more builds:

```
koji tag-build {{tag}} {{NVR1 NVR2 ...}}
```

- Don't wait on task:

```
koji tag-build {{tag}} {{NVR1 NVR2 ...}} --nowait
```

- Force operation:

```
koji tag-build {{tag}} {{NVR1 NVR2 ...}} --force
```

- Display help:

```
koji tag-build --help
```

koji

Interact with kojihub.

More information: https://docs.pagure.org/koji.

- Execute a koji subcommand:

  ```
  koji {{subcommand}}
  ```

- Introduce yourself to test connectivity with koijhub:

  ```
  koji moshimoshi
  ```

- Display help about global options:

  ```
  koji --help
  ```

- Display help to get all available commands:

  ```
  koji help
  ```

- Display help for a specific subcommand (like `build`, `tag-build`, `download-build`, `buildinfo`,`call`, etc.):

  ```
  koji {{subcommand}} --help
  ```

- Display version:

  ```
  koji version
  ```

konsave

Save and apply your Linux customizations with just one command.

More information: https://github.com/Prayag2/konsave.

- Save the current configuration as a profile:

```
konsave {{[-s|--save]}} {{profile_name}}
```

- Apply a profile:

```
konsave {{[-a|--apply]}} {{profile_name}}
```

- Save the current configuration as a profile, overwriting existing profiles if they exist with the same name:

```
konsave {{[-s|--save]}} {{profile_name}} {{[-f|--force]}}
```

- List all profiles:

```
konsave {{[-l|--list]}}
```

- Remove a profile:

```
konsave {{[-r|--remove]}} {{profile_name}}
```

- Export a profile as a `.knsv` to the home directory:

```
konsave {{[-e|--export-profile]}} {{profile_name}}
```

- Import a `.knsv` profile:

```
konsave {{[-i|--import-profile]}} {{path/to/
profile_name.knsv}}
```

konsole

KDE's terminal emulator.

More information: https://docs.kde.org/stable5/en/konsole/konsole/command-line-options.html.

- Open the terminal in a specific directory:

```
konsole --workdir {{path/to/directory}}
```

- [e]xecute a specific command and don't close the window after it exits:

```
konsole --noclose -e "{{command}}"
```

- Open a new tab:

```
konsole --new-tab
```

- Open the terminal in the background and bring to the front when `<Ctrl Shift F12>` is pressed:

```
konsole --background-mode
```

kpackagetool5

KPackage Manager: install, list, remove Plasma packages.

More information: https://manned.org/kpackagetool5.

- List all known package types that can be installed:

```
kpackagetool5 --list-types
```

- Install the package from a directory:

```
kpackagetool5 {{[-t|--type]}} {{package_type}} {{[-i|--install]}} {{path/to/directory}}
```

- Update installed package from a directory:

```
kpackagetool5 {{[-t|--type]}} {{package_type}} {{[-u|--upgrade]}} {{path/to/directory}}
```

- List installed plasmoids (`--global` for all users):

```
kpackagetool5 {{[-t|--type]}} Plasma/Applet {{[-l|--list]}} {{[-g|--global]}}
```

- Remove a plasmoid by name:

```
kpackagetool5 {{[-t|--type]}} Plasma/Applet {{[-r|--remove]}} "{{name}}"
```

kpackagetool6

KPackage Manager: install, list, remove Plasma packages.

More information: https://manned.org/kpackagetool6.

- List all known package types that can be installed:

```
kpackagetool6 --list-types
```

- Install the package from a directory:

```
kpackagetool6 {{[-t|--type]}} {{package_type}} {{[-i|--install]}} {{path/to/directory}}
```

- Update installed package from a directory:

```
kpackagetool6 {{[-t|--type]}} {{package_type}} {{[-u|--upgrade]}} {{path/to/directory}}
```

- List installed plasmoids (`--global` for all users):

```
kpackagetool6 {{[-t|--type]}} Plasma/Applet {{[-l|--list]}} {{[-g|--global]}}
```

- Remove a plasmoid by name:

```
kpackagetool6 {{[-t|--type]}} Plasma/Applet {{[-r|--remove]}} "{{name}}"
```

kpartx

Create device maps from partition tables.

More information: https://manned.org/kpartx.

- Add partition mappings and print created mappings:

```
kpartx -av {{whole_disk.img}}
```

- Delete partition mappings:

```
kpartx -d {{whole_disk.img}}
```

- List partition mappings:

```
kpartx -l {{whole_disk.img}}
```

krdpserver

Start an RDP server.

More information: https://invent.kde.org/plasma/krdp#running-the-example-server.

- Start an RDP server with a username and password:

```
krdpserver {{[-u|--username]}} {{user_name}} {{[-p|--password]}} {{password}}
```

- Share a specific monitor marked by its index:

```
krdpserver {{[-u|--username]}} {{user_name}} {{[-p|--password]}} {{password}} --monitor {{n}}
```

kreadconfig5

Read KConfig entries for KDE Plasma.

More information: https://userbase.kde.org/KDE_System_Administration/Configuration_Files.

- Read a key from the global configuration:

```
kreadconfig5 --group {{group_name}} --key {{key_name}}
```

- Read a key from a specific configuration file:

```
kreadconfig5 --file {{path/to/file}} --group {{group_name}} --key {{key_name}}
```

- Check if systemd is used to start the Plasma session:

```
kreadconfig5 --file {{startkderc}} --group {{General}} --key {{systemdBoot}}
```

krfb-virtualmonitor

Create a virtual monitor and allow that monitor to be used with VNC.

More information: https://invent.kde.org/network/krfb.

- Create a virtual monitor:

```
krfb-virtualmonitor --resolution {{1920}}x{{1080}} --name
{{monitor_name}} --password {{password}} --port {{5900}}
```

kscreen-console

Query KScreen's status.

More information: https://manned.org/kscreen-console.

- Show all outputs and configuration files to attach to a bug report:

```
kscreen-console bug
```

- Show paths to KScreen configuration files:

```
kscreen-console config
```

- Show KScreen output information and configuration:

```
kscreen-console outputs
```

- Monitor for changes:

```
kscreen-console monitor
```

- Show the current KScreen configuration as JSON:

```
kscreen-console json
```

- Display help:

```
kscreen-console {{[-h|--help]}}
```

- Display help including Qt specific command-line options:

```
kscreen-console --help-all
```

kscreen-doctor

Change and manipulate the screen setup.

More information: https://invent.kde.org/plasma/libkscreen.

- Show display output information:

  `kscreen-doctor {{[-o|--outputs]}}`

- Set the rotation of a display output with an ID of 1 to the right:

  `kscreen-doctor {{output.1.rotation.right}}`

- Set the scale of a display output with an ID of `HDMI-2` to 2 (200%):

  `kscreen-doctor {{output.HDMI-2.scale.2}}`

ksvgtopng5

Convert SVG files to PNG format.

More information: https://invent.kde.org/plasma/kde-cli-tools/-/blob/master/ksvgtopng/ksvgtopng.cpp.

- Convert an SVG file (should be an absolute path) to PNG:

```
ksvgtopng5 {{width}} {{height}} {{path/to/file.svg}} {{output_filename.png}}
```

kwallet-query

Read and write to a KDE Wallet.

More information: https://manned.org/kwallet-query.

- List all entries in the `Passwords` folder of `kdewallet`:

```
kwallet-query {{kdewallet}} {{[-l|--list-entries]}}
```

- List all entries in a specific folder:

```
kwallet-query {{kdewallet}} {{[-l|--list-entries]}} {{[-f|--folder]}} {{folder_name}}
```

- List all available folders:

```
kwallet-query {{kdewallet}} {{[-l|--list-entries]}} {{[-f|--folder]}} ""
```

- Display help:

```
kwallet-query {{[-h|--help]}}
```

kwrite

Text editor of the KDE Desktop project.

See also: **kate**.

More information: https://docs.kde.org/stable5/en/kate/kwrite/command-line-options.html.

- Open a text file:

```
kwrite {{path/to/file}}
```

- Open multiple text files:

```
kwrite {{file1 file2 ...}}
```

- Open a text file with a specific encoding:

```
kwrite --encoding {{UTF-8}} {{path/to/file}}
```

- Open a text file and navigate to a specific line and column:

```
kwrite --line {{line_number}} --column {{column_number}} {{path/to/file}}
```

kwriteconfig5

Write KConfig entries for KDE Plasma.

More information: https://userbase.kde.org/KDE_System_Administration/Configuration_Files.

- Display help:

```
kwriteconfig5 --help
```

- Set a global configuration key:

```
kwriteconfig5 --group {{group_name}} --key {{key}} {{value}}
```

- Set a key in a specific configuration file:

```
kwriteconfig5 --file {{path/to/file}} --group {{group_name}} --key {{key}} {{value}}
```

- Delete a key:

```
kwriteconfig5 --group {{group_name}} --key {{key}} --delete
```

- Use systemd to start the Plasma session when available:

```
kwriteconfig5 --file {{startkderc}} --group {{General}} --key {{systemdBoot}} {{true}}
```

- Hide the title bar when a window is maximized (like Ubuntu):

```
kwriteconfig5 --file {{~/.config/kwinrc}} --group {{Windows}} --key {{BorderlessMaximizedWindows}} {{true}}
```

- Configure KRunner to open with the Meta (Command/Windows) global hotkey:

```
kwriteconfig5 --file {{~/.config/kwinrc}} --group {{ModifierOnlyShortcuts}} --key {{Meta}} "{{org.kde.kglobalaccel,/component/krunner_desktop,org.kde.kglobalaccel.Component,invokeShortcut,_launch}
```

laptop-detect

Attempt to determine if the script is running on a laptop or desktop.

More information: https://gitlab.com/debiants/laptop-detect.

- Return an exit status of 0 if the current device is likely a laptop, else returns 1:

```
laptop-detect
```

- Print the type of device that the current system is detected as:

```
laptop-detect --verbose
```

- Display version:

```
laptop-detect --version
```

larasail

Manage Laravel on Digital Ocean servers.

More information: https://github.com/thedevdojo/larasail.

- Set up the server with Laravel dependencies using the default PHP version:

  ```
  larasail setup
  ```

- Set up the server with Laravel dependencies using a specific PHP version:

  ```
  larasail setup {{php71}}
  ```

- Add a new Laravel site:

  ```
  larasail host {{domain}} {{path/to/site_directory}}
  ```

- Retrieve the Larasail user password:

  ```
  larasail pass
  ```

- Retrieve the Larasail MySQL password:

  ```
  larasail mysqlpass
  ```

last

View the last logged in users.

More information: https://manned.org/last.

- View last login infromation (e.g., username, terminal, boot time, kernel) of all users as read from `/var/log/wtmp`:

```
last
```

- List login information of a specific user:

```
last {{username}}
```

- Specify how many of the last logins to show:

```
last {{[-n|--limit]}} {{login_count}}
```

- Print the full date and time for entries and then display the hostname column last to prevent truncation:

```
last {{[-F|--fulltimes]}} {{[-a|--hostlast]}}
```

- View all logins by a specific user and show the IP address instead of the hostname:

```
last {{username}} {{[-i|--ip]}}
```

- List information since a specific time and date:

```
last {{[-s|--since]}} {{-7days}}
```

- View all recorded reboots (i.e., the last logins of the pseudo user "reboot"):

```
last reboot
```

- Display help:

```
last {{[-h|--help]}}
```

lastb

List last logged in users.

More information: https://manned.org/lastb.

- List last logged in users:

```
sudo lastb
```

- List all last logged in users since a given time:

```
sudo lastb {{[-s|--since]}} {{YYYY-MM-DD}}
```

- List all last logged in users until a given time:

```
sudo lastb {{[-t|--until]}} {{YYYY-MM-DD}}
```

- List all logged in users at a specific time:

```
sudo lastb {{[-p|--present]}} {{hh:mm}}
```

- List all last logged in users and translate the IP into a hostname:

```
sudo lastb {{[-d|--dns]}}
```

lastlog

Show the most recent login of all users or of a user.

More information: https://manned.org/lastlog.

- Display the most recent login of all users:

```
lastlog
```

- Display the lastlog record of the specified user:

```
lastlog --user {{username}}
```

- Display records older than 7 days:

```
lastlog --before 7
```

- Display records more recent than 3 days:

```
lastlog --time 3
```

latte-dock

Replacement dock for Plasma desktop.

More information: https://github.com/KDE/latte-dock.

- Clear QML cache:

```
latte-dock --clear-cache
```

- Import and load default layout on startup:

```
latte-dock --default-layout
```

- Load a specific layout on startup:

```
latte-dock --layout {{layout_name}}
```

- Import and load a specific layout:

```
latte-dock --import-layout {{path/to/file}}
```

lbu

Manage **apk** overlay files on a diskless Alpine Linux system.

Note: Subcommands like **include** write to **/etc**, which is stored in RAM. You need to run **lbu commit** to save the changes.

More information: https://wiki.alpinelinux.org/wiki/Alpine_local_backup.

- Commit changes to persistent storage (only files in `/etc` by default):

```
lbu {{[ci|commit]}}
```

- List files that would be saved using `commit`:

```
lbu {{[st|status]}}
```

- Display changes in tracked files that would be saved using `commit`:

```
lbu diff
```

- Include a specific file or directory in the `apk` overlay:

```
lbu {{[inc|include]}} {{path/to/file_or_directory}}
```

- Exclude a specific file or directory in `/etc` from the `apk` overlay:

```
lbu {{[ex|exclude]}} {{path/to/file_or_directory}}
```

- Display the list of manually included/excluded files:

```
lbu {{include|exclude}} -l
```

- List backups (previously created overlays):

```
lbu {{[lb|list-backup]}}
```

- Revert to a backup overlay:

```
lbu revert {{overlay_filename.tar.gz}}
```

lchage

Display or change user password policy.

More information: https://manned.org/lchage.

- Disable password expiration for the user:

```
sudo lchage --date -1 {{username}}
```

- Display the password policy for the user:

```
sudo lchage --list {{username}}
```

- Require password change for the user a certain number of days after the last password change:

```
sudo lchage --maxdays {{number_of_days}} {{username}}
```

- Start warning the user a certain number of days before the password expires:

```
sudo lchage --warndays {{number_of_days}} {{username}}
```

lci

LOLCODE interpreter written in C.

More information: https://github.com/justinmeza/lci.

- Run a LOLCODE file:

```
lci {{path/to/file}}
```

- Display help:

```
lci -h
```

- Display version:

```
lci -v
```

ldapdomaindump

Dump users, computers, groups, OS and membership information via LDAP to HTML, JSON and greppable output.

See also: **`ldapsearch`**.

More information: https://github.com/dirkjanm/ldapdomaindump#usage.

- Dump all information using the given LDAP account:

```
ldapdomaindump {{[-u|--user]}} {{domain}}\{{username}} {{[-
p|--password]}} {{password|ntlm_hash}} {{hostname|ip}}
```

- Dump all information, resolving computer hostnames:

```
ldapdomaindump {{[-r|--resolve]}} {{[-u|--user]}} {{domain}}\
{{username}} {{[-p|--password]}}{{password}} {{hostname|ip}}
```

- Dump all information, resolving computer hostnames with the selected DNS server:

```
ldapdomaindump {{[-r|--resolve]}} {{[-n|--dns-server]}}
{{domain_controller_ip}} {{[-u|--user]}} {{domain}}\
{{username}} {{[-p|--password]}}{{password}} {{hostname|ip}}
```

- Dump all information to the given directory without JSON output:

```
ldapdomaindump --no-json {{[-o|--outdir]}} {{path/to/
directory}} {{[-u|--user]}} {{domain}}\{{username}} {{[-p|--
password]}}{{password}} {{hostname|ip}}
```

ldconfig

Configure symlinks and cache for shared library dependencies.

More information: https://manned.org/ldconfig.

- Update symlinks and rebuild the cache (usually run when a new library is installed):

```
sudo ldconfig
```

- Update the symlinks for a given directory:

```
sudo ldconfig -n {{path/to/directory}}
```

- Print the libraries in the cache and check whether a given library is present:

```
ldconfig {{[-p|--print-cache]}} | grep {{library_name}}
```

ldd

Display shared library dependencies of a binary.

Do not use on an untrusted binary, use objdump for that instead.

More information: https://manned.org/ldd.

- Display shared library dependencies of a binary:

```
ldd {{path/to/binary}}
```

- Display all information about dependencies:

```
ldd {{[-v|--verbose]}} {{path/to/binary}}
```

- Display unused direct dependencies:

```
ldd {{[-u|--unused]}} {{path/to/binary}}
```

- Report missing data objects and perform data relocations:

```
ldd {{[-d|--data-relocs]}} {{path/to/binary}}
```

- Report missing data objects and functions, and perform relocations for both:

```
ldd {{[-r|--function-relocs]}} {{path/to/binary}}
```

lddd

Find broken library links on the system.

This tool is only available on Arch Linux.

More information: https://manned.org/lddd.

- Scan directories to find and list packages with broken library links that need to be rebuilt:

```
lddd
```

ledctl

Intel(R) Enclosure LED Control Application.

More information: https://manned.org/ledctl.

- Turn on the "Locate" LED for specified device(s):

```
sudo ledctl locate={{/dev/sda,/dev/sdb,...}}
```

- Turn off the "Locate" LED for specified device(s):

```
sudo ledctl locate_off={{/dev/sda,/dev/sdb,...}}
```

- Turn off the "Status" LED and "Failure" LED for specified device(s):

```
sudo ledctl off={{/dev/sda,/dev/sdb,...}}
```

- Turn off the "Status" LED, "Failure" LED and "Locate" LED for specified device(s):

```
sudo ledctl normal={{/dev/sda,/dev/sdb,...}}
```

legit

Complementary command-line interface for Git.

More information: https://frostming.github.io/legit.

- Switch to a specified branch, stashing and restoring unstaged changes:

```
git sw {{target_branch}}
```

- Synchronize current branch, automatically merging or rebasing, and stashing and unstashing:

```
git sync
```

- Publish a specified branch to the remote server:

```
git publish {{branch_name}}
```

- Remove a branch from the remote server:

```
git unpublish {{branch_name}}
```

- List all branches and their publication status:

```
git branches {{glob_pattern}}
```

- Remove the last commit from the history:

```
git undo --hard
```

lex

This command is an alias of **`flex`**.

- View documentation for the original command:

```
tldr flex
```

lftp

Sophisticated file transfer program.

More information: https://lftp.yar.ru/lftp-man.html.

- Connect to an FTP server:

```
lftp {{[-u|--user]}} {{username}} {{ftp.example.com}}
```

- Download multiple files (glob expression):

```
mget {{path/to/*.png}}
```

- Upload multiple files (glob expression):

```
mput {{path/to/*.zip}}
```

- Delete multiple files on the remote server:

```
mrm {{path/to/*.txt}}
```

- Rename a file on the remote server:

```
mv {{original_filename}} {{new_filename}}
```

- Download or update an entire directory:

```
mirror {{path/to/remote_directory}} {{path/to/
local_output_directory}}
```

- Upload or update an entire directory:

```
mirror {{[-R|--reverse]}} {{path/to/local_directory}} {{path/
to/remote_output_directory}}
```

libinput

Interface with libinput.

More information: https://wayland.freedesktop.org/libinput/doc/latest/.

- List all devices recognized by libinput:

```
sudo libinput list-devices
```

- Print all libinput events:

```
sudo libinput debug-events
```

- Show a GUI to visualize libinput's events:

```
sudo libinput debug-gui
```

- Debug tablet axis values:

```
sudo libinput debug-tablet
```

- Display help:

```
libinput {{[-h|--help]}}
```

libtool

A generic library support script that hides the complexity of using shared libraries behind a consistent, portable interface.

More information: https://www.gnu.org/software/libtool/manual/libtool.html#Invoking-libtool.

- Compile a source file into a `libtool` object:

```
libtool {{[c|compile]}} gcc {{[-c|--compile]}} {{path/to/source.c}} {{[-o|--output]}} {{path/to/source.lo}}
```

- Create a library or an executable:

```
libtool {{[l|link]}} gcc {{[-o|--output]}} {{path/to/library.lo}} {{path/to/source.lo}}
```

- Automatically set the library path so that another program can use uninstalled `libtool` generated programs or libraries:

```
libtool {{[e|execute]}} gdb {{path/to/program}}
```

- Install a shared library:

```
libtool {{[i|install]}} cp {{path/to/library.la}} {{path/to/installation_directory}}
```

- Complete the installation of `libtool` libraries on the system:

```
libtool {{[f|finish]}} {{path/to/installation_directory}}
```

- Delete installed libraries or executables:

```
libtool {{[u|uninstall]}} {{path/to/installed_library.la}}
```

- Delete uninstalled libraries or executables:

```
libtool {{[cl|clean]}} rm {{path/to/source.lo}} {{path/to/library.la}}
```

libtoolize

An **autotools** tool to prepare a package for using **libtool**.

It performs various tasks, including generating necessary files and directories to integrate **libtool** seamlessly into a project.

More information: https://www.gnu.org/software/libtool/manual/libtool.html#Invoking-libtoolize.

- Initialize a project for `libtool` by copying necessary files (avoiding symbolic links) and overwriting existing files if needed:

```
libtoolize {{[-cf|--copy --force]}}
```

libuser-lid

This command is an alias of **`lid`**.

- View documentation for the original command:

  ```
  tldr lid.libuser
  ```

lid

Display a user's groups or a group's users.

More information: https://manned.org/lid.8.

- List primary and secondary groups of a specific user:

```
sudo lid {{username}}
```

- List users of a specific group:

```
sudo lid --group {{group_name}}
```

lid

`lid` can refer to multiple commands with the same name.

- View documentation for the `libuser` tool:

  `tldr lid.libuser`

- View documentation for the `idutils` tool:

  `tldr lid.idutils`

light

Control the backlight of your screen.

More information: https://manned.org/light.

- Get the current backlight value in percent:

```
light
```

- Set the backlight value to 50 percent:

```
light -S {{50}}
```

- Reduce 20 percent from the current backlight value:

```
light -U {{20}}
```

- Add 20 percent to the current backlight value:

```
light -A {{20}}
```

limine-enroll-config

Embed or reset the BLAKE2B hash of **`limine.conf`** in the Limine EFI executable.

Used to ensure the configuration file has not been tampered with when Secure Boot is enabled.

More information: https://codeberg.org/Limine/Limine/src/branch/trunk/USAGE.md#secure-boot.

- Embed a config file's BLAKE2B hash into the Limine EFI executable:

```
limine-enroll-config {{path/to/BOOTX64.EFI}} {{blake2b_hash}}
```

- Remove the enrolled hash from the executable, disabling the config integrity check:

```
limine-enroll-config --reset {{path/to/BOOTX64.EFI}}
```

- Display help:

```
limine-enroll-config --help
```

limine-entry-tool

A helper script to manage Limine bootloader entries on UEFI systems.

More information: https://gitlab.com/Zesko/limine-entry-tool.

- Scan for other active UEFI boot entries and add them to the Limine menu:

```
limine-entry-tool --scan
```

- Add a new kernel boot entry with an initramfs and a kernel file:

```
limine-entry-tool --add "{{kernel_name}}" "{{path/to/
initramfs}}" "{{path/to/vmlinuz}}"
```

- Add a new Unified Kernel Image (UKI) boot entry:

```
limine-entry-tool --add-uki "{{kernel_name}}" "{{path/to/
uki.efi}}"
```

- Remove a kernel boot entry and its associated files from the ESP:

```
limine-entry-tool --remove "{{kernel_name}}"
```

- Remove an entire OS entry by its name or machine ID:

```
limine-entry-tool --remove-os "{{OS_name|machine_id}}"
```

- Add an EFI boot entry for an alternative bootloader (e.g., Windows):

```
limine-entry-tool --add-efi "{{EFI_entry_name}}" "{{path/to/
loader.efi}}"
```

limine-scan

This command is an alias of **`limine-entry-tool --scan`**.

- View documentation for the original command:

  ```
  tldr limine-entry-tool
  ```

limine-snapper-sync

Synchronize Limine boot menu entries with Btrfs snapshots managed by Snapper.

More information: https://gitlab.com/Zesko/limine-snapper-sync.

- Manually synchronize Snapper snapshots to create boot entries in Limine:

  `limine-snapper-sync`

- List the current snapshot entries managed by Limine:

  `limine-snapper-list`

- Display detailed information about bootable snapshots and their files:

  `limine-snapper-info`

- Restore the system from a selected bootable snapshot:

  `limine-snapper-restore`

- Start the watcher process to monitor for Snapper snapshot changes:

  `limine-snapper-watcher`

limine

An advanced, portable, multiprotocol bootloader.

See also: **`limine-enroll-config`**, **`limine-entry-tool`**, **`limine-snapper-sync`**.

More information: https://codeberg.org/Limine/Limine.

- Install Limine to an MBR-partitioned disk:

```
limine bios-install {{/dev/sdX}}
```

- Install Limine to a GPT-partitioned disk with a specific BIOS boot partition:

```
limine bios-install {{/dev/sdX}} {{partition_number}}
```

- Install Limine to a disk image file:

```
limine bios-install {{path/to/image.iso}}
```

- Install Limine for modern UEFI systems (requires the `limine-entry-tool` package):

```
limine-install
```

- Update boot entries after a kernel upgrade (requires the `limine-entry-tool` package):

```
limine-update
```

- Scan for other operating systems to add to the boot menu (requires the `limine-entry-tool` package):

```
limine-scan
```

line

Read a single line of input.

More information: https://manned.org/line.1.

- Read input:

```
line
```

links

Command-line text-only web browser.

See also: `links2'.

More information: https://links.twibright.com/user_en.html#subch-browser_calibration.

- Visit a website:

```
links {{https://example.com}}
```

- Apply restrictions for anonymous account:

```
links -anonymous {{https://example.com}}
```

- Enable Cookies (1 to enable):

```
links -enable-cookies {{0|1}} {{https://example.com}}
```

- Navigate forwards and backwards through the links on a page:

```
{{<ArrowUp>|<ArrowDown>}}
```

- Go forwards and backwards one page:

```
{{<ArrowLeft>|<ArrowRight>}}
```

- Exit:

```
<q><y>
```

links2

Command-line web browser with optional graphics support.

See also: **`links`**.

More information: https://links.twibright.com/user_en.html#subch-browser_calibration.

- Visit a website in graphics mode:

```
links2 -g {{https://example.com}}
```

linux-boot-prober

Probe a partition for bootable operating systems.

More information: https://github.com/MaddieM4/os-prober.

- Probe a partition:

```
sudo linux-boot-prober {{/dev/sdXY}}
```

linux32

This command is an alias of **`setarch linux32`**.

- View documentation for the original command:

  `tldr setarch`

linux64

This command is an alias of **`setarch linux64`**.

- View documentation for the original command:

  `tldr setarch`

lldpctl

Print link layer discovery protocol information.

More information: https://manned.org/lldpctl.

- Start the lldp daemon:

```
systemctl start lldpd
```

- Display collected neighbor data:

```
sudo lldpctl
```

- Display help:

```
lldpctl {{[-h|--help]}}
```

lnav

Advanced log file viewer to analyze logs with little to no setup.

More information: https://docs.lnav.org/en/latest/cli.html.

- View logs of a program, specifying log files, directories or URLs:

```
lnav {{path/to/log_or_directory|url}}
```

- View logs of a specific remote host (SSH passwordless login required):

```
lnav {{ssh}} {{user}}@{{host1.example.com}}:{{/var/log/syslog.log}}
```

- Validate the format of log files against the configuration and report any errors:

```
lnav -C {{path/to/log_directory}}
```

lndir

Create a shadow directory of symbolic links to another directory tree.

More information: https://manned.org/lndir.

- Create a shadow directory in the current directory:

```
lndir {{path/to/directory}}
```

loadkeys

Load the kernel keymap for the console.

See also: **`localectl`**.

More information: https://manned.org/loadkeys.

- Load a specific keyboard layout for the current console:

```
sudo loadkeys {{en|de|fi|dvorak|defkeymap|...}}
```

- Load a default keymap:

```
sudo loadkeys {{[-d|--default]}}
```

- Create a kernel source table:

```
loadkeys {{[-m|--mktable]}}
```

- Create a binary keymap:

```
loadkeys {{[-b|--bkeymap]}}
```

- Search and parse keymap without action:

```
loadkeys {{[-p|--parse]}}
```

- Load the keymap suppressing all output:

```
loadkeys {{[-q|--quiet]}}
```

- Set a keymap for a specific console:

```
sudo loadkeys {{[-C|--console]}} {{/dev/ttyN}} {{uk}}
```

- Load a keymap from the specified file for the console:

```
loadkeys {{[-C|--console]}} {{/dev/ttyN}} /{{path/to/file}}
```

locale-gen

Generate locales defined in **`/etc/locale.gen`**.

More information: https://manned.org/locale-gen.

- Generate locales:

```
locale-gen
```

- Do not delete undefined locales:

```
locale-gen --keep-existing
```

locale

Get locale-specific information.

More information: https://manned.org/locale.

- List all global environment variables describing the user's locale:

```
locale
```

- List all available locales:

```
locale {{[-a|--all-locales]}}
```

- Display all available locales and the associated metadata:

```
locale {{[-a|--all-locales]}} {{[-v|--verbose]}}
```

- Display the current date format:

```
locale date_fmt
```

localectl

Control the system locale and keyboard layout settings.

More information: https://www.freedesktop.org/software/systemd/man/localectl.html.

- Show the current settings of the system locale and keyboard mapping:

```
localectl
```

- List available locales:

```
localectl list-locales
```

- Set a system locale variable:

```
localectl set-locale {{LANG}}={{en_US.UTF-8}}
```

- List available keymaps:

```
localectl list-keymaps
```

- Set the system keyboard mapping for the console and X11:

```
localectl set-keymap {{us}}
```

localedef

Manage locale definition files.

More information: https://manned.org/localedef.

- List compiled locales:

```
localedef --list-archive
```

- Display help:

```
localedef {{[-?|--help]}}
```

locate

Find filenames quickly.

More information: https://manned.org/locate.

- Look for pattern in the database. Note: The database is recomputed periodically (usually weekly or daily):

```
locate "{{pattern}}"
```

- Look for a pattern case-insensitively:

```
locate {{[-i|--ignore-case]}} "{{pattern}}"
```

- Look for a file by its exact filename (a pattern containing no globbing characters is interpreted as `*pattern*`):

```
locate "*/{{filename}}"
```

- Recompute the database. You need to do it if you want to find recently added files:

```
sudo updatedb
```

logger

Add messages to the system log.

More information: https://manned.org/logger.

- Log a message to syslog:

```
logger {{message}}
```

- Take input from `stdin` and log to syslog:

```
echo {{log_entry}} | logger
```

- Send the output to a remote syslog server running at a given port. Default port is 514:

```
echo {{log_entry}} | logger {{[-n|--server]}} {{hostname}} {{[-P|--port]}} {{port}}
```

- Use a specific tag for every line logged. Default is the name of logged in user:

```
echo {{log_entry}} | logger {{[-t|--tag]}} {{tag}}
```

- Log messages with a given priority. Default is `user.notice`. See `man logger` for all priority options:

```
echo {{log_entry}} | logger {{[-p|--priority]}} {{user.warning}}
```

login

Initiates a session for a user.

More information: https://manned.org/login.

- Log in as a user:

```
login {{user}}
```

- Log in as user without authentication if user is preauthenticated:

```
login -f {{user}}
```

- Log in as user and preserve environment:

```
login -p {{user}}
```

- Log in as a user on a remote host:

```
login -h {{host}} {{user}}
```

loginctl

Manage the systemd login manager.

See also: **`lslogins`**.

More information: https://www.freedesktop.org/software/systemd/man/loginctl.html.

- Print all current sessions:

```
loginctl
```

- Print all properties of a specific session:

```
loginctl show-session {{session_id}} {{[-a|--all]}}
```

- Print all properties of a specific user:

```
loginctl show-user {{username}}
```

- Print a specific property of a user:

```
loginctl show-user {{username}} {{[-p|--property]}} {{property_name}}
```

- Execute a `loginctl` operation on a remote host:

```
loginctl list-users {{[-H|--host]}} {{hostname}}
```

- Log a user out on all of their sessions:

```
loginctl terminate-user {{username}}
```

- Display help:

```
loginctl {{[-h|--help]}}
```

logread

Read the **`logd`** ring buffer log.

More information: https://openwrt.org/docs/guide-user/base-system/log.essentials.

- Print the log:

```
logread
```

- Print `n` messages:

```
logread -l {{n}}
```

- Filter messages by (Keyword/`regex`):

```
logread -e {{pattern}}
```

- Print log messages as they happen:

```
logread -f
```

- Display help:

```
logread -h
```

logrotate

Rotates, compresses, and mails system logs.

More information: https://manned.org/logrotate.

- Trigger a run manually:

```
logrotate {{path/to/logrotate.conf}} --force
```

- Run using a specific command to mail reports:

```
logrotate {{path/to/logrotate.conf}} --mail {{/usr/bin/mail_command}}
```

- Run without using a state (lock) file:

```
logrotate {{path/to/logrotate.conf}} --state /dev/null
```

- Run and skip the state (lock) file check:

```
logrotate {{path/to/logrotate.conf}} --skip-state-lock
```

- Tell `logrotate` to log verbose output into the log file:

```
logrotate {{path/to/logrotate.conf}} --log {{path/to/log_file}}
```

logsave

Save the output of a command in a logfile.

More information: https://manned.org/logsave.

- Execute command with specified argument(s) and save its output to log file:

```
logsave {{path/to/logfile}} {{command}}
```

- Take input from `stdin` and save it in a log file:

```
logsave {{logfile}} -
```

- Append the output to a log file, instead of replacing its current contents:

```
logsave -a {{logfile}} {{command}}
```

- Show verbose output:

```
logsave -v {{logfile}} {{command}}
```

logwatch

Summarizes many different logs for common services (e.g. apache, pam_unix, sshd, etc.) in a single report.

More information: https://manned.org/logwatch.

- Analyze logs for a range of dates at a certain level of detail:

```
logwatch --range {{yesterday|today|all|help}} --detail {{low|medium|others}}'
```

- Restrict report to only include information for a selected service:

```
logwatch --range {{all}} --service {{apache|pam_unix|...}}
```

look

Display lines beginning with a prefix in a file.

Note: The lines in the file must be sorted.

See also: **`grep`**, **`sort`**.

More information: https://manned.org/look.

- Search for lines beginning with a specific prefix in a specific file:

```
look {{prefix}} {{path/to/file}}
```

- Case-insensitively search only on blank and alphanumeric characters:

```
look {{[-f|--ignore-case]}} {{[-d|--alphanum]}} {{prefix}} {{path/to/file}}
```

- Specify a string termination character (space by default):

```
look {{[-t|--terminate]}} {{,}}
```

- Search in `/usr/share/dict/words` (`--ignore-case` and `--alphanum` are assumed):

```
look {{prefix}}
```

- Search in `/usr/share/dict/web2` (`--ignore-case` and `--alphanum` are assumed):

```
look {{[-a|--alternative]}} {{prefix}}
```

lookandfeeltool

This command is an alias of **`plasma-apply-lookandfeel`**.

- View documentation for the original command:

  `tldr plasma-apply-lookandfeel`

losetup

Set up and control loop devices.

More information: https://manned.org/losetup.

- List loop devices with detailed info:

```
losetup {{[-a|--all]}}
```

- Attach a file to a given loop device:

```
sudo losetup {{/dev/loop}} /{{path/to/file}}
```

- Attach a file to a new free loop device and scan the device for partitions:

```
sudo losetup --show {{[-P|--partscan]}} {{[-f|--find]}} /
{{path/to/file}}
```

- Attach a file to a read-only loop device:

```
sudo losetup {{[-r|--read-only]}} {{/dev/loop}} /{{path/to/
file}}
```

- Detach all loop devices:

```
sudo losetup {{[-D|--detach-all]}}
```

- Detach a given loop device:

```
sudo losetup {{[-d|--detach]}} {{/dev/loop}}
```

lrunzip

This command is an alias of **`lrzip -d`**.

- View documentation for the original command:

  ```
  tldr lrzip
  ```

lrzip

A large file compression program.

See also: **`lrunzip`**, **`lrztar`**, **`lrzuntar`**.

More information: https://manned.org/lrzip.

- Compress a file with LZMA - slow compression, fast decompression:

```
lrzip {{path/to/file}}
```

- Compress a file with BZIP2 - good middle ground for compression/speed:

```
lrzip -b {{path/to/file}}
```

- Compress with ZPAQ - extreme compression, but very slow:

```
lrzip -z {{path/to/file}}
```

- Compress with LZO - light compression, extremely fast decompression:

```
lrzip -l {{path/to/file}}
```

- Compress a file and password protect/encrypt it:

```
lrzip -e {{path/to/file}}
```

- Override the number of processor threads to use:

```
lrzip -p {{8}} {{path/to/file}}
```

lrztar

A wrapper for **`lrzip`** to simplify compression of directories.

See also: **`tar`**, **`lrzuntar`**, **`lrunzip`**.

More information: https://manned.org/lrztar.

- Archive a directory with tar, then compress:

```
lrztar {{path/to/directory}}
```

- Same as above, with ZPAQ - extreme compression, but very slow:

```
lrztar -z {{path/to/directory}}
```

- Specify the output file:

```
lrztar -o {{path/to/file}} {{path/to/directory}}
```

- Override the number of processor threads to use:

```
lrztar -p {{8}} {{path/to/directory}}
```

- Force overwriting of existing files:

```
lrztar -f {{path/to/directory}}
```

lrzuntar

This command is an alias of **`lrztar -d`**.

- View documentation for the original command:

  ```
  tldr lrztar
  ```

lsattr

List file attributes on a Linux filesystem.

More information: https://manned.org/lsattr.

- Display the attributes of the files in the current directory:

```
lsattr
```

- List the attributes of files in a particular path:

```
lsattr {{path}}
```

- List file attributes recursively in the current and subsequent directories:

```
lsattr -R
```

- Show attributes of all the files in the current directory, including hidden ones:

```
lsattr -a
```

- Display attributes of directories in the current directory:

```
lsattr -d
```

lsb_release

Get LSB (Linux Standard Base) and distribution-specific information.

More information: https://manned.org/lsb_release.

- Print all available information:

```
lsb_release {{[-a|--all]}}
```

- Print a description (usually the full name) of the operating system:

```
lsb_release {{[-d|--description]}}
```

- Print only the operating system name (ID), in short format (omitting the field name):

```
lsb_release {{[-is|--id --short]}}
```

- Print the release number and codename of the distribution, in short format:

```
lsb_release {{[-rcs|--release --codename --short]}}
```

lsblk

List information about devices.

More information: https://manned.org/lsblk.

- List all storage devices in a tree-like format:

```
lsblk
```

- Also list empty devices:

```
lsblk {{[-a|--all]}}
```

- Print the SIZE column in bytes rather than in a human-readable format:

```
lsblk {{[-b|--bytes]}}
```

- Output info about filesystems:

```
lsblk {{[-f|--fs]}}
```

- Use ASCII characters for tree formatting:

```
lsblk {{[-i|--ascii]}}
```

- Output info about block-device topology:

```
lsblk {{[-t|--topology]}}
```

- Exclude the devices specified by the comma-separated list of major device numbers:

```
lsblk {{[-e|--exclude]}} {{1,7,...}}
```

- Display a customized summary using a comma-separated list of columns:

```
lsblk {{[-o|--output]}}
{{NAME,ROTA,SERIAL,MODEL,TRAN,TYPE,SIZE,FSTYPE,MOUNTPOINT,...}}
```

lscpu

Display information about the CPU architecture.

More information: https://manned.org/lscpu.

- Display information about all CPUs:

```
lscpu
```

- Display information in a table:

```
lscpu {{[-e|--extended]}}
```

- Display only information about online CPUs in a table:

```
lscpu {{[-e|--extended]}} {{[-b|--online]}}
```

- Display only information about offline CPUs in a table:

```
lscpu {{[-e|--extended]}} {{[-c|--offline]}}
```

- Display details about CPU caches:

```
lscpu {{[-C|--caches]}}
```

- Display information in JSON format:

```
lscpu {{[-J|--json]}}
```

lsdev

Display installed hardware detected in the **`/proc`** director.

More information: https://manned.org/lsdev.

- Show list of installed hardware:

```
lsdev
```

lsfd

List open files and the corresponding processes in Linux.

More information: https://manned.org/lsfd.

- List all open file descriptors:

```
lsfd
```

- List all files kept open by a specific program:

```
lsfd {{[-Q|--filter]}} 'PID == {{process_ID}}'
```

- Check what program has a specific file open:

```
lsfd {{[-Q|--filter]}} "NAME == '{{path/to/file}}'"
```

- List open IPv4 or IPv6 sockets:

```
lsfd {{-i4|-i6}}
```

- Display help:

```
lsfd {{[-h|--help]}}
```

lshw

List detailed information about hardware configurations as root user.

See also: **`inxi`**, **`hwinfo`**.

More information: https://ezix.org/project/wiki/HardwareLiSter.

- Launch the X11 GUI (if available):

```
sudo lshw -X
```

- List all hardware in tabular format:

```
sudo lshw -short
```

- List multiple class of hardware (all disks and storage controllers) in tabular format:

```
sudo lshw {{[-c|-class]}} disk {{[-c|-class]}} storage -short
```

- Save all network interfaces to an HTML/XML/JSON file:

```
sudo lshw {{[-c|-class]}} network -{{html|xml|json}} >
interfaces{{.html|.xml|.json}}
```

- List network interfaces without revealing sensitive information (IP addresses, serial numbers, etc.):

```
sudo lshw {{[-c|-class]}} network -sanitize
```

- List a particular class of hardware:

```
sudo lshw {{[-c|-class]}} {{system|bridge|memory|processor|
address|storage|disk|tape|bus|network|display|input|printer|
multimedia|communication|power|volume|generic}}
```

lsinitrd

Show the contents of an initramfs image.

See also: **dracut**.

More information: https://github.com/dracutdevs/dracut/blob/master/man/lsinitrd.1.asc.

- Show the contents of the initramfs image for the current kernel:

```
lsinitrd
```

- Show the contents of the initramfs image for the specified kernel:

```
lsinitrd --kver {{kernel_version}}
```

- Show the contents of the specified initramfs image:

```
lsinitrd {{path/to/initramfs.img}}
```

- List modules included in the initramfs image:

```
lsinitrd --mod
```

- Unpack the initramfs to the current directory:

```
lsinitrd --unpack
```

lsipc

Show information on System V IPC facilities currently employed in the system.

See also: **`ipcs`** for the older tool.

More information: https://manned.org/lsipc.

- Show information about all active IPC facilities:

```
lsipc
```

- Show information about active shared [m]emory segments, message [q]ueues or [s]empahore sets:

```
lsipc {{--shmems|--queues|--semaphores}}
```

- Show full details on the resource with a specific ID:

```
lsipc {{--shmems|--queues|--semaphores}} {{[-i|--id]}}
{{resource_id}}
```

- Print the given output columns (see all supported columns with `--help`):

```
lsipc {{[-o|--output]}}
{{KEY,ID,PERMS,SEND,STATUS,NSEMS,RESOURCE,...}}
```

- Use [r]aw, [J]SON, [l]ist or [e]xport (key="value") format:

```
lsipc {{--raw|--json|--list|--export}}
```

- Don't truncate the output:

```
lsipc --notruncate
```

lslocks

List local system locks.

More information: https://manned.org/lslocks.

- List all local system locks:

```
lslocks
```

- List locks with defined column headers:

```
lslocks {{[-o|--output]}} {{PID}},{{COMMAND}},{{PATH}}
```

- List locks producing a raw output (no columns), and without column headers:

```
lslocks {{[-r|--raw]}} {{[-n|--noheadings]}}
```

- List locks by PID input:

```
lslocks {{[-p|--pid]}} {{PID}}
```

- List locks with JSON output to `stdout`:

```
lslocks {{[-J|--json]}}
```

lslogins

Show information about users on a Linux system.

More information: https://manned.org/lslogins.

- Display users in the system:

```
lslogins
```

- Display users belonging to a specific group:

```
lslogins {{[-g|--groups]}} {{groups}}
```

- Display user accounts:

```
lslogins {{[-u|--user-accs]}}
```

- Display last logins:

```
lslogins {{[-L|--last]}}
```

- Display system accounts:

```
lslogins {{[-s|--system-accs]}}
```

- Display supplementary groups:

```
lslogins {{[-G|--supp-groups]}}
```

lsmem

List the characteristics of the system's memory, including size and type of memory modules.

More information: https://manned.org/lsmem.

- List memory information:

```
lsmem
```

- List memory information in JSON format:

```
lsmem {{[-J|--json]}}
```

- List memory information with size in bytes:

```
lsmem {{[-b|--bytes]}}
```

lsmod

Show the status of Linux kernel modules.

See also: **kmod** for other module management commands.

More information: https://manned.org/lsmod.

- List all currently loaded kernel modules:

```
lsmod
```

lsns

List information about all namespaces or about the specified namespace.

More information: https://manned.org/lsns.

- List all namespaces:

```
lsns
```

- List namespaces in JSON format:

```
lsns {{[-J|--json]}}
```

- List namespaces associated with the specified process:

```
lsns {{[-p|--task]}} {{pid}}
```

- List the specified type of namespaces only:

```
lsns {{[-t|--type]}} {{mnt|net|ipc|user|pid|uts|cgroup|time}}
```

- List namespaces, only showing the namespace ID, type, PID, and command:

```
lsns {{[-o|--output]}} {{NS,TYPE,PID,COMMAND}}
```

lspci

List all PCI devices.

More information: https://manned.org/lspci.

- Show a brief list of devices:

```
lspci
```

- Display [v]erbose information (Note: the `-v` flag can be repeated to increase verbosity):

```
lspci -v
```

- Display [k]ernel drivers and modules handling each device:

```
lspci -k
```

- [s]elect a specific device:

```
lspci -s {{00:18.3}}
```

- Dump info in a ([m]achine) readable form:

```
lspci -vm
```

- Show PCI vendor and device codes as both [n]umbers and [n]ames:

```
lspci -nn
```

lsscsi

List SCSI devices (or hosts) and their attributes.

More information: https://manned.org/lsscsi.

- List all SCSI devices:

```
lsscsi
```

- List all SCSI devices with detailed attributes:

```
lsscsi {{[-L|--list]}}
```

- List all SCSI devices with human-readable disk capacity:

```
lsscsi {{[-s|--size]}}
```

lsusb

Display information about USB buses and devices connected to them.

More information: https://manned.org/lsusb.

- List all the USB devices available:

```
lsusb
```

- List the USB hierarchy as a tree:

```
lsusb {{[-t|--tree]}}
```

- List verbose information about USB devices:

```
sudo lsusb {{[-v|--verbose]}}
```

- List detailed information about a USB device:

```
sudo lsusb {{[-v|--verbose]}} -s {{bus}}:{{device number}}
```

- List devices with a specified vendor and product ID only:

```
lsusb -d {{vendor}}:{{product}}
```

ltrace

Display dynamic library calls of a process.

More information: https://manned.org/ltrace.

- Print (trace) library calls of a program binary:

```
ltrace ./{{program}}
```

- Count library calls. Print a handy summary at the bottom:

```
ltrace -c {{path/to/program}}
```

- Trace calls to malloc and free, omit those done by libc:

```
ltrace -e malloc+free-@libc.so* {{path/to/program}}
```

- Write to file instead of terminal:

```
ltrace {{[-o|--output]}} {{file}} {{path/to/program}}
```

lvchange

Change attributes or the activation state of logical volumes.

More information: https://manned.org/lvchange.

- Activate a logical volume:

```
lvchange {{[-a|--activate]}} y {{/dev/vg_name/lv_name}}
```

- Deactivate a logical volume:

```
lvchange {{[-a|--activate]}} n {{/dev/vg_name/lv_name}}
```

- Enable autoactivation for a logical volume:

```
lvchange {{[-a|--activate]}} ay {{/dev/vg_name/lv_name}}
```

- Set a logical volume to read-only (use `rw` for read-write):

```
lvchange {{[-p|--permission]}} r {{/dev/vg_name/lv_name}}
```

- Skip activation for a logical volume:

```
lvchange {{[-k|--setactivationskip]}} y {{/dev/vg_name/
lv_name}}
```

- Refresh a logical volume using the latest metadata:

```
lvchange --refresh {{/dev/vg_name/lv_name}}
```

lvconvert

Convert or modify the type, redundancy, or state of logical volumes.

More information: https://manned.org/lvconvert.

- Convert a linear logical volume to RAID1 (add one mirror: two copies total):

```
lvconvert --type raid1 {{[-m|--mirrors]}} 1 {{/dev/vg_name/lv_name}}
```

- Remove mirroring and convert back to a linear logical volume:

```
lvconvert {{[-m|--mirrors]}} 0 {{/dev/vg_name/lv_name}}
```

- Merge a snapshot back into its origin logical volume (applies on next activation):

```
lvconvert --merge {{/dev/vg_name/snapshot_lv}}
```

- Repair a degraded RAID logical volume:

```
lvconvert --repair {{/dev/vg_name/lv_name}}
```

- Convert an existing logical volume into a thin pool with a separate metadata LV:

```
lvconvert --type thin-pool --poolmetadata {{/dev/vg_name/pool_metadata_lv}} {{/dev/vg_name/pool_lv}}
```

lvcreate

Create a logical volume in an existing volume group. A volume group is a collection of logical and physical volumes.

See also: **`lvm`**.

More information: https://manned.org/lvcreate.

- Create a logical volume of 10 gigabytes in the volume group vg1:

```
lvcreate {{[-L|--size]}} {{10G}} {{vg1}}
```

- Create a 1500 megabyte linear logical volume named mylv in the volume group vg1:

```
lvcreate {{[-L|--size]}} {{1500}} {{[-n|--name]}} {{mylv}} {{vg1}}
```

- Create a logical volume called mylv that uses 60% of the total space in volume group vg1:

```
lvcreate {{[-l|--extents]}} {{60%VG}} {{[-n|--name]}} {{mylv}} {{vg1}}
```

- Create a logical volume called mylv that uses all the unallocated space in the volume group vg1:

```
lvcreate {{[-l|--extents]}} {{100%FREE}} {{[-n|--name]}} {{mylv}} {{vg1}}
```

lvdisplay

Display information about Logical Volume Manager (LVM) logical volumes.

See also: **`lvm`**, **`lvs`**.

More information: https://manned.org/lvdisplay.

- Display information about all logical volumes:

  ```
  sudo lvdisplay
  ```

- Display the information in a short format (same as running `lvs`):

  ```
  sudo lvdisplay {{[-C|--columns]}}
  ```

- Display information about all logical volumes in volume group vg1:

  ```
  sudo lvdisplay {{vg1}}
  ```

- Display information about logical volume lv1 in volume group vg1:

  ```
  sudo lvdisplay {{vg1/lv1}}
  ```

lvextend

Increase the size of a logical volume.

See also: **`lvm`**.

More information: https://manned.org/lvextend.8.

- Increase a volume's size to 120 GB:

```
sudo lvextend {{[-L|--size]}} {{120G}} {{logical_volume}}
```

- Increase a volume's size by 40 GB as well as the underlying filesystem:

```
sudo lvextend {{[-L|--size]}} +{{40G}} {{[-r|--resizefs]}}
{{logical_volume}}
```

- Increase a volume's size to 100% of the free physical volume space:

```
sudo lvextend {{[-l|--extents]}} +{{100}}%FREE
{{logical_volume}}
```

- Increase a volume's size to 100% of the free physical volume space and resize the underlying filesystem:

```
sudo lvextend {{[-l|--extents]}} +{{100}}%FREE {{[-r|--
resizefs]}} {{logical_volume}}
```

lvm

Manage physical volumes, volume groups, and logical volumes using the Logical Volume Manager (LVM) interactive shell.

More information: https://manned.org/lvm.

- Start the Logical Volume Manager interactive shell:

```
sudo lvm
```

- Initialize a drive or partition to be used as a physical volume:

```
sudo lvm pvcreate {{/dev/sdXY}}
```

- Display information about physical volumes:

```
sudo lvm pvdisplay
```

- Create a volume group called vg1 from the physical volume on `/dev/sdXY`:

```
sudo lvm vgcreate {{vg1}} {{/dev/sdXY}}
```

- Display information about volume groups:

```
sudo lvm vgdisplay
```

- Create a logical volume with size 10G from volume group vg1:

```
sudo lvm lvcreate {{[-L|--size]}} {{10G}} {{vg1}}
```

- Display information about logical volumes:

```
sudo lvm lvdisplay
```

- Display help for a specific command:

```
lvm help {{command}}
```

lvm_import_vdo

Import a VDO volume created by the VDO manager into an LVM-managed logical volume (irreversible).

More information: https://manned.org/lvm_import_vdo.

- Import a VDO volume with automatic names for the VG/LV:

```
lvm_import_vdo {{/dev/mapper/vdo_volume}}
```

- Import and set the destination VG/LV name:

```
lvm_import_vdo {{[-n|--name]}} {{vg_name/lv_name}} {{/dev/mapper/vdo_volume}}
```

- Show what would be done without changing anything:

```
lvm_import_vdo --dry-run {{/dev/mapper/vdo_volume}}
```

- Convert in place without using a temporary snapshot (less safe):

```
lvm_import_vdo --no-snapshot {{/dev/mapper/vdo_volume}}
```

- Verbose output and automatically answer "yes" to prompts:

```
lvm_import_vdo {{[-v|--verbose]}} {{[-y|--yes]}} {{/dev/mapper/vdo_volume}}
```

- Use a VDO manager configuration file during import:

```
lvm_import_vdo --vdo-config {{path/to/vdo.conf}} {{/dev/mapper/vdo_volume}}
```

lvmconfig

Display and manipulate LVM configuration information.

More information: https://manned.org/lvmconfig.

- Display the effective configuration in use (after merging all config sources):

```
lvmconfig --typeconfig current --mergedconfig
```

- Show only settings that differ from their defaults:

```
lvmconfig --typeconfig diff
```

- List all configuration keys:

```
lvmconfig {{[-l|--list]}}
```

- Print the default configuration with full comments and extra spacing:

```
lvmconfig --typeconfig default --withcomments --withspaces
```

- Validate the full merged configuration and report errors:

```
lvmconfig --mergedconfig --validate
```

- Write the current effective configuration to a file:

```
lvmconfig --typeconfig current {{[-f|--file]}} {{path/to/output.conf}}
```

lvmdevices

Manage the LVM devices file that lists block devices allowed for physical volumes.

More information: https://manned.org/lvmdevices.

- List devices recorded in the devices file:

```
lvmdevices
```

- Add a device to the devices file:

```
lvmdevices --adddev {{/dev/sdXN}}
```

- Remove a device from the devices file:

```
lvmdevices --deldev {{/dev/sdXN}}
```

- Add a physical volume by its PVID:

```
lvmdevices --addpvid {{PVID}}
```

- Remove a physical volume by its PVID:

```
lvmdevices --delpvid {{PVID}}
```

- Update the devices file after device names change:

```
lvmdevices --update
```

- Check the devices file for problems:

```
lvmdevices --check
```

- Display version:

```
lvmdevices --version
```

lvmdiskscan

Scan for devices that may be used as physical volumes by LVM (deprecated; prefer **pvs**).

More information: https://manned.org/lvmdiskscan.

- Scan all devices:

```
lvmdiskscan
```

- Show only physical volumes (PVs):

```
lvmdiskscan {{[-l|--lvmpartition]}}
```

- Increase verbosity (repeat for more detail):

```
lvmdiskscan {{[-v|--verbose]}}
```

lvmdump

Collect diagnostic information about LVM2 (Logical Volume Manager).

By default, outputs a compressed tar archive with system and configuration data in the home directory.

More information: https://manned.org/lvmdump.

- Generate a basic dump:

```
lvmdump
```

- Generate an extended dump with metadata and daemon info:

```
lvmdump -a -l -m
```

- Dump the information into a directory instead of a tarball:

```
lvmdump -d {{path/to/directory}}
```

- Display help:

```
lvmdump -h
```

lvmpersist

Manage persistent reservations (PR) on block devices or all PVs in a volume group.

More information: https://manned.org/lvmpersist.

- Start PR on all PVs in a VG with a local key (exclusive access by default):

```
lvmpersist start --ourkey {{0x1234abcd}} --vg {{vg_name}}
```

- Start PR for a shared VG (allow multiple hosts):

```
lvmpersist start --ourkey {{0x1234abcd}} --access {{sh}} --vg
{{vg_name}}
```

- Stop PR on a VG and unregister the local key:

```
lvmpersist stop --ourkey {{0x1234abcd}} --vg {{vg_name}}
```

- Take over a local VG by preempting another host while starting PR:

```
lvmpersist start --ourkey {{0xmy_key}} --removekey
{{0xother_key}} --vg {{vg_name}}
```

- Remove another host's key from a shared VG:

```
lvmpersist remove --ourkey {{0xmy_key}} --removekey
{{0xother_key}} --vg {{vg_name}}
```

- Show registered keys and reservations for a VG:

```
lvmpersist read --vg {{vg_name}}
```

- Operate on specific devices instead of a VG:

```
lvmpersist start --ourkey {{0x1234abcd}} --device {{/dev/
sdX}} --device {{/dev/mapper/mpathY}}
```

lvmpolld

LVM poll daemon that supervises long-running LVM operations.

More information: https://manned.org/lvmpolld.

- Start the daemon in the foreground:

```
lvmpolld {{[-f|--foreground]}}
```

- Start in the foreground with debug logging:

```
lvmpolld {{[-f|--foreground]}} {{[-l|--log]}} debug
```

- Set the idle shutdown timeout (seconds):

```
lvmpolld {{[-t|--timeout]}} {{300}}
```

- Use a custom socket path:

```
lvmpolld {{[-s|--socket]}} {{/tmp/lvmpolld.socket}}
```

- Use a custom PID file:

```
lvmpolld {{[-p|--pidfile]}} {{/tmp/lvmpolld.pid}}
```

- Dump the current state:

```
lvmpolld --dump
```

lvmsadc

LVM system activity data collector (not supported under LVM2; prefer **`dmstats`**).

More information: https://manned.org/lvmsadc.

- Run the collector (legacy LVM1 systems only):

```
lvmsadc
```

- Report I/O statistics using the device-mapper replacement:

```
dmstats report {{/dev/mapper/device}}
```

lvmsar

LVM system activity reporter.

Not supported under LVM2; prefer **`dmstats`**.

More information: https://manned.org/lvmsar.

- Run the legacy reporter (LVM1 systems only):

```
lvmsar
```

- Report I/O statistics for a device using device-mapper stats:

```
dmstats report {{/dev/mapper/device}}
```

- List statistics regions for a device:

```
dmstats list {{/dev/mapper/device}}
```

lvreduce

Reduce the size of a logical volume.

See also: **`lvm`**.

More information: https://manned.org/lvreduce.

- Reduce a volume's size to 120 GB:

```
lvreduce {{[-L|--size]}} {{120G}} {{logical_volume}}
```

- Reduce a volume's size by 40 GB as well as the underlying filesystem:

```
lvreduce {{[-L|--size]}} -{{40G}} {{[-r|--resizefs]}}
{{logical_volume}}
```

lvremove

Remove logical volumes.

See also: **`lvm`**.

More information: https://manned.org/lvremove.

- Remove a logical volume in a volume group:

```
sudo lvremove {{volume_group}}/{{logical_volume}}
```

- Remove all logical volumes in a volume group:

```
sudo lvremove {{volume_group}}
```

lvrename

Rename a logical volume.

More information: https://manned.org/lvrename.

- Rename an LV using full paths:

```
sudo lvrename {{/dev/vg_name/old_lv}} {{/dev/vg_name/new_lv}}
```

- Rename an LV using the volume group and names:

```
sudo lvrename {{vg_name}} {{old_lv}} {{new_lv}}
```

- Answer "yes" to any prompts:

```
sudo lvrename {{[-y|--yes]}} {{/dev/vg_name/old_lv}} {{/dev/vg_name/new_lv}}
```

lvresize

Change the size of a logical volume.

See also: **`lvm`**.

More information: https://manned.org/lvresize.

- Change the size of a logical volume to 120 GB:

```
lvresize {{[-L|--size]}} 120G {{volume_group}}/
{{logical_volume}}
```

- Extend the size of a logical volume as well as the underlying filesystem by 120 GB:

```
lvresize {{[-L|--size]}} +120G {{[-r|--resizefs]}}
{{volume_group}}/{{logical_volume}}
```

- Extend the size of a logical volume to 100% of the free physical volume space:

```
lvresize {{[-l|--extents]}} 100%FREE {{volume_group}}/
{{logical_volume}}
```

- Reduce the size of a logical volume as well as the underlying filesystem by 120 GB:

```
lvresize {{[-L|--size]}} -120G {{[-r|--resizefs]}}
{{volume_group}}/{{logical_volume}}
```

lvs

Display information about logical volumes.

See also: **`lvm`**.

More information: https://manned.org/lvs.

- Display information about logical volumes:

```
sudo lvs
```

- Display all logical volumes:

```
sudo lvs {{[-a|--all]}}
```

- Change default display to show more details:

```
sudo lvs {{[-v|--verbose]}}
```

- Display only specific fields:

```
sudo lvs {{[-o|--options]}} {{field_name_1}},{{field_name_2}}
```

- Append field to default display:

```
sudo lvs {{[-o|--options]}} +{{field_name}}
```

- Suppress heading line:

```
sudo lvs --noheadings
```

- Use a separator to separate fields:

```
sudo lvs --separator {{=}}
```

lvscan

Scan (list) all logical volumes in the system.

Part of the LVM (Logical Volume Manager) suite.

More information: https://manned.org/lvscan.

- List all logical volumes:

```
lvscan
```

- List only active logical volumes:

```
lvscan --active
```

- List only inactive logical volumes:

```
lvscan --inactive
```

- Display logical volumes in JSON format:

```
lvscan --reportformat json
```

lxc-attach

Attach to a container.

More information: https://linuxcontainers.org/lxc/getting-started/.

- Attach to a container:

```
sudo lxc-attach {{container_name}}
```

- Display help:

```
lxc-attach {{[-?|--help]}}
```

lxc-checkconfig

Check a kernel for LXC support.

More information: https://linuxcontainers.org/lxc/getting-started/.

- Check the current kernel for LXC support:

```
lxc-checkconfig
```

- Check any kernel for LXC support:

```
CONFIG=/{{path/to/configuration}} lxc-checkconfig
```

lxc-console

Attach to a container.

More information: https://linuxcontainers.org/lxc/manpages//man1/lxc-console.1.html.

- Start a console in a container:

```
agetty {{[-L|--local-line]}} {{38400}} tty1
```

- Connect to an lxc console:

```
sudo lxc-console {{container_name}}
```

- Exit `lxc-console`:

```
<Ctrl a><q>
```

- Display help:

```
lxc-console {{[-?|--help]}}
```

lxc-create

Create linux containers.

More information: https://linuxcontainers.org/lxc/getting-started/.

- Create a container interactively in `/var/lib/lxc/`:

```
sudo lxc-create {{[-n|--name]}} {{container_name}} {{[-t|--template]}} download
```

- Create a container in a target directory:

```
sudo lxc-create {{[-P|--lxcpath]}} /{{path/to/directory}}/ {{[-n|--name]}} {{container_name}} {{[-t|--template]}} download
```

- Create a container passing options to a template:

```
sudo lxc-create {{[-n|--name]}} {{container_name}} {{[-t|--template]}} download -- {{[-d|--dist]}} {{distro-name}} {{[-r|--release]}} {{release-version}} {{[-a|--arch]}} {{arch}}
```

- Display help:

```
lxc-create {{[-?|--help]}}
```

lxc-destroy

Delete a container.

More information: https://linuxcontainers.org/lxc/getting-started/.

- Delete a container:

```
sudo lxc-destroy {{container_name}}
```

- Display help:

```
lxc-destroy {{[-?|--help]}}
```

lxc-info

Get info on a container.

More information: https://linuxcontainers.org/lxc/getting-started/.

- Query information from a container:

```
sudo lxc-info {{container_name}}
```

- Display help:

```
lxc-info {{[-?|--help]}}
```

lxc-ls

List Linux containers.

More information: https://linuxcontainers.org/lxc/manpages/man1/lxc-ls.1.html.

- List all containers:

```
sudo lxc-ls
```

- List active containers (including frozen and running):

```
sudo lxc-ls --active
```

- List only frozen containers:

```
sudo lxc-ls --frozen
```

- List only stopped containers:

```
sudo lxc-ls --stopped
```

- List containers in a fancy, column-based output:

```
sudo lxc-ls {{[-f|--fancy]}}
```

- Display help:

```
lxc-ls {{[-?|--help]}}
```

lxc network

Manage networks for LXD containers.

More information: https://documentation.ubuntu.com/lxd/latest/reference/manpages/lxc/network/.

- List all available networks:

```
lxc network list
```

- Show the configuration of a specific network:

```
lxc network show {{network_name}}
```

- Add a running instance to a specific network:

```
lxc network attach {{network_name}} {{container_name}}
```

- Create a new managed network:

```
lxc network create {{network_name}}
```

- Set a bridge interface of a specific network:

```
lxc network set {{network_name}} bridge.external_interfaces {{eth0}}
```

- Disable NAT for a specific network:

```
lxc network set {{network_name}} ipv{{4}}.nat false
```

lxc profile

Manage profiles for LXD containers.

More information: https://documentation.ubuntu.com/lxd/latest/reference/manpages/lxc/profile/.

- List all available profiles:

```
lxc profile list
```

- Show the configuration of a specific profile:

```
lxc profile show {{profile_name}}
```

- Edit a specific profile in the default editor:

```
lxc profile edit {{profile_name}}
```

- Edit a specific profile importing the configuration values from a file:

```
lxc < {{config.yaml}} profile edit {{profile_name}}
```

- Launch a new container with specific profiles:

```
lxc launch {{container_image}} {{container_name}} {{[-p|--profile]}} {{profile1}} {{[-p|--profile]}} {{profile2}}
```

- Change the profiles of a running container:

```
lxc profile assign {{container_name}} {{profile1,profile2}}
```

lxc-start

Start a container.

More information: https://linuxcontainers.org/lxc/getting-started/.

- Start the lxc service:

```
systemctl start lxc-net
```

- Start a container:

```
sudo lxc-start {{container_name}}
```

- Start a container in the foreground:

```
sudo lxc-start {{container_name}} {{[-F|--foreground]}}
```

- Exit out of a foreground container (run this in a separate terminal):

```
sudo lxc-stop {{container_name}}
```

- Display help:

```
lxc-start {{[-?|--help]}}
```

lxc-stop

Stop a container.

More information: https://linuxcontainers.org/lxc/getting-started/.

- Stop a container:

```
sudo lxc-stop {{container_name}}
```

- Display help:

```
lxc-stop {{[-?|--help]}}
```

lxc-top

Display resource usage of LXC containers.

More information: https://linuxcontainers.org/lxc/manpages/man1/lxc-top.1.html.

- Start `lxc-top`:

```
lxc-top
```

- Adjust update interval:

```
lxc-top {{[-d|--delay]}} {{5}}
```

- Sort by [n]ame, [c]pu use, [b]lock I/O, [m]emory, or [k]ernel memory:

```
lxc-top {{[-s|--sort]}} {{n|c|b|m|k}}
```

lxc

Manage Linux containers using the lxd REST API.

Any container names or patterns can be prefixed with the name of a remote server.

More information: https://manned.org/lxc.

- List local containers matching a string. Omit the string to list all local containers:

```
lxc list {{match_string}}
```

- List images matching a string. Omit the string to list all images:

```
lxc image list [{{remote}}:]{{match_string}}
```

- Create a new container from an image:

```
lxc init [{{remote}}:]{{image}} {{container}}
```

- Start a container:

```
lxc start [{{remote}}:]{{container}}
```

- Stop a container:

```
lxc stop [{{remote}}:]{{container}}
```

- Show detailed info about a container:

```
lxc info [{{remote}}:]{{container}}
```

- Take a snapshot of a container:

```
lxc snapshot [{{remote}}:]{{container}} {{snapshot}}
```

- Execute a specific command inside a container:

```
lxc exec [{{remote}}:]{{container}} {{command}}
```

lxi

Control LXI compatible instruments such as oscilloscopes.

More information: https://github.com/lxi-tools/lxi-tools#32-lxi.

- Discover LXI devices on available networks:

```
lxi discover
```

- Capture a screenshot, detecting a plugin automatically:

```
lxi screenshot {{[-a|--address]}} {{ip_address}}
```

- Capture a screenshot using a specified plugin:

```
lxi screenshot {{[-a|--address]}} {{ip_address}} {{[-p|--plugin]}} {{rigol-1000z}}
```

- Send an SCPI command to an instrument:

```
lxi scpi {{[-a|--address]}} {{ip_address}} "{{*IDN?}}"
```

- Run a benchmark for request and response performance:

```
lxi benchmark {{[-a|--address]}} {{ip_address}}
```

lxterminal

Terminal emulator for LXDE.

More information: https://github.com/lxde/lxterminal.

- Open an LXTerminal window:

```
lxterminal
```

- Open an LXTerminal window, run a command, and then exit:

```
lxterminal {{[-e|--command]}} "{{command}}"
```

- Open an LXTerminal window with multiple tabs:

```
lxterminal --tabs={{tab_name1,tab_name2,...}}
```

- Open an LXTerminal window with a specific title:

```
lxterminal --title={{title_name}}
```

- Open an LXTerminal window with a specific working directory:

```
lxterminal --working-directory={{path/to/directory}}
```

lynis

System and security auditing tool.

More information: https://cisofy.com/documentation/lynis/.

- Check that Lynis is up-to-date:

```
sudo lynis update info
```

- Run a security audit of the system:

```
sudo lynis audit system
```

- Run a security audit of a Dockerfile:

```
sudo lynis audit dockerfile {{path/to/dockerfile}}
```

lz

List all files inside a '.tar.gz' compressed archive.

More information: https://manned.org/lz.1.

- List all files inside a compressed archive:

```
lz {{path/to/file.tar.gz}}
```

mac2unix

Change macOS-style line endings to Unix-style.

Replaces CR with LF.

See also: **`unix2dos`**, **`unix2mac`**, **`dos2unix`**.

More information: https://manned.org/mac2unix.

- Change the line endings of a file:

```
mac2unix {{path/to/file}}
```

- Create a copy with Unix-style line endings:

```
mac2unix {{[-n|--newfile]}} {{path/to/file}} {{path/to/new_file}}
```

- Display file information:

```
mac2unix {{[-i|--info]}} {{path/to/file}}
```

- Keep/add/remove Byte Order Mark:

```
mac2unix --{{keep-bom|add-bom|remove-bom}} {{path/to/file}}
```

macchanger

Manipulate network interface MAC addresses.

More information: https://manned.org/macchanger.

- View the current and permanent MAC addresses of a interface:

```
macchanger {{[-s|--show]}} {{interface}}
```

- Set interface to a random MAC:

```
macchanger {{[-r|--random]}} {{interface}}
```

- Set an interface to a random MAC address, and pretend to be a [b]urned-[i]n-[a]ddress:

```
macchanger {{[-r|--random]}} {{[-b|--bia]}} {{interface}}
```

- Set an interface to a specific MAC address:

```
macchanger {{[-m|--mac]}} {{XX:XX:XX:XX:XX:XX}} {{interface}}
```

- Print the identifications (the first three bytes of a MAC address) of all known vendors:

```
macchanger {{[-l|--list]}}
```

- Reset an interface to its permanent hardware MAC address:

```
macchanger {{[-p|--permanent]}} {{interface}}
```

machinectl

Control the systemd machine manager.

Execute operations on virtual machines, containers and images.

More information: https://www.freedesktop.org/software/systemd/man/machinectl.html.

- Start a machine as a service using `systemd-nspawn`:

```
sudo machinectl start {{machine_name}}
```

- Stop a running machine:

```
sudo machinectl stop {{machine_name}}
```

- Display a list of running machines:

```
machinectl list
```

- Open an interactive shell inside the machine:

```
sudo machinectl shell {{machine_name}}
```

maim

Screenshot utility.

More information: https://manned.org/maim.

- Capture a screenshot and save it to the given path:

```
maim {{path/to/screenshot.png}}
```

- Capture a screenshot of the selected region:

```
maim {{[-s|--select]}} {{path/to/screenshot.png}}
```

- Capture a screenshot of the selected region and save it in the clipboard (requires `xclip`):

```
maim {{[-s|--select]}} | xclip {{[-se|-selection]}} {{[c|clipboard]}} {{[-t|-target]}} image/png
```

- Capture a screenshot of the current active window (requires `xdotool`):

```
maim {{[-i|--window]}} $(xdotool getactivewindow) {{path/to/screenshot.png}}
```

makepkg

Create a package which can be used with **`pacman`**.

Uses the **`PKGBUILD`** file in the current working directory by default.

More information: https://manned.org/makepkg.8.

- Make a package:

  `makepkg`

- Make a package and install its dependencies:

  `makepkg {{[-s|--syncdeps]}}`

- Make a package, install its dependencies then install it to the system:

  `makepkg {{[-si|--syncdeps --install]}}`

- Make a package, but skip checking the source's hashes:

  `makepkg --skipchecksums`

- Clean up work directories after a successful build:

  `makepkg {{[-c|--clean]}}`

- Verify the hashes of the sources:

  `makepkg --verifysource`

- Generate and save the source information into `.SRCINFO`:

  `makepkg --printsrcinfo > .SRCINFO`

- Download the source and install only the build dependencies for a program:

  `makepkg {{[-so|--syncdeps --nobuild]}}`

mako

Notification daemon for Wayland compositors.

Can be controlled with **`makoctl`**.

More information: https://manned.org/mako.

- Start the `mako` notification daemon:

  ```
  mako
  ```

- Start with a custom configuration file:

  ```
  mako {{[-c|--config]}} {{path/to/config}}
  ```

- Set maximum number of visible notifications:

  ```
  mako --max-visible {{5}}
  ```

- Set default timeout in milliseconds (0 to disable):

  ```
  mako --default-timeout {{2000}}
  ```

- Group notifications by application name:

  ```
  mako --group-by {{app_name}}
  ```

- Display help:

  ```
  mako {{[-h|--help]}}
  ```

makoctl dismiss

Dismiss notifications in **mako**.

More information: https://manned.org/makoctl.

- Dismiss the most recent notification:

```
makoctl dismiss
```

- Dismiss a specific notification by ID:

```
makoctl dismiss -n {{notification_id}}
```

- Dismiss all notifications:

```
makoctl dismiss {{[-a|--all]}}
```

- Dismiss all notifications in the first group:

```
makoctl dismiss {{[-g|--group]}}
```

- Dismiss without adding to history:

```
makoctl dismiss {{[-h|--no-history]}}
```

makoctl invoke

Invoke actions on notifications in **mako**.

More information: https://manned.org/makoctl.

- Invoke the default action on the most recent notification:

```
makoctl invoke
```

- Invoke a specific action on a notification (if not specified, default is used):

```
makoctl invoke -n {{notification_id}} {{action_name}}
```

makoctl mode

Manage notification modes in **mako**.

Modes can be used to change notification behavior (e.g., do-not-disturb).

More information: https://manned.org/makoctl.

- List all currently active modes:

```
makoctl mode
```

- Add a mode:

```
makoctl mode -a {{do-not-disturb}}
```

- Remove a mode:

```
makoctl mode -r {{do-not-disturb}}
```

- Toggle a mode (add if absent, remove if present):

```
makoctl mode -t {{do-not-disturb}}
```

- Set specific modes, replacing all current modes:

```
makoctl mode -s {{mode1 mode2 ...}}
```

makoctl

Control the **mako** notification daemon.

Some subcommands such as **dismiss**, **invoke**, and **mode** have their own usage documentation.

More information: https://manned.org/makoctl.

- List all current notifications:

```
makoctl list
```

- List notification history:

```
makoctl history
```

- Reload the configuration file:

```
makoctl reload
```

- Restore the most recently expired notification from history:

```
makoctl restore
```

- Display help:

```
makoctl help
```

mandb

Manage the pre-formatted manual page database.

More information: https://manned.org/mandb.

- Purge and process manual pages:

  `mandb`

- Update a single entry:

  `mandb --filename {{path/to/file}}`

- Create entries from scratch instead of updating:

  `mandb --create`

- Only process user databases:

  `mandb --user-db`

- Do not purge obsolete entries:

  `mandb --no-purge`

- Check the validity of manual pages:

  `mandb --test`

mangohud

Display a monitoring HUD on top of a Vulkan or OpenGL graphical application.

More information: https://github.com/flightlessmango/MangoHud.

- Use `mangohud` on top of an application:

  `mangohud {{command}}`

- Change overlay position:

  `<RightShift F11>`

- Change overlay verbosity:

  `<RightShift F10>`

- Toggle overlay visibility:

  `<RightShift F12>`

manpath

Determine the search path for manual pages.

More information: https://manned.org/manpath.

- Display the search path used to find man pages:

```
manpath
```

- Show the entire global manpath:

```
manpath {{[-g|--global]}}
```

mashtree

Make a fast tree from genomes.

Does not make a phylogeny.

More information: https://github.com/lskatz/mashtree.

- Fastest method in mashtree to create a tree from fastq and/or fasta files using multiple threads, piping into a newick file:

```
mashtree --numcpus {{12}} {{*.fastq.gz}} {{*.fasta}} >
{{mashtree.dnd}}
```

- Most accurate method in mashtree to create a tree from fastq and/or fasta files using multiple threads, piping into a newick file:

```
mashtree --mindepth {{0}} --numcpus {{12}} {{*.fastq.gz}}
{{*.fasta}} > {{mashtree.dnd}}
```

- Most accurate method to create a tree with confidence values (note that any options for `mashtree` itself has to be on the right side of the `--`):

```
mashtree_bootstrap.pl --reps {{100}} --numcpus {{12}}
{{*.fastq.gz}} -- --min-depth {{0}} >
{{mashtree.bootstrap.dnd}}
```

matchpathcon

Lookup the persistent SELinux security context setting of a path.

See also: **`semanage-fcontext`**, **`secon`**, **`chcon`**, **`restorecon`**.

More information: https://manned.org/matchpathcon.8.

- Lookup the persistent security context setting of an absolute path:

```
matchpathcon /{{path/to/file}}
```

- Restrict lookup to settings on a specific file type:

```
matchpathcon -m {{file|dir|pipe|chr_file|blk_file|lnk_file|
sock_file}} /{{path/to/file}}
```

- [V]erify that the persistent and current security context of a path agree:

```
matchpathcon -V /{{path/to/file}}
```

mate-about

Show information about MATE desktop environment.

More information: https://manned.org/mate-about.

- Display MATE version:

```
mate-about --version
```

mate-calc-cmd

Calculate mathematic expressions in MATE desktop environment in terminal.

More information: https://manned.org/mate-calc-cmd.

- Start an interactive calculator session:

```
mate-calc-cmd
```

- Calculate a specific mathematic expression:

```
{{2 + 5}}
```

mate-calc

Calculate mathematic expressions in MATE desktop environment.

More information: https://manned.org/mate-calc.

- Start the calculator:

```
mate-calc
```

- Calculate a specific mathematic expression:

```
mate-calc --solve {{2 + 5}}
```

mate-screenshot

Make screenshots in MATE desktop environment.

More information: https://manned.org/mate-screenshot.

- Create a fullscreen screenshot:

```
mate-screenshot
```

- Create an active window screenshot:

```
mate-screenshot --window
```

- Create a specific area screenshot:

```
mate-screenshot --area
```

- Create a screenshot interactively:

```
mate-screenshot --interactive
```

- Create a screenshot without borders:

```
mate-screenshot --window --remove-border
```

- Create a screenshot with a specific effect:

```
mate-screenshot --effect={{shadow|border|none}}
```

- Create a screenshot with a specific delay in seconds:

```
mate-screenshot --delay={{5}}
```

mate-search-tool

Search files in MATE desktop environment.

More information: https://manned.org/mate-search-tool.

- Search files containing a specific string in their name in a specific directory:

```
mate-search-tool --named={{string}} --path={{path/to/directory}}
```

- Search files without waiting a user confirmation:

```
mate-search-tool --start --named={{string}} --path={{path/to/directory}}
```

- Search files with name matching a specific `regex`:

```
mate-search-tool --start --regex={{string}} --path={{path/to/directory}}
```

- Set a sorting order in search results:

```
mate-search-tool --start --named={{string}} --path={{path/to/directory}} --sortby={{name|folder|size|type|date}}
```

- Set a descending sorting order:

```
mate-search-tool --start --named={{string}} --path={{path/to/directory}} --descending
```

- Search files owned by a specific user/group:

```
mate-search-tool --start --{{user|group}}={{value}} --path={{path/to/directory}}
```

mbw

Memory Bandwidth Benchmark.

More information: https://github.com/raas/mbw.

- Run 3 memory bandwidth tests with 512MB size:

```
mbw -n 3 512
```

- Run 3 memory bandwidth tests with 512MB memory size, output only statistics, not averages:

```
mbw -n 3 -q -a 512
```

- Run memcpy test 3 times with 512MB size, only display statistics:

```
mbw -n 3 -q -t{{0}} 512
```

- Run the memcpy test 10 times with 1024 byte blocks allocated 8192MB of memory:

```
mbw -n 10 -q -t{{2}} -b 1024 8192
```

- Run dumb test with 2048MB size, output only statistics, run forever:

```
mbw -n 0 -t{{1}} -q 2048
```

mcookie

Generate random 128-bit hexadecimal numbers.

More information: https://manned.org/mcookie.

- Generate a random number:

```
mcookie
```

- Generate a random number, using the contents of a file as a seed for the randomness:

```
mcookie {{[-f|--file]}} {{path/to/file}}
```

- Generate a random number, using a specific number of bytes from a file as a seed for the randomness:

```
mcookie {{[-f|--file]}} {{path/to/file}} {{[-m|--max-size]}} {{number_of_bytes}}
```

- Print the details of the randomness used, such as the origin and seed for each source:

```
mcookie {{[-v|--verbose]}}
```

mdadm

RAID management utility.

More information: https://manned.org/mdadm.

- Create array:

```
sudo mdadm --create {{/dev/md/MyRAID}} --level {{raid_level}}
--raid-devices {{number_of_disks}} {{/dev/sdXN}}
```

- Stop array:

```
sudo mdadm --stop {{/dev/md0}}
```

- Mark disk as failed:

```
sudo mdadm --fail {{/dev/md0}} {{/dev/sdXN}}
```

- Remove disk:

```
sudo mdadm --remove {{/dev/md0}} {{/dev/sdXN}}
```

- Add disk to array:

```
sudo mdadm --assemble {{/dev/md0}} {{/dev/sdXN}}
```

- Show RAID info:

```
sudo mdadm --detail {{/dev/md0}}
```

- Reset disk by deleting RAID metadata:

```
sudo mdadm --zero-superblock {{/dev/sdXN}}
```

mdbook

Create online books by writing Markdown files.

More information: https://rust-lang.github.io/mdBook/cli/index.html.

- Create an mdbook project in the current directory:

```
mdbook init
```

- Create an mdbook project in a specific directory:

```
mdbook init {{path/to/directory}}
```

- Clean the directory with the generated book:

```
mdbook clean
```

- Serve a book at http://localhost:3000, auto build when file changes:

```
mdbook serve
```

- Watch a set of Markdown files and automatically build when a file is changed:

```
mdbook watch
```

mediamtx

Real-time media server and proxy.

More information: https://github.com/bluenviron/mediamtx.

- Run MediaMTX:

```
mediamtx
```

- Run MediaMTX with a custom configuration location:

```
mediamtx {{path/to/config.yml}}
```

- Start MediaMTX as a daemon:

```
systemctl start mediamtx
```

megadl

This command is an alias of **`megatools-dl`**.

- View documentation for the original command:

```
tldr megatools-dl
```

megatools-dl

Download files from **`mega.nz`**.

Part of the **`megatools`** suite.

More information: https://megatools.megous.com/man/megatools-dl.html.

- Download files from a `mega.nz` link into the current directory:

```
megatools-dl {{https://mega.nz/...}}
```

- Download files from a `mega.nz` link into a specific directory:

```
megatools-dl --path {{path/to/directory}} {{https://
mega.nz/...}}
```

- Interactively choose which files to download:

```
megatools-dl --choose-files {{https://mega.nz/...}}
```

- Limit the download speed in KiB/s:

```
megatools-dl --limit-speed {{speed}} {{https://mega.nz/...}}
```

mesg

Check or set a terminal's ability to receive messages from other users, usually from the **`write`** command.

See also: **`write`**, **`talk`**.

More information: https://manned.org/mesg.

- Check terminal's openness to write messages:

```
mesg
```

- Disallow receiving messages from other users:

```
mesg n
```

- Allow receiving messages from other users:

```
mesg y
```

- Enable verbose mode, printing a warning if the command is not executed from a terminal:

```
mesg {{[-v|--verbose]}}
```

mhwd-gpu

Configure graphics cards in Manjaro.

More information: https://wiki.manjaro.org/index.php/Configure_NVIDIA_(non-free)_settings_and_load_them_on_Startup/en#Configure_The_Resolution.2FRefresh_Rate.

- Show current Xorg configuration path:

```
mhwd-gpu --status
```

- Check if Xorg configuration has a valid symlink:

```
mhwd-gpu --check
```

- Set a custom Xorg configuration for an Nvidia GPU:

```
sudo mhwd-gpu --setmod nvidia --setxorg /{{path/to/nvidia.conf}}
```

- Set a custom Xorg configuration for an AMD GPU:

```
sudo mhwd-gpu --setmod {{catalyst|ati}} --setxorg /{{path/to/amdgpu.conf}}
```

- Display help:

```
mhwd-gpu --help
```

mhwd-kernel

Manage and install Linux kernels in Manjaro.

More information: https://wiki.manjaro.org/index.php/Manjaro_Kernels.

- List all available kernels:

```
mhwd-kernel {{[-l|--list]}}
```

- List all installed kernels:

```
mhwd-kernel {{[-li|--listinstalled]}}
```

- Install a kernel:

```
sudo mhwd-kernel {{[-i|--install]}} {{kernel}}
```

- Remove a kernel:

```
sudo mhwd-kernel {{[-r|--remove]}} {{kernel}}
```

- Install a kernel, replacing the currently running kernel:

```
sudo mhwd-kernel {{[-i|--install]}} {{kernel}} rmc
```

mhwd

Manjaro Hardware Detection utility.

More information: https://wiki.manjaro.org/index.php/Manjaro_Hardware_Detection_Overview.

- List available drivers:

```
mhwd {{[-l|--list]}}
```

- List installed drivers:

```
mhwd {{[-li|--listinstalled]}}
```

- Install a driver:

```
mhwd {{[-i|--install]}} {{pci|usb}} {{driver_name}}
```

- Remove a driver:

```
mhwd {{[-r|--remove]}} {{pci|usb}} {{driver_name}}
```

- Display detailed information about detected hardware:

```
mhwd {{[-l|--list]}} {{[-d|--detail]}}
```

- Automatically install best available driver for the detected graphics card:

```
mhwd {{[-a|--auto]}} pci {{free|nonfree}} 0300
```

microcom

A minimalistic terminal program, used to access remote devices via a serial, CAN or telnet connection from the console.

More information: https://manned.org/microcom.

- Open a serial port using the specified baud rate:

```
microcom {{[-p|--port]}} {{/dev/ttyXYZ}} {{[-s|--speed]}} {{baud_rate}}
```

- Establish a telnet connection to the specified host:

```
microcom {{[-t|--telnet]}} {{hostname}}:{{port}}
```

mimetype

Automatically determine the MIME type of a file.

More information: https://manned.org/mimetype.

- Print the MIME type of a given file:

```
mimetype {{path/to/file}}
```

- Display only the MIME type, and not the filename:

```
mimetype --brief {{path/to/file}}
```

- Display a description of the MIME type:

```
mimetype --describe {{path/to/file}}
```

- Determine the MIME type of `stdin` (does not check a filename):

```
{{command}} | mimetype --stdin
```

- Display debug information about how the MIME type was determined:

```
mimetype --debug {{path/to/file}}
```

- Display all the possible MIME types of a given file in confidence order:

```
mimetype --all {{path/to/file}}
```

- Explicitly specify the 2-letter language code of the output:

```
mimetype --language {{path/to/file}}
```

minicom

Communicate with the serial interface of a device.

More information: https://manned.org/minicom.

- Open a given serial port:

```
sudo minicom {{[-D|--device]}} {{/dev/ttyXYZ}}
```

- Open a given serial port with a given baud rate:

```
sudo minicom {{[-D|--device]}} {{/dev/ttyXYZ}} {{[-b|--baudrate]}} {{115200}}
```

- Enter the configuration menu before communicating with a given serial port:

```
sudo minicom {{[-D|--device]}} {{/dev/ttyXYZ}} {{[-s|--setup]}}
```

- Capture the output of a serial port to a file:

```
sudo minicom {{[-D|--device]}} {{/dev/ttyXYZ}} {{[-C|--capturefile]}} {{path/to/file}}
```

- Exit minicom:

```
<Ctrl a><x><Enter>
```

- Display help:

```
minicom {{[-h|--help]}}
```

mke2fs

Create a Linux filesystem inside a partition.

More information: https://manned.org/mke2fs.

- Create an ext2 filesystem in partition 1 of device b (`sdb1`):

```
mke2fs -t ext2 {{/dev/sdb1}}
```

- Create an ext3 filesystem in partition 1 of device b (`sdb1`):

```
mke2fs -t ext3 {{/dev/sdb1}}
```

- Create an ext4 filesystem in partition 1 of device b (`sdb1`):

```
mke2fs -t ext4 {{/dev/sdb1}}
```

mkfs.bcachefs

Create a **bcachefs** filesystem inside a partition.

See also: **bcachefs**.

More information: https://bcachefs-docs.readthedocs.io/en/latest/mgmt-formatting.html.

- Create a bcachefs filesystem inside partition Y on a device X:

```
sudo mkfs.bcachefs {{/dev/sdXY}}
```

- Create a bcachefs filesystem with a volume label:

```
sudo mkfs.bcachefs {{[-L|--fs_label]}} {{volume_label}} {{/dev/sdXY}}
```

mkfs.btrfs

Create a BTRFS filesystem.

Defaults to **`raid1`**, which specifies 2 copies of a data block spread across 2 different devices.

More information: https://btrfs.readthedocs.io/en/latest/mkfs.btrfs.html.

- Create a Btrfs filesystem on an empty partition:

```
sudo mkfs.btrfs {{/dev/sdXY}}
```

- Create a btrfs filesystem on a single device:

```
sudo mkfs.btrfs {{[-m|--metadata]}} single {{[-d|--data]}}
single {{/dev/sdX}}
```

- Create a btrfs filesystem on multiple devices with raid1:

```
sudo mkfs.btrfs {{[-m|--metadata]}} raid1 {{[-d|--data]}}
raid1 {{/dev/sdX /dev/sdY /dev/sdZ ...}}
```

- Set a label for the filesystem:

```
sudo mkfs.btrfs {{[-L|--label]}} "{{label}}" {{/dev/sdX /dev/
sdY ...}}
```

- Overwrite existing filesystem if one is detected:

```
sudo mkfs.btrfs {{[-f|--force]}} {{/dev/sdX}}
```

mkfs.cramfs

Create a ROM filesystem inside a partition.

More information: https://manned.org/mkfs.cramfs.

- Create a ROM filesystem out of a directory inside partition Y on device X:

```
sudo mkfs.cramfs {{path/to/directory}} {{/dev/sdXY}}
```

- Create a ROM filesystem with a volume-name:

```
sudo mkfs.cramfs -n {{volume_name}} {{path/to/directory}} {{/dev/sdXY}}
```

mkfs.erofs

Create an EROFS filesystem in an image.

More information: https://manned.org/mkfs.erofs.

- Create an EROFS filesystem based on the root directory:

```
mkfs.erofs image.erofs root/
```

- Create an EROFS image with a specific UUID:

```
mkfs.erofs -U {{UUID}} image.erofs root/
```

- Create a compressed EROFS image:

```
mkfs.erofs -zlz4hc image.erofs root/
```

- Create an EROFS image where all files are owned by root:

```
mkfs.erofs --all-root image.erofs root/
```

mkfs.exfat

Create an exfat filesystem inside a partition.

More information: https://manned.org/mkfs.exfat.

- Create an exfat filesystem inside partition Y on device X:

```
sudo mkfs.exfat {{/dev/sdXY}}
```

- Create filesystem with a volume-name:

```
sudo mkfs.exfat {{[-L|--volume-label]}} {{volume_name}} {{/dev/sdXY}}
```

- Create filesystem with a volume-id:

```
sudo mkfs.exfat {{[-U|--volume-guid]}} {{volume_id}} {{/dev/sdXY}}
```

mkfs.ext4

Create an ext4 filesystem inside a partition.

More information: https://manned.org/mkfs.ext4.

- Create an ext4 filesystem inside partition Y on device X:

```
sudo mkfs.ext4 {{/dev/sdXY}}
```

- Create an ext4 filesystem with a volume-label:

```
sudo mkfs.ext4 -L {{volume_label}} {{/dev/sdXY}}
```

mkfs.f2fs

Create an F2FS filesystem inside a partition.

More information: https://manned.org/mkfs.f2fs.

- Create an F2FS filesystem inside partition Y on device X:

```
sudo mkfs.f2fs {{/dev/sdXY}}
```

- Create an F2FS filesystem with a volume label:

```
sudo mkfs.f2fs -l {{volume_label}} {{/dev/sdXY}}
```

mkfs.fat

Create an MS-DOS filesystem inside a partition.

More information: https://manned.org/mkfs.fat.

- Create a fat filesystem inside partition Y on device X:

```
sudo mkfs.fat {{/dev/sdXY}}
```

- Create filesystem with a volume-name:

```
sudo mkfs.fat -n {{volume_name}} {{/dev/sdXY}}
```

- Create filesystem with a volume-id:

```
sudo mkfs.fat -i {{volume_id}} {{/dev/sdXY}}
```

- Use 4 instead of 2 file allocation tables:

```
sudo mkfs.fat -f 4 {{/dev/sdXY}}
```

- Specify filesystem type:

```
sudo mkfs.fat -F {{12|16|32}} {{/dev/sdXY}}
```

mkfs

Build a Linux filesystem on a hard disk partition.

This command is deprecated in favor of filesystem specific mkfs.type utils.

More information: https://manned.org/mkfs.

- Build a Linux ext2 filesystem on a partition:

```
sudo mkfs {{/dev/sdXY}}
```

- Build a filesystem of a specified type:

```
sudo mkfs {{[-t|--type]}} {{ext4}} {{/dev/sdXY}}
```

- Build a filesystem of a specified type and check for bad blocks:

```
sudo mkfs -c {{[-t|--type]}} {{ntfs}} {{/dev/sdXY}}
```

mkfs.minix

Create a Minix filesystem inside a partition.

More information: https://manned.org/mkfs.minix.

- Create a Minix filesystem inside partition Y on device X:

```
sudo mkfs.minix {{/dev/sdXY}}
```

mkfs.ntfs

Create a NTFS filesystem inside a partition.

More information: https://manned.org/mkfs.ntfs.

- Create a NTFS filesystem inside partition Y on device X:

```
sudo mkfs.ntfs {{/dev/sdXY}}
```

- Create filesystem with a volume-label:

```
sudo mkfs.ntfs {{[-L|--label]}} {{volume_label}} {{/dev/sdXY}}
```

- Create filesystem with specific UUID:

```
sudo mkfs.ntfs {{[-U|--with-uuid]}} {{UUID}} {{/dev/sdXY}}
```

mkfs.vfat

This command is an alias of **`mkfs.fat`**.

- View documentation for the original command:

```
tldr mkfs.fat
```

mkfs.xfs

Create an XFS filesystem inside a partition.

More information: https://manned.org/mkfs.xfs.

- Create an XFS filesystem inside partition Y on device X:

```
sudo mkfs.xfs {{/dev/sdXY}}
```

- Create an XFS filesystem with a volume label:

```
sudo mkfs.xfs -L {{volume_label}} {{/dev/sdXY}}
```

mkhomedir_helper

Create the user's home directory after creating the user.

More information: https://manned.org/mkhomedir_helper.

- Create a home directory for a user based on `/etc/skel` with umask 022:

```
sudo mkhomedir_helper {{username}}
```

- Create a home directory for a user based on `/etc/skel` with all permissions for owner (0) and read permission for group (3):

```
sudo mkhomedir_helper {{username}} {{037}}
```

- Create a home directory for a user based on a custom skeleton:

```
sudo mkhomedir_helper {{username}} {{umask}} {{path/to/skeleton_directory}}
```

mkinitcpio

Generate initial ramdisk environments for booting the Linux kernel based on the specified preset(s).

More information: https://manned.org/mkinitcpio.8.

- Perform a dry run (print what would be done without actually doing it):

```
mkinitcpio
```

- Generate ramdisk environments based on all existing presets (used to regenerate all the initramfs images after a change in `/etc/mkinitcpio.conf`):

```
sudo mkinitcpio {{[-P|--allpresets]}}
```

- Generate a ramdisk environment based on the `linux` preset:

```
sudo mkinitcpio {{[-p|--preset]}} linux
```

- Generate a ramdisk environment based on the `linux-lts` preset:

```
sudo mkinitcpio {{[-p|--preset]}} linux-lts
```

- Generate an initramfs image using an alternative configuration file:

```
sudo mkinitcpio {{[-c|--config]}} {{path/to/mkinitcpio.conf}} {{[-g|--generate]}} {{path/to/initramfs.img}}
```

- Generate an initramfs image for a kernel other than the one currently running (the installed kernel releases can be found in `/usr/lib/modules/`):

```
sudo mkinitcpio {{[-k|--kernel]}} {{kernel_version}} {{[-g|--generate]}} {{path/to/initramfs.img}}
```

- List all available hooks:

```
mkinitcpio {{[-L|--listhooks]}}
```

- Display help for a specific hook:

```
mkinitcpio {{[-H|--hookhelp]}} {{hook_name}}
```

mkinitfs

Generate an initramfs on Alpine Linux.

More information: https://manned.org/mkinitfs.

- Generate an initramfs with the features specified in `/etc/mkinitfs/mkinitfs.conf`:

```
mkinitfs
```

- Use a different configuration file:

```
mkinitfs -c {{path/to/config}}
```

- Compress the initramfs using the specified compression algorithm (default: gzip):

```
mkinitfs -C {{gzip|xz|zstd|lz4|none}}
```

- List files that will be included in the initramfs image:

```
mkinitfs -l
```

- List all available features:

```
mkinitfs -L
```

mkisofs

Create ISO files from directories.

Also aliased as **genisoimage**.

More information: https://manned.org/mkisofs.

- Create an ISO from a directory:

```
mkisofs -o {{filename.iso}} {{path/to/source_directory}}
```

- Set the disc label when creating an ISO:

```
mkisofs -o {{filename.iso}} -V "{{label_name}}" {{path/to/
source_directory}}
```

mklost+found

Create a lost+found directory.

More information: https://linux.die.net/man/8/mklost+found.

- Create a `lost+found` directory in the current directory:

```
mklost+found
```

mknod

Create block or character device special files.

More information: https://www.gnu.org/software/coreutils/manual/html_node/mknod-invocation.html.

- Create a block device:

```
sudo mknod {{path/to/device_file}} b {{major_device_number}} {{minor_device_number}}
```

- Create a character device:

```
sudo mknod {{path/to/device_file}} c {{major_device_number}} {{minor_device_number}}
```

- Create a FIFO (queue) device:

```
sudo mknod {{path/to/device_file}} p
```

- Create a device file with default SELinux security context:

```
sudo mknod {{[-Z |--context=]}}{{path/to/device_file}} {{type}} {{major_device_number}} {{minor_device_number}}
```

mkosi

Build modern, legacy-free Linux images.

Part of **systemd**.

More information: https://manned.org/mkosi.

- Show current build configuration to verify what would be built:

```
mkosi summary
```

- Build an image with default settings (if no distribution is selected, the distribution of the host system is used):

```
mkosi build --distribution {{fedora|debian|ubuntu|arch|opensuse|...}}
```

- Build an image and run an interactive shell in a systemd-nspawn container of the image:

```
mkosi shell
```

- Boot an image in a virtual machine using QEMU (only supported for disk images or CPIO images when a kernel is provided):

```
mkosi qemu
```

- Display help:

```
mkosi help
```

mksquashfs

Create or append files and directories to squashfs filesystems.

More information: https://manned.org/mksquashfs.

- Create or append files and directories to a squashfs filesystem (compressed using `gzip` by default):

```
mksquashfs {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}} {{filesystem.squashfs}}
```

- Create or append files and directories to a squashfs filesystem, using a specific [comp]ression algorithm:

```
mksquashfs {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}} {{filesystem.squashfs}} -comp
{{gzip|lzo|lz4|xz|zstd|lzma}}
```

- Create or append files and directories to a squashfs filesystem, [e]xcluding some of them:

```
mksquashfs {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}} {{filesystem.squashfs}} -e {{file|
directory1 file|directory2 ...}}
```

- Create or append files and directories to a squashfs filesystem, [e]xcluding those ending with gzip:

```
mksquashfs {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}} {{filesystem.squashfs}} -wildcards -
e "{{*.gz}}"
```

- Create or append files and directories to a squashfs filesystem, [e]xcluding those matching a `regex`:

```
mksquashfs {{path/to/file_or_directory1 path/to/
file_or_directory2 ...}} {{filesystem.squashfs}} -regex -e
"{{regex}}"
```

mkswap

Set up a Linux swap area on a device or in a file.

Note: **`path/to/file`** can either point to a regular file or a swap partition.

More information: https://manned.org/mkswap.

- Set up a given swap area:

```
sudo mkswap {{path/to/file}}
```

- Check a partition for bad blocks before creating the swap area:

```
sudo mkswap {{[-c|--check]}} {{path/to/file}}
```

- Specify a label for the partition (to allow `swapon` to use the label):

```
sudo mkswap {{[-L|--label]}} {{label}} {{/dev/sdXY}}
```

- Use the specified UUID:

```
sudo mkswap {{[-U|--uuid]}} {{clear|random|time|UUID_value}}
```

- Set up a swap file (for btrfs, see `tldr btrfs filesystem` instead):

```
sudo mkswap {{[-s|--size]}} {{file_size}} {{[-F|--file]}}
{{path/to/swapfile}}
```

mktemp

Create a temporary file or directory.

More information: https://www.gnu.org/software/coreutils/manual/html_node/mktemp-invocation.html.

- Create an empty temporary file and print its absolute path:

```
mktemp
```

- Use a custom directory (defaults to `$TMPDIR`, or `/tmp`):

```
mktemp {{[-p |--tmpdir=]}}/{{path/to/temporary_directory}}
```

- Use a custom path template (`X`s are replaced with random alphanumeric characters):

```
mktemp {{/tmp/example.XXXXXXXX}}
```

- Use a custom file name template:

```
mktemp -t {{example.XXXXXXXX}}
```

- Create an empty temporary file with the given suffix and print its absolute path:

```
mktemp --suffix {{.ext}}
```

- Create an empty temporary directory and print its absolute path:

```
mktemp {{[-d|--directory]}}
```

- Print the name of a temporary file or directory without actually creating it:

```
mktemp {{[-u|--dry-run]}}
```

mkvpropedit

Modify properties of existing Matroska files without a complete remux.

More information: https://mkvtoolnix.download/doc/mkvpropedit.html.

- Delete title:

```
mkvpropedit {{[-d|--delete]}} title {{path/to/file.mkv}}
```

- Mark subtitle track 3 as SDH - Subtitles for deaf and hearing impared:

```
mkvpropedit {{path/to/file.mkv}} {{[-e|--edit]}} track:s3
{{[-s|--set]}} flag-hearing-impaired=1
```

- Mark audio track 2 as Default:

```
mkvpropedit {{path/to/file.mkv}} {{[-e|--edit]}} track:a2
{{[-s|--set]}} flag-default=1
```

- Delete the name of video track 1:

```
mkvpropedit {{path/to/file.mkv}} {{[-e|--edit]}} track:v1
{{[-d|--delete]}} name
```

mlabel

Set an MS-DOS volume label for FAT and VFAT filesystems.

More information: https://www.gnu.org/software/mtools/manual/mtools.html#mlabel.

- Set a filesystem label:

```
mlabel -i {{/dev/sda}} ::"{{new_label}}"
```

mmcli

Control and monitor the ModemManager.

More information: https://www.freedesktop.org/software/ModemManager/man/latest/mmcli.1.html.

- List available modems:

```
mmcli --list-modems
```

- Print information about a modem:

```
mmcli --modem={{modem}}
```

- Enable a modem:

```
mmcli --modem={{modem}} --enable
```

- List SMS messages available on the modem:

```
sudo mmcli --modem={{modem}} --messaging-list-sms
```

- Delete a message from the modem, specifying its path:

```
sudo mmcli --modem={{modem}} --messaging-delete-sms={{path/to/message_file}}
```

mmdebstrap

Create a Debian chroot.

Alternative to **debootstrap**.

More information: https://gitlab.mister-muffin.de/josch/mmdebstrap/.

- Create a Debian Stable directory chroot:

```
sudo mmdebstrap stable {{path/to/debian-root}}/
```

- Create a Debian Bookworm tarball chroot using a mirror:

```
mmdebstrap bookworm {{path/to/debian-bookworm.tar}} {{http://
mirror.example.org/debian}}
```

- Create a Debian Sid tarball chroot with additional packages:

```
mmdebstrap sid {{path/to/debian-sid.tar}} --
include={{pkg1,pkg2}}
```

mocp

Music on Console (MOC) audio player.

More information: https://manned.org/mocp.

- Launch the MOC terminal UI:

```
mocp
```

- Launch the MOC terminal UI in a specific directory:

```
mocp {{path/to/directory}}
```

- Start the MOC server in the background, without launching the MOC terminal UI:

```
mocp {{[-S|--server]}}
```

- Add a specific song to the play queue while MOC is in the background:

```
mocp {{[-q|--enqueue]}} {{path/to/audio_file}}
```

- Add songs recursively to the play queue while MOC is in the background:

```
mocp {{[-a|--append]}} {{path/to/directory}}
```

- Clear the play queue while MOC is in the background:

```
mocp {{[-c|--clear]}}
```

- Play or stop the currently queued song while MOC is in the background:

```
mocp --{{play|stop}}
```

- Stop the MOC server while it's in the background:

```
mocp {{[-x|--exit]}}
```

modetest

Diagnose Direct Rendering Manager and Kernel Mode Setting.

More information: https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18841850/Video_Mixer#modetest.

- List connectors and their available modes for a specific driver:

```
modetest -M {{mgag200}} -c
```

- Set the resolution of a connector:

```
sudo modetest -M {{mgag200}} -s {{connector_id}}:{{1600}}x{{1200}}
```

modinfo

Extract information about a Linux kernel module.

See also: **kmod** for other module management commands.

More information: https://manned.org/modinfo.

- List all attributes of a kernel module:

```
modinfo {{kernel_module}}
```

- List the specified attribute only:

```
modinfo {{[-F|--field]}} {{author|description|license|parm|
filename|version|...}} {{kernel_module}}
```

modprobe

Add or remove modules from the Linux kernel.

See also: **kmod** for other module management commands.

More information: https://manned.org/modprobe.

- Pretend to load a module into the kernel, but don't actually do it:

```
sudo modprobe {{[-n|--dry-run]}} {{module_name}}
```

- Load a module into the kernel:

```
sudo modprobe {{module_name}}
```

- Remove a module from the kernel:

```
sudo modprobe {{[-r|--remove]}} {{module_name}}
```

- Remove a module and those that depend on it from the kernel:

```
sudo modprobe {{[-r|--remove]}} --remove-holders
{{module_name}}
```

- Show a kernel module's dependencies:

```
sudo modprobe {{[-D|--show-depends]}} {{module_name}}
```

module

Modify a users' environment.

More information: https://lmod.readthedocs.io/en/latest/010_user.html.

- Display available modules:

```
module avail
```

- Search for a module by name:

```
module avail {{module_name}}
```

- Load a module:

```
module load {{module_name}}
```

- Display loaded modules:

```
module list
```

- Unload a specific loaded module:

```
module unload {{module_name}}
```

- Unload all loaded modules:

```
module purge
```

- Specify user-created modules:

```
module use {{path/to/module_file1 path/to/module_file2 ...}}
```

- Save the current set of loaded modules:

```
module save {{collection_name}}
```

mokutil

Configure Secure Boot Machine Owner Keys (MOK).

Some operations, such as enabling and disabling Secure Boot or enrolling keys require a reboot.

More information: https://github.com/lcp/mokutil.

- Show if Secure Boot is enabled:

```
mokutil --sb-state
```

- Enable Secure Boot:

```
mokutil --enable-validation
```

- Disable Secure Boot:

```
mokutil --disable-validation
```

- List enrolled keys:

```
mokutil --list-enrolled
```

- Enroll a new key:

```
mokutil --import {{path/to/key.der}}
```

- List the keys to be enrolled:

```
mokutil --list-new
```

- Set shim verbosity:

```
mokutil --set-verbosity true
```

mono

Runtime for the .NET Framework.

More information: https://www.mono-project.com/docs/.

- Run a .NET assembly in debug mode:

```
mono --debug {{path/to/program.exe}}
```

- Run a .NET assembly:

```
mono {{path/to/program.exe}}
```

mons

Quickly manage two displays.

More information: https://github.com/Ventto/mons.

- Enable only the primary monitor:

```
mons -o
```

- Enable only the secondary monitor:

```
mons -s
```

- Duplicate the primary monitor onto the secondary monitor, using the resolution of the primary monitor:

```
mons -d
```

- Mirror the primary monitor onto the secondary monitor, using the resolution of the secondary monitor:

```
mons -m
```

mopac

MOPAC (Molecular Orbital PACkage) is a semiempirical quantum chemistry program based on Dewar and Thiel's NDDO approximation.

More information: https://github.com/openmopac/mopac.

- Perform calculations according to an input file (`.mop`, `.dat`, and `.arc`):

```
mopac {{path/to/input_file}}
```

- Minimal working example with HF that writes to the current directory and streams the output file:

```
touch test.out; echo "PM7\n#comment\n\nH 0.95506 0.05781 -0.03133\nF 1.89426 0.05781 -0.03133" > test.mop; mopac test.mop & tail {{[-f|--follow]}} test.out
```

more

Interactively display a file, allowing scrolling and searching.

See also: **`less`**.

More information: https://manned.org/more.

- Open a file:

  `more {{path/to/file}}`

- Display a specific line:

  `more +{{line_number}} {{path/to/file}}`

- Go to the next page:

  `<Space>`

- Search for a string (press `<n>` to go to the next match):

  `</>{{something}}<Enter>`

- Exit:

  `<q>`

- Display help about interactive commands:

  `<h>`

mount.cifs

Mount SMB (Server Message Block) or CIFS (Common Internet File System) shares.

Note: You can also do the same thing by passing the **`-t cifs`** option to **`mount`**.

More information: https://manned.org/mount.cifs.

- Connect using the specified username or $USER by default (you will be prompted for a password):

```
mount.cifs -o user={{username}} //{{server}}/{{share_name}}
{{mountpoint}}
```

- Connect as the guest user (without a password):

```
mount.cifs -o guest //{{server}}/{{share_name}}
{{mountpoint}}
```

- Set ownership information for the mounted directory:

```
mount.cifs -o uid={{user_id|username}},gid={{group_id|
groupname}} //{{server}}/{{share_name}} {{mountpoint}}
```

mount.ddi

Mount Discoverable Disk Images.

See also: **`systemd-dissect`** for other commands relevant to DDIs.

More information: https://www.freedesktop.org/software/systemd/man/systemd-dissect.html.

- Mount an OS image:

```
mount.ddi {{path/to/image.raw}} {{/mnt/image}}
```

mount

Get access to an entire filesystem in one directory.

More information: https://manned.org/mount.8.

- Show all mounted filesystems:

```
mount
```

- Mount a device to a directory:

```
mount {{path/to/device_file}} {{path/to/target_directory}}
```

- Create a specific directory if it does not exist and mount a device to it:

```
mount {{[-m|--mkdir]}} {{path/to/device_file}} {{path/to/target_directory}}
```

- Mount a device to a directory for a specific user:

```
mount {{[-o|--options]}} uid={{user_id}},gid={{group_id}} {{path/to/device_file}} {{path/to/target_directory}}
```

- Mount a CD-ROM device (with the filetype ISO9660) to `/cdrom` (readonly):

```
mount {{[-t|--types]}} iso9660 {{[-o|--options]}} ro {{/dev/cdrom}} /cdrom
```

- Mount all the filesystems defined in `/etc/fstab`:

```
mount {{[-a|--all]}}
```

- Mount a specific filesystem described in `/etc/fstab` (e.g. `/dev/sda1 /my_drive ext2 defaults 0 2`):

```
mount {{/my_drive}}
```

- Mount a directory to another directory:

```
mount {{[-B|--bind]}} {{path/to/old_directory}} {{path/to/new_directory}}
```

mount.smb3

This command is an alias of **`mount.cifs`**.

Note: For SMB versions before 3 you have to use **`mount.cifs`** instead.

- View documentation for the original command:

```
tldr mount.cifs
```

mount.steamos

Mount or unmount SteamOS filesystem partitions.

More information: https://gitlab.com/users/evlaV/projects.

- Mount all necessary partitions from a device to a target directory:

```
sudo mount.steamos {{/dev/sdX}} {{/mnt}}
```

- Mount with options to exclude specific partitions (e.g. `/home`, overlays):

```
sudo mount.steamos {{[-o|--options]}} nohome,nooverlay {{/dev/sdX}} {{/mnt}}
```

- Unmount all partitions mounted under a target directory:

```
sudo mount.steamos -u {{/mnt}}
```

- Display help:

```
mount.steamos {{[-h|--help]}}
```

mountpoint

Test if a directory is a filesystem mountpoint.

More information: https://manned.org/mountpoint.

- Check if a directory is a mountpoint:

```
mountpoint {{path/to/directory}}
```

- Check if a directory is a mountpoint without showing any output:

```
mountpoint {{[-q|--quiet]}} {{path/to/directory}}
```

- Show major/minor numbers of a mountpoint's filesystem:

```
mountpoint {{[-d|--fs-devno]}} {{path/to/directory}}
```

mpg123

Console MPEG audio player.

More information: https://manned.org/mpg123.

- Play the specified mp3 files:

```
mpg123 {{path/to/file1.mp3 path/to/file2.mp3 ...}}
```

- Play the mp3 from `stdin`:

```
cat {{file.mp3}} | mpg123 -
```

- Jump forward to the next song:

```
<f>
```

- Jump back to the beginning for the song:

```
<b>
```

- Stop or replay the current file:

```
<s>
```

- Fast forward:

```
<.>
```

- Quit:

```
<q>
```

mpstat

Report CPU statistics.

More information: https://manned.org/mpstat.

- Display CPU statistics every 2 seconds:

```
mpstat {{2}}
```

- Display 5 reports, one by one, at 2 second intervals:

```
mpstat {{2}} {{5}}
```

- Display 5 reports, one by one, from a given processor, at 2 second intervals:

```
mpstat -P {{0}} {{2}} {{5}}
```

mssh

GTK+ based SSH client for interacting with multiple SSH servers at once.

More information: https://manned.org/mssh.

- Open a new window and connect to multiple SSH servers:

```
mssh {{user@host1 user@host2 ...}}
```

- Open a new window and connect to a group of servers predefined in `~/.mssh_clusters`:

```
mssh {{[-a|--alias]}} {{alias_name}}
```

mt

Control magnetic tape drive operation (commonly LTO tape).

More information: https://manned.org/mt.

- Check the status of a tape drive:

```
mt -f {{/dev/nstX}} status
```

- Rewind the tape to beginning:

```
mt -f {{/dev/nstX}} rewind
```

- Move forward a given files, then position the tape on first block of next file:

```
mt -f {{/dev/nstX}} fsf {{count}}
```

- Rewind the tape, then position the tape at beginning of the given file:

```
mt -f {{/dev/nstX}} asf {{count}}
```

- Position the tape at the end of valid data:

```
mt -f {{/dev/nstX}} eod
```

- Rewind the tape and unload/eject it:

```
mt -f {{/dev/nstX}} eject
```

- Write EOF (End-of-file) mark at the current position:

```
mt -f {{/dev/nstX}} eof
```

mux

Intercept and multiplex streams of input events.

More information: https://gitlab.com/interception/linux/tools/-/tree/master#mux.

- Create a new muxer with a specified name:

```
mux -c {{muxer_name1 muxer_name2 ...}}
```

- Set the muxer's internal queue size (default is 100):

```
mux -s {{size}}
```

- Read input from a named muxer (can be repeated in "switch mode"):

```
mux -i {{input_muxer_name}}
```

- Write output to a named muxer (can be repeated):

```
mux -o {{output_muxer_name}}
```

nala

Package management utility with better formatting.

Front-end for the **`python-apt`** API.

More information: https://gitlab.com/volian/nala.

- Install a package, or update it to the latest available version:

```
sudo nala install {{package}}
```

- Remove a package:

```
sudo nala remove {{package}}
```

- Remove a package and its configuration files:

```
nala purge {{package}}
```

- Search package names and descriptions using a word, `regex` (default) or glob:

```
nala search "{{pattern}}"
```

- Update the list of available packages and upgrade the system:

```
sudo nala upgrade
```

- Remove all unused packages and dependencies from your system:

```
sudo nala autoremove
```

- Fetch fast mirrors to improve download speeds:

```
sudo nala fetch
```

- Display the history of all transactions:

```
nala history
```

namcap

Check binary packages and source **PKGBUILD**s for common packaging mistakes.

More information: https://manned.org/namcap.

- Check a specific PKGBUILD file:

```
namcap {{path/to/pkgbuild}}
```

- Check a specific package file:

```
namcap {{path/to/package.pkg.tar.zst}}
```

- Check a file, printing extra informational messages:

```
namcap {{[-i|--info]}} {{path/to/file}}
```

named

Execute the DNS (Dynamic Name Service) server daemon that converts host names to IP addresses and vice versa.

More information: https://manned.org/named.

- Read the default configuration file `/etc/named.conf`, read any initial data and listen for queries:

```
named
```

- Read a custom configuration file:

```
named -c {{path/to/named.conf}}
```

- Use IPv4 or IPv6 only, even if the host machine is capable of utilising other protocols:

```
named {{-4|-6}}
```

- Listen for queries on a specific port instead of the default port 53:

```
named -p {{port}}
```

- Run the server in the foreground and do not daemonize:

```
named -f
```

namei

Follows a pathname (which can be a symbolic link) until a terminal point is found (a file/directory/char device etc).

This program is useful for finding "too many levels of symbolic links" problems.

More information: https://manned.org/namei.

- Resolve the pathnames specified as the argument parameters:

```
namei {{path/to/a}} {{path/to/b}} {{path/to/c}}
```

- Display the results in a long-listing format:

```
namei {{[-l|--long]}} {{path/to/a}} {{path/to/b}} {{path/to/c}}
```

- Show the mode bits of each file type in the style of `ls`:

```
namei {{[-m|--modes]}} {{path/to/a}} {{path/to/b}} {{path/to/c}}
```

- Show owner and group name of each file:

```
namei {{[-o|--owners]}} {{path/to/a}} {{path/to/b}} {{path/to/c}}
```

- Don't follow symlinks while resolving:

```
namei {{[-n|--nosymlinks]}} {{path/to/a}} {{path/to/b}} {{path/to/c}}
```

nautilus

Default file explorer for GNOME desktop environment.

Also known as GNOME Files.

See also: **`dolphin`**, **`caja`**, **`thunar`**, **`vifm`**.

More information: https://manned.org/nautilus.

- Launch Nautilus:

```
nautilus
```

- Launch Nautilus as root user:

```
nautilus admin:/
```

- Launch Nautilus and display a specific directory:

```
nautilus {{path/to/directory}}
```

- Launch Nautilus with a specific file or directory selected:

```
nautilus --select {{path/to/file_or_directory}}
```

- Launch Nautilus in a separated window:

```
nautilus --new-window
```

- Close all Nautilus instances:

```
nautilus --quit
```

- Display help:

```
nautilus --help
```

navi

An interactive cheatsheet tool for the command-line and application launchers.

More information: https://github.com/denisidoro/navi.

- Browse through all available cheatsheets:

```
navi
```

- Browse the cheatsheet for `navi` itself:

```
navi fn welcome
```

- Print a command from the cheatsheet without executing it:

```
navi --print
```

- Output shell widget source code (It automatically detects your shell if possible, but can also be specified manually):

```
navi widget {{shell}}
```

- Autoselect and execute the snippet that best matches a query:

```
navi --query '{{query}}' --best-match
```

ncal

This command is an alias of **`cal`**.

- View documentation for the original command:

  ```
  tldr cal
  ```

ncat

Read, write, redirect, and encrypt data across a network.

An alternative implementation of a similar utility called **`netcat`**/**`nc`**.

More information: https://nmap.org/ncat/guide/index.html.

- Listen for input on the specified port and write it to the specified file:

```
ncat {{[-l|--listen]}} {{port}} > {{path/to/file}}
```

- Accept multiple connections and keep ncat open after they have been closed:

```
ncat {{[-lk|--listen --keep-open]}} {{port}}
```

- Write output of specified file to the specified host on the specified port:

```
ncat {{address}} {{port}} < {{path/to/file}}
```

- Accept multiple incoming connections on an encrypted channel evading detection of traffic content:

```
ncat --ssl {{[-k|--keep-open]}} {{[-l|--listen]}} {{port}}
```

- Connect to an open `ncat` connection over SSL:

```
ncat --ssl {{host}} {{port}}
```

- Check connectivity to a remote host on a particular port with timeout:

```
ncat {{[-w|--wait]}} {{seconds}} {{[-vz|--verbose -z]}}
{{host}} {{port}}
```

ndctl

Utility for managing Non-Volatile DIMMs.

More information: https://manned.org/ndctl.

- Create an `fsdax` mode namespace:

```
ndctl create-namespace --mode={{fsdax}}
```

- Change the mode of a namespace to `raw`:

```
ndctl create-namespace --reconfigure={{namespaceX.Y}} --
mode={{raw}}
```

- Check a sector mode namespace for consistency, and repair if needed:

```
ndctl check-namespace --repair {{namespaceX.Y}}
```

- List all namespaces, regions, and buses (including disabled ones):

```
ndctl list --namespaces --regions --buses --idle
```

- List a specific namespace and include lots of additional information:

```
ndctl list -vvv --namespace={{namespaceX.Y}}
```

- Run a monitor to watch for SMART health events for NVDIMMs on the `ACPI.NFIT` bus:

```
ndctl monitor --bus={{ACPI.NFIT}}
```

- Remove a namespace (when applicable) or reset it to an initial state:

```
ndctl destroy-namespace --force {{namespaceX.Y}}
```

needrestart

Check which daemons need to be restarted after library upgrades.

More information: https://manned.org/needrestart.

- List outdated processes:

```
needrestart
```

- Interactively restart services:

```
sudo needrestart
```

- List outdated processes in [v]erbose or [q]uiet mode:

```
needrestart -{{v|q}}
```

- Check if the [k]ernel is outdated:

```
needrestart -k
```

- Check if the CPU microcode is outdated:

```
needrestart -w
```

- List outdated processes in [b]atch mode:

```
needrestart -b
```

- List outdated processed using a specific [c]onfiguration file:

```
needrestart -c {{path/to/config}}
```

- Display help:

```
needrestart --help
```

nemo

Manage files and directories in Cinnamon desktop environment.

More information: https://manned.org/nemo.

- Open the current user home directory:

```
nemo
```

- Open specific directories in separate windows:

```
nemo {{path/to/directory1 path/to/directory2 ...}}
```

- Open specific directories in tabs:

```
nemo --tabs {{path/to/directory1 path/to/directory2 ...}}
```

- Open a directory with a specific window size:

```
nemo --geometry={{600}}x{{400}} {{path/to/directory}}
```

- Close all windows:

```
nemo --quit
```

nerdctl

Docker-compatible CLI for containerd.

More information: https://github.com/containerd/nerdctl/blob/main/docs/command-reference.md.

- List all containers (running and stopped):

```
nerdctl ps {{[-a|--all]}}
```

- Start a container from an image, with a custom name:

```
nerdctl run --name {{container_name}} {{image}}
```

- Start or stop an existing container:

```
nerdctl {{start|stop}} {{container_name}}
```

- Pull an image from a container registry:

```
nerdctl pull {{image}}
```

- Display the list of already downloaded images:

```
nerdctl images
```

- Open an interactive tty with Bourne shell (`sh`) inside a running container:

```
nerdctl exec {{[-it|--interactive --tty]}} {{container_name}} sh
```

- Remove stopped containers:

```
nerdctl rm {{container1 container2 ...}}
```

- Fetch and follow the logs of a container:

```
nerdctl logs {{[-f|--follow]}} {{container_name}}
```

nethogs

Monitor bandwidth usage per process.

More information: https://github.com/raboof/nethogs.

- Start NetHogs as root (default device is `eth0`):

```
sudo nethogs
```

- Monitor bandwidth on specific device:

```
sudo nethogs {{device}}
```

- Monitor bandwidth on multiple devices:

```
sudo nethogs {{device1 device2 ...}}
```

- Specify refresh rate:

```
sudo nethogs -t {{seconds}}
```

netplan

Network configuration utility using YAML.

More information: https://netplan.readthedocs.io/en/stable/cli/.

- Apply a network configuration and make it persistent:

```
sudo netplan apply
```

- Generate backend configuration files:

```
sudo netplan generate
```

- Configure a network interface to use DHCP:

```
sudo netplan set ethernets.{{interface_name}}.dhcp4=true
```

- Try configuration changes without applying them permanently:

```
sudo netplan try --timeout {{seconds}}
```

- Return to previous working configuration after failed apply:

```
sudo netplan --debug apply
```

- Display the current netplan configuration status:

```
netplan status
```

netselect-apt

Create a **`sources.list`** file for a Debian mirror with the lowest latency.

More information: https://manned.org/netselect-apt.

- Create `sources.list` using the lowest latency server:

```
sudo netselect-apt
```

- Specify Debian branch, stable is used by default:

```
sudo netselect-apt {{testing}}
```

- Include non-free section:

```
sudo netselect-apt {{[-n|--non-free]}}
```

- Specify a country for the mirror list lookup:

```
sudo netselect-apt {{[-c|--country]}} {{India}}
```

netselect

Speed test for choosing a fast network server.

More information: https://github.com/apenwarr/netselect.

- Choose the server with the lowest latency:

```
sudo netselect {{host_1 host_2 ...}}
```

- Display nameserver resolution and statistics:

```
sudo netselect -vv {{host_1 host_2 ...}}
```

- Define maximum TTL (time to live):

```
sudo netselect -m {{10}} {{host_1 host_2 ...}}
```

- Print n fastest servers among the hosts:

```
sudo netselect -s {{n}} {{host_1 host_2 host_3 ...}}
```

- Display help:

```
netselect
```

networkctl

Query the status of network links.

Manage the network configuration using **`systemd-networkd`**.

More information: https://www.freedesktop.org/software/systemd/man/networkctl.html.

- List existing links with their status:

```
networkctl list
```

- Show an overall network status:

```
networkctl status
```

- Bring network devices up:

```
networkctl up {{interface1 interface2 ...}}
```

- Bring network devices down:

```
networkctl down {{interface1 interface2 ...}}
```

- Renew dynamic configurations (e.g. IP addresses received from a DHCP server):

```
networkctl renew {{interface1 interface2 ...}}
```

- Reload configuration files (.netdev and .network):

```
networkctl reload
```

- Reconfigure network interfaces (if you edited the config, you need to call `networkctl reload` first):

```
networkctl reconfigure {{interface1 interface2 ...}}
```

newgrp

Switch primary group membership.

More information: https://manned.org/newgrp.

- Change user's primary group membership:

  `newgrp {{group_name}}`

- Reset primary group membership to user's default group in `/etc/passwd`:

  `newgrp`

newrole

Run a new shell with a different SELinux role.

Allows users to switch to a different SELinux security context.

See also: **`runcon`**, **`semanage-user`**.

More information: https://manned.org/newrole.

- Start a new shell with a specific SELinux role:

```
newrole {{[-r|--role]}} {{role_name}}
```

- Start a new shell with a specific SELinux type:

```
newrole {{[-t|--type]}} {{type_name}}
```

- Start a new shell with a specific SELinux level (format: `s0-s0:c0.c1023` where levels range from `s0` to `s15`, `-` indicates level range, categories start with `c`, `:` separates level from categories, `.` indicates category range):

```
newrole {{[-l|--level]}} {{s0-s0:c0.c1023}}
```

- Display the current SELinux context:

```
id {{[-Z|--context]}}
```

- Start a new shell with both role and type:

```
newrole {{[-r|--role]}} {{role_name}} {{[-t|--type]}} {{type_name}}
```

nft

Allows configuration of tables, chains and rules provided by the Linux kernel firewall.

Nftables replaces iptables.

More information: https://wiki.nftables.org/wiki-nftables/index.php/Main_Page.

- View current configuration:

```
sudo nft list ruleset
```

- Add a new table with family "inet" and table "filter":

```
sudo nft add table {{inet}} {{filter}}
```

- Add a new chain to accept all inbound traffic:

```
sudo nft add chain {{inet}} {{filter}} {{input}} \{ type {{filter}} hook {{input}} priority {{0}} \; policy {{accept}} \; \}
```

- Add a new rule to accept several TCP ports:

```
sudo nft add rule {{inet}} {{filter}} {{input}} {{tcp}} {{dport \{ telnet, ssh, http, https \} accept}}
```

- Add a NAT rule to translate all traffic from the `192.168.0.0/24` subnet to the host's public IP:

```
sudo nft add rule {{nat}} {{postrouting}} ip saddr {{192.168.0.0/24}} {{masquerade}}
```

- Show rule handles:

```
sudo nft --handle --numeric list chain {{family}} {{table}} {{chain}}
```

- Delete a rule:

```
sudo nft delete rule {{inet}} {{filter}} {{input}} handle {{3}}
```

- Save current configuration:

```
sudo nft list ruleset > {{/etc/nftables.conf}}
```

nitch

A small and incredibly fast system fetch written fully in Nim.

More information: https://github.com/ssleert/nitch.

- Display system information (hostname, kernel, uptime, etc.):

```
nitch
```

- Display help:

```
nitch {{[-h|--help]}}
```

- Display version:

```
nitch {{[-v|--version]}}
```

nitrogen

Desktop background browser and setter for X Window.

More information: https://manned.org/nitrogen.

- View and set the wallpapers from a specific directory:

```
nitrogen {{path/to/directory}}
```

- Set the wallpaper with automatic size settings:

```
nitrogen --set-auto {{path/to/file}}
```

- Restore the previous wallpaper:

```
nitrogen --restore
```

nixos-container

Starts NixOS containers using Linux containers.

More information: https://nixos.org/manual/nixos/stable/#ch-containers.

- List running containers:

```
sudo nixos-container list
```

- Create a NixOS container with a specific configuration file:

```
sudo nixos-container create {{container_name}} --config-file {{path/to/nix_config_file}}
```

- Start, stop, terminate, or destroy a specific container:

```
sudo nixos-container {{start|stop|terminate|destroy|status}} {{container_name}}
```

- Run a command in a running container:

```
sudo nixos-container run {{container_name}} -- {{command}} {{command_arguments}}
```

- Update a container configuration:

```
sudo $EDITOR /var/lib/container/{{container_name}}/etc/nixos/configuration.nix && sudo nixos-container update {{container_name}}
```

- Enter an interactive shell session on an already-running container:

```
sudo nixos-container root-login {{container_name}}
```

nixos-option

Inspect a NixOS configuration.

More information: https://nixos.org/manual/nixos/stable/index.html#sec-modularity.

- List all subkeys of a given option key:

```
nixos-option {{option_key}}
```

- List current boot kernel modules:

```
nixos-option boot.kernelModules
```

- List authorized keys for a specific user:

```
nixos-option users.users.{{username}}.openssh.authorizedKeys.{{keyFiles|keys}}
```

- List all remote builders:

```
nixos-option nix.buildMachines
```

- List all subkeys of a given key on another NixOS configuration:

```
NIXOS_CONFIG={{path_to_configuration.nix}} nixos-option {{option_key}}
```

- Show recursively all values of a user:

```
nixos-option {{[-r|--recursive]}} users.users.{{user}}
```

nixos-rebuild

Reconfigure a NixOS machine.

More information: https://nixos.org/nixos/manual/#sec-changing-config.

- Build and switch to the new configuration, making it the boot default:

```
sudo nixos-rebuild switch
```

- Build and switch to the new configuration, making it the boot default and naming the boot entry:

```
sudo nixos-rebuild switch {{[-p|--profile-name]}} {{name}}
```

- Build and switch to the new configuration, making it the boot default and installing updates:

```
sudo nixos-rebuild switch --upgrade
```

- Rollback changes to the configuration, switching to the previous generation:

```
sudo nixos-rebuild switch --rollback
```

- Build the new configuration and make it the boot default without switching to it:

```
sudo nixos-rebuild boot
```

- Build and activate the new configuration, but don't make a boot entry (for testing purposes):

```
sudo nixos-rebuild test
```

- Build the configuration and open it in a virtual machine:

```
sudo nixos-rebuild build-vm
```

- List available generations similar to the boot loader menu:

```
nixos-rebuild list-generations
```

nl

Number lines from a file or from **stdin**.

More information: https://manned.org/nl.

- Number non-blank lines in a file:

```
nl {{path/to/file}}
```

- Read from stdin:

```
{{command}} | nl
```

- Number [a]ll body lines including blank lines or do [n]ot number body lines:

```
nl {{[-b|--body-numbering]}} {{a|n}} {{path/to/file}}
```

- Number only the body lines that match a basic regex (BRE) [p]attern:

```
nl {{[-b|--body-numbering]}} p'FooBar[0-9]' {{path/to/file}}
```

- Use a specific increment for line numbering:

```
nl {{[-i|--line-increment]}} {{increment}} {{path/to/file}}
```

- Specify the line numbering format to [r]ight or [l]eft justified, keeping leading [z]eros or [n]ot:

```
nl {{[-n|--number-format]}} {{rz|ln|rn}}
```

- Specify the line numbering's width (6 by default):

```
nl {{[-w|--number-width]}} {{col_width}} {{path/to/file}}
```

- Use a specific string to separate the line numbers from the lines (TAB by default):

```
nl {{[-s|--number-separator]}} {{separator}} {{path/to/file}}
```

nm-online

Ask NetworkManager whether the network is connected.

More information: https://networkmanager.pages.freedesktop.org/NetworkManager/NetworkManager/nm-online.html.

- Find out whether the network is connected and print the result to `stdout`:

```
nm-online
```

- Wait `n` seconds for a connection (30 by default):

```
nm-online --timeout {{n}}
```

nmcli agent

Run **`nmcli`** as a NetworkManager secret agent or polkit agent.

More information: https://networkmanager.pages.freedesktop.org/NetworkManager/NetworkManager/nmcli.html#agent.

- Register `nmcli` as a secret agent and listen for secret requests:

```
nmcli {{[a|agent]}} {{[s|secret]}}
```

- Register `nmcli` as a polkit agent and listen for authorization requests:

```
nmcli {{[a|agent]}} {{[p|polkit]}}
```

- Register `nmcli` as a secret agent and a polkit agent:

```
nmcli {{[a|agent]}} {{[a|all]}}
```

nmcli connection

Manage connections with NetworkManager.

More information: https://networkmanager.pages.freedesktop.org/NetworkManager/NetworkManager/nmcli.html#connection.

- List all NetworkManager connections (shows name, UUID, type and device):

```
nmcli {{[c|connection]}}
```

- Activate a connection:

```
nmcli {{[c|connection]}} {{[u|up]}} {{uuid}}
```

- Deactivate a connection:

```
nmcli {{[c|connection]}} {{[d|down]}} {{uuid}}
```

- Create an auto-configured dual stack connection:

```
nmcli {{[c|connection]}} {{[a|add]}} ifname
{{interface_name}} type {{ethernet}} ipv4.method {{auto}}
ipv6.method {{auto}}
```

- Create a static IPv6-only connection:

```
nmcli {{[c|connection]}} {{[a|add]}} ifname
{{interface_name}} type {{ethernet}} ip6 {{2001:db8::2/64}}
gw6 {{2001:db8::1}} ipv6.dns {{2001:db8::1}} ipv4.method
{{ignore}}
```

- Create a static IPv4-only connection:

```
nmcli {{[c|connection]}} {{[a|add]}} ifname
{{interface_name}} type {{ethernet}} ip4 {{10.0.0.7/8}} gw4
{{10.0.0.1}} ipv4.dns {{10.0.0.1}} ipv6.method {{ignore}}
```

- Create a VPN connection using OpenVPN from an OVPN file:

```
nmcli {{[c|connection]}} {{[i|import]}} type {{openvpn}} file
{{path/to/vpn_config.ovpn}}
```

nmcli device

Manage network interfaces and establish new Wi-Fi connections using NetworkManager.

More information: https://networkmanager.pages.freedesktop.org/NetworkManager/NetworkManager/nmcli.html#device.

- Print the statuses of all network interfaces:

```
nmcli {{[d|device]}}
```

- Print the available Wi-Fi access points:

```
nmcli {{[d|device]}} {{[w|wifi]}}
```

- Connect to a Wi-Fi network with the specified SSID (you will be prompted for a password):

```
nmcli {{[d|device]}} {{[w|wifi]}} {{[c|connect]}} {{ssid}} {{[-a|--ask]}}
```

- Print the password and QR code for the current Wi-Fi network:

```
nmcli {{[d|device]}} {{[w|wifi]}} {{[s|show-password]}}
```

- Print detailed information about a device:

```
nmcli {{[d|device]}} {{[sh|show]}} {{wlan0}}
```

nmcli general

Manage general settings of NetworkManager.

More information: https://networkmanager.pages.freedesktop.org/NetworkManager/NetworkManager/nmcli.html#general.

- Show the general status of NetworkManager:

```
nmcli {{[g|general]}}
```

- Show the hostname of the current device:

```
nmcli {{[g|general]}} {{[h|hostname]}}
```

- Change the hostname of the current device:

```
sudo nmcli {{[g|general]}} {{[h|hostname]}} {{new_hostname}}
```

- Show the permissions of NetworkManager:

```
nmcli {{[g|general]}} {{[p|permissions]}}
```

- Show the current logging level and domains:

```
nmcli {{[g|general]}} {{[l|logging]}}
```

- Set the logging level and/or domains (see `man NetworkManager.conf` for all available domains):

```
sudo nmcli {{[g|general]}} {{[l|logging]}} {{[l|level]}}
{{INFO|OFF|ERR|WARN|DEBUG|TRACE}} domain
{{domain_1,domain_2,...}}
```

nmcli monitor

Monitor changes to the NetworkManager connection status.

More information: https://networkmanager.pages.freedesktop.org/NetworkManager/NetworkManager/nmcli.html#monitor.

- Start monitoring NetworkManager changes:

```
nmcli {{[m|monitor]}}
```

nmcli networking

Manage the networking status of NetworkManager.

More information: https://networkmanager.pages.freedesktop.org/NetworkManager/NetworkManager/nmcli.html#networking.

- Show the networking status of NetworkManager:

```
nmcli {{[n|networking]}}
```

- Enable or disable networking and all interfaces managed by NetworkManager:

```
nmcli {{[n|networking]}} {{on|off}}
```

- Show the last known connectivity state:

```
nmcli {{[n|networking]}} {{[c|connectivity]}}
```

- Show the current connectivity state:

```
nmcli {{[n|networking]}} {{[c|connectivity]}} {{[c|check]}}
```

nmcli radio

Show the status of radio switches or enable/disable them using NetworkManager.

More information: https://networkmanager.pages.freedesktop.org/NetworkManager/NetworkManager/nmcli.html#radio.

- Show status of Wi-Fi:

```
nmcli {{[r|radio]}} {{[w|wifi]}}
```

- Turn Wi-Fi on or off:

```
nmcli {{[r|radio]}} {{[w|wifi]}} {{on|off}}
```

- Show status of WWAN:

```
nmcli {{[r|radio]}} {{[ww|wwan]}}
```

- Turn WWAN on or off:

```
nmcli {{[r|radio]}} {{[ww|wwan]}} {{on|off}}
```

- Show status of both switches:

```
nmcli {{[r|radio]}}
```

- Turn both switches on or off:

```
nmcli {{[r|radio]}} {{[a|all]}} {{on|off}}
```

nmcli

Manage the network configuration using NetworkManager.

See also: **`nmtui`**, **`iw`**, **`iwctl`**.

More information: https://networkmanager.pages.freedesktop.org/NetworkManager/NetworkManager/nmcli.html.

- View documentation for managing network interfaces and establishing new Wi-Fi connections:

```
tldr nmcli device
```

- View documentation for managing network connections:

```
tldr nmcli connection
```

- View documentation for running `nmcli` as a NetworkManager secret/polkit agent:

```
tldr nmcli agent
```

- View documentation for managing general settings of NetworkManager:

```
tldr nmcli general
```

- View documentation for NetworkManager's activity monitor:

```
tldr nmcli monitor
```

- View documentation for enabling/disabling and checking the status of networking:

```
tldr nmcli networking
```

- View documentation for managing radio switches:

```
tldr nmcli radio
```

nmon

A system administrator, tuner, and benchmark tool.

More information: https://manned.org/nmon.

- Start `nmon`:

  ```
  nmon
  ```

- Save records to file ("-s 300 -c 288" by default):

  ```
  nmon -f
  ```

- Save records to file with a total of 240 measurements, by taking 30 seconds between each measurement:

  ```
  nmon -f -s {{30}} -c {{240}}
  ```

nmtui-connect

This command is an alias of **`nmtui connect`**.

- View documentation for the original command:

  ```
  tldr nmtui
  ```

nmtui-edit

This command is an alias of **`nmtui edit`**.

- View documentation for the original command:

  ```
  tldr nmtui
  ```

nmtui-hostname

This command is an alias of **`nmtui hostname`**.

- View documentation for the original command:

  ```
  tldr nmtui
  ```

nmtui

Text user interface for controlling NetworkManager.

Use **<ArrowKeys>** to navigate, **<Enter>** to select an option.

See also: **nmcli**.

More information: https://networkmanager.pages.freedesktop.org/NetworkManager/NetworkManager/nmtui.html.

- Open the user interface:

```
nmtui
```

- List available connections, with the option to activate or deactivate them:

```
nmtui connect
```

- Connect to a given network:

```
nmtui connect {{name|uuid|device|SSID}}
```

- Edit/Add/Delete a given network:

```
nmtui edit {{name|id}}
```

- Set the system hostname:

```
nmtui hostname
```

nologin

Alternative shell that prevents a user from logging in.

More information: https://manned.org/nologin.5.

- Set a user's login shell to `nologin` to prevent the user from logging in:

```
chsh {{[-s|--shell]}} {{user}} nologin
```

- Customize message for users with the login shell of `nologin`:

```
echo "{{declined_login_message}}" > /etc/nologin.txt
```

nordvpn

Command-line interface for NordVPN.

More information: https://support.nordvpn.com/hc/en-us/articles/20196094470929-Installing-NordVPN-on-Linux-distributions.

- Interactively log into a NordVPN account:

```
nordvpn login
```

- Display the connection status:

```
nordvpn status
```

- Connect to the nearest NordVPN server:

```
nordvpn {{[c|connect]}}
```

- List all available countries:

```
nordvpn countries
```

- Connect to a NordVPN server in a specific country:

```
nordvpn {{[c|connect]}} {{Germany}}
```

- Connect to a NordVPN server in a specific country and city:

```
nordvpn {{[c|connect]}} {{Germany}} {{Berlin}}
```

- Set autoconnect option:

```
nordvpn {{[s|set]}} autoconnect on
```

nova

The OpenStack project that provides a way to provision compute instances.

More information: https://docs.openstack.org/nova/latest/.

- List VMs on current tenant:

```
nova list
```

- List VMs of all tenants (admin user only):

```
nova list --all-tenants
```

- Boot a VM on a specific host:

```
nova boot --nic net-id={{net_id}} --image {{image_id}} --
flavor {{flavor}} --availability-zone nova:{{host_name}}
{{vm_name}}
```

- Start a server:

```
nova start {{server}}
```

- Stop a server:

```
nova stop {{server}}
```

- Attach a network interface to a specific VM:

```
nova interface-attach --net-id {{net_id}} {{server}}
```

nsenter

Run a new command in a running process' namespace.

Particularly useful for Docker images or chroot jails.

More information: https://manned.org/nsenter.

- Run a specific command using the same namespaces as an existing process:

```
nsenter {{[-t|--target]}} {{pid}} {{[-a|--all]}} {{command}} {{command_arguments}}
```

- Run a specific command in an existing process's mount|UTS|IPC|network|PID|user|cgroup|time namespace:

```
nsenter {{[-t|--target]}} {{pid}} --{{mount|uts|ipc|net|pid|user|cgroup}} {{command}} {{command_arguments}}
```

- Run a specific command in an existing process's UTS, time, and IPC namespaces:

```
nsenter {{[-t|--target]}} {{pid}} {{[-u|--uts]}} {{[-T|--time]}} {{[-i|--ipc]}} -- {{command}} {{command_arguments}}
```

- Run a specific command in an existing process's namespace by referencing procfs:

```
nsenter {{[-p|--pid=]}}/proc/{{pid}}/pid/net -- {{command}} {{command_arguments}}
```

nsnake

Snake game in the terminal.

More information: https://github.com/alexdantas/nsnake/.

- Start a snake game:

  `nsnake`

- Navigate the snake:

  `{{<ArrowUp>|<ArrowDown>|<ArrowLeft>|<ArrowRight>}}`

- Pause/unpause the game:

  `<p>`

- Quit the game:

  `<q>`

- Display help during the game:

  `<h>`

nstat

View network statistics.

More information: https://manned.org/nstat.

- View network statistics since last time `nstat` was run:

  `nstat`

- View all-time network statistics:

  `nstat {{[-a|--ignore]}}`

- Display help:

  `nstat {{[-h|--help]}}`

nsxiv

Neo Simple X Image Viewer.

More information: https://nsxiv.codeberg.page/man.

- Open images:

```
nsxiv {{path/to/file1 path/to/file2 ...}}
```

- Open images from directories in image mode:

```
nsxiv {{path/to/directory1 path/to/directory2 ...}}
```

- Search directories recursively for images to view:

```
nsxiv {{[-r|--recursive]}} {{path/to/directory1 path/to/
directory2 ...}}
```

- Quit nsxiv:

```
<q>
```

- Switch to thumbnail mode or open selected image in image mode:

```
<Enter>
```

- Count images forward in image mode:

```
<n>
```

- Count images backward in image mode:

```
<p>
```

ntfsfix

Fix common problems on an NTFS partition.

More information: https://manned.org/ntfsfix.

- Fix a given NTFS partition:

```
sudo ntfsfix {{/dev/sdXN}}
```

ntpd

The official NTP (Network Time Protocol) daemon to synchronize the system clock to remote time servers or local reference clocks.

More information: https://manned.org/ntpd.

- Start the daemon:

```
sudo ntpd
```

- Synchronize system time with remote servers a single time (quit after synchronizing):

```
sudo ntpd --quit
```

- Synchronize a single time allowing "Big" adjustments:

```
sudo ntpd --panicgate --quit
```

ntpdate

Synchronize and set the date and time via NTP.

More information: https://manned.org/ntpdate.

- Synchronize and set date and time:

```
sudo ntpdate {{host}}
```

- Query the host without setting the time:

```
ntpdate -q {{host}}
```

- Use an unprivileged port in case a firewall is blocking privileged ports:

```
sudo ntpdate -u {{host}}
```

- Force time to be stepped using `settimeofday` instead of `slewed`:

```
sudo ntpdate -b {{host}}
```

ntpq

Query the Network Time Protocol (NTP) daemon.

More information: https://manned.org/ntpq.1.

- Start `ntpq` in interactive mode:

```
ntpq
```

- Print a list of NTP peers:

```
ntpq {{[-p|--peers]}}
```

- Print a list of NTP peers without resolving hostnames from IP addresses:

```
ntpq {{[-n|--numeric]}} {{[-p|--peers]}}
```

- Use `ntpq` in debugging mode:

```
ntpq {{[-d|--debug-level]}}
```

- Print NTP system variables values:

```
ntpq {{[-c|--command]}} {{rv}}
```

numactl

Control NUMA policy for processes or shared memory.

More information: https://manned.org/numactl.

- Run a command on node 0 with memory allocated on node 0 and 1:

```
numactl --cpunodebind={{0}} --membind={{0,1}} -- {{command}} {{command_arguments}}
```

- Run a command on CPUs (cores) 0-4 and 8-12 of the current cpuset:

```
numactl --physcpubind={{+0-4,8-12}} -- {{command}} {{command_arguments}}
```

- Run a command with its memory interleaved on all CPUs:

```
numactl --interleave={{all}} -- {{command}} {{command_arguments}}
```

numlockx

Control the number lock key status in X11 sessions.

More information: https://manned.org/numlockx.

- Show the current number lock status:

```
numlockx status
```

- Turn the number lock on:

```
numlockx on
```

- Turn the number lock off:

```
numlockx off
```

- Toggle the current state:

```
numlockx toggle
```

obabel

Translate chemistry-related data.

More information: https://open-babel.readthedocs.io/en/latest/Command-line_tools/babel.html.

- Convert a .mol file to XYZ coordinates:

```
obabel {{path/to/file.mol}} -O {{path/to/output_file.xyz}}
```

- Convert a SMILES string to a 500x500 picture:

```
obabel -:"{{SMILES}}" -O {{path/to/output_file.png}} -xp 500
```

- Convert a file of SMILES string to separate 3D .mol files:

```
obabel {{path/to/file.smi}} -O {{path/to/output_file.mol}} --gen3D -m
```

- Render multiple inputs into one picture:

```
obabel {{path/to/file1 path/to/file2 ...}} -O {{path/to/output_file.png}}
```

objcopy

Copy the contents of an object file to another file.

More information: https://manned.org/objcopy.

- Copy data to another file:

```
objcopy {{path/to/source_file}} {{path/to/target_file}}
```

- Translate object files from one format to another:

```
objcopy --input-target={{input_format}} --output-target
{{output_format}} {{path/to/source_file}} {{path/to/
target_file}}
```

- Strip all symbol information from the file:

```
objcopy --strip-all {{path/to/source_file}} {{path/to/
target_file}}
```

- Strip debugging information from the file:

```
objcopy --strip-debug {{path/to/source_file}} {{path/to/
target_file}}
```

- Copy a specific section from the source file to the destination file:

```
objcopy --only-section {{section}} {{path/to/source_file}}
{{path/to/target_file}}
```

oma

A package management frontend for dpkg-based Linux distributions.

More information: https://github.com/AOSC-Dev/oma#usage.

- Enter the interactive package management interface:

```
sudo oma
```

- Install a package:

```
sudo oma install {{package_name}}
```

- Remove a package:

```
sudo oma remove {{package_name}}
```

- Search for a package:

```
oma search {{keyword}}
```

- Show detailed information for a package:

```
oma show
```

- Upgrade all installed packages to their latest versions:

```
sudo oma upgrade
```

- Update the list of available packages and versions (done automatically before `oma install` and `oma upgrade`):

```
sudo oma refresh
```

- Display help:

```
oma help
```

oniux

Isolate an arbitrary application and route traffic over the Tor network.

Note: This is experimental software.

More information: https://gitlab.torproject.org/tpo/core/oniux.

- Isolate an application:

  ```
  oniux {{command}}
  ```

- Query a website:

  ```
  oniux curl {{https://example.com}}
  ```

- Query an onion site:

  ```
  oniux curl {{http://example.onion}}
  ```

- Run an entire shell in "torified" isolation:

  ```
  oniux bash
  ```

- Isolate graphical applications in desktop environments:

  ```
  oniux hexchat
  ```

oomctl

Analyze the state stored in **`systemd-oomd`**.

More information: https://www.freedesktop.org/software/systemd/man/oomctl.html.

- Show the current state of the cgroups and system contexts stored by `systemd-oomd`:

```
oomctl dump
```

openfortivpn

A VPN client, for Fortinet's proprietary PPP+SSL VPN solution.

More information: https://manned.org/openfortivpn.

- Connect to a VPN with a username and password:

```
openfortivpn {{[-u|--username]}} {{username}} {{[-p|--password]}} {{password}}
```

- Connect to a VPN using a specific configuration file (defaults to `/etc/openfortivpn/config`):

```
sudo openfortivpn {{[-c|--config]}} {{path/to/config}}
```

- Connect to a VPN by specifying the host and port:

```
openfortivpn {{host}}:{{port}}
```

- Trust a given gateway by passing in its certificate's sha256 sum:

```
openfortivpn --trusted-cert {{sha256_sum}}
```

openrc

The OpenRC service manager.

See also: **`rc-status`**, **`rc-update`**, **`rc-service`**.

More information: https://wiki.gentoo.org/wiki/OpenRC.

- Change to a specific runlevel:

```
sudo openrc {{runlevel_name}}
```

- Change to a specific runlevel, but don't stop any existing services:

```
sudo openrc {{[-n|--no-stop]}} {{runlevel_name}}
```

openvpn3

OpenVPN 3 Linux client.

More information: https://community.openvpn.net/openvpn/wiki/OpenVPN3Linux.

- Start a new VPN session:

```
openvpn3 session-start {{[-c|--config]}} {{path/to/
config.conf}}
```

- List established sessions:

```
openvpn3 sessions-list
```

- Disconnect the currently established session started with given configuration:

```
openvpn3 session-manage {{[-c|--config]}} {{path/to/
config.conf}} {{[-D|--disconnect]}}
```

- Import VPN configuration:

```
openvpn3 config-import {{[-c|--config]}} {{path/to/
config.conf}}
```

- List imported configurations:

```
openvpn3 configs-list
```

opera-stable

This command is an alias of **`chromium`**.

More information: https://opera.com.

- View documentation for the original command:

```
tldr chromium
```

opkg

A lightweight package manager used to install OpenWrt packages.

More information: https://openwrt.org/docs/guide-user/additional-software/opkg.

- Install a package:

```
opkg install {{package}}
```

- Remove a package:

```
opkg remove {{package}}
```

- Update the list of available packages:

```
opkg update
```

- Upgrade one or more specific package(s):

```
opkg upgrade {{package(s)}}
```

- Display information for a specific package:

```
opkg info {{package}}
```

- List all the available packages:

```
opkg list
```

- Find out which package owns a file:

```
opkg search /{{path/to/file}}
```

- List all files belonging to a package:

```
opkg files {{package}}
```

optimus-manager

GPU switching utility for Nvidia Optimus laptops.

More information: https://github.com/Askannz/optimus-manager.

- Switch between different GPU modes:

```
optimus-manager --switch {{nvidia|integrated|hybrid}}
```

- Clean up:

```
optimus-manager --cleanup
```

ostree

Version control for binary files similar to **`git`** but optimized for operating system root filesystems.

OSTree is the foundation for immutable image-based operating systems such as Fedora Silverblue, Fedora IoT or Fedora CoreOS.

More information: https://ostreedev.github.io/ostree.

- Initialize a repository of the files in `$PWD` with metadata in `$PWD/path/to/repo`:

```
ostree init --repo {{path/to/repo}}
```

- Create a commit (snapshot) of the files:

```
ostree commit --repo {{path/to/repo}} --branch {{branch_name}}
```

- Show files in commit:

```
ostree ls --repo {{path/to/repo}} {{commit_id}}
```

- Show metadata of commit:

```
ostree show --repo {{path/to/repo}} {{commit_id}}
```

- Show list of commits:

```
ostree log --repo {{path/to/repo}} {{branch_name}}
```

- Show repo summary:

```
ostree summary --repo {{path/to/repo}} --view
```

- Show available refs (branches):

```
ostree refs --repo {{path/to/repo}}
```

pacaur

A utility for Arch Linux to build and install packages from the Arch User Repository.

More information: https://github.com/rmarquis/pacaur.

- Synchronize and update all packages (includes AUR):

```
pacaur -Syu
```

- Synchronize and update only AUR packages:

```
pacaur -Syua
```

- Install a new package (includes AUR):

```
pacaur -S {{package}}
```

- Remove a package and its dependencies (includes AUR packages):

```
pacaur -Rs {{package}}
```

- Search the package database for a keyword (includes AUR):

```
pacaur -Ss {{keyword}}
```

- List all currently installed packages (includes AUR packages):

```
pacaur -Qs
```

paccache

A **pacman** cache cleaning utility.

More information: https://manned.org/paccache.

- Remove all but the 3 most recent package versions from the pacman cache:

  `paccache {{[-r|--remove]}}`

- Set the number of package versions to keep:

  `paccache {{[-rk|--remove --keep]}} {{num_versions}}`

- Perform a dry-run and show the number of candidate packages for deletion:

  `paccache {{[-d|--dryrun]}}`

- Move candidate packages to a directory instead of deleting them:

  `paccache {{[-m|--move]}} {{path/to/directory}}`

paccapability

Query and display the capabilities supported by libalpm.

More information: https://github.com/andrewgregory/pacutils/blob/master/doc/paccapability.pod.

- List all available capabilities:

```
paccapability
```

- Check for the specified capability:

```
paccapability {{nls|downloader|signatures}}
```

- Display help:

```
paccapability --help
```

- Display version:

```
paccapability --version
```

paccheck

Check installed packages on an Arch-based system to verify dependencies, integrity, and consistency.

More information: https://github.com/andrewgregory/pacutils/blob/master/doc/paccheck.pod.

- List and check all installed packages:

```
paccheck
```

- Check the specified packages:

```
paccheck {{package1 package2 ...}}
```

- Only display messages if a problem is found:

```
paccheck --quiet
```

- Check that all package dependencies are satisfied:

```
paccheck --depends
```

- Display help:

```
paccheck --help
```

- Display version:

```
paccheck --version
```

pacconf

Query and display pacman's configuration options, showing either the full configuration or specific directive values as parsed by pacman.

More information: https://github.com/andrewgregory/pacutils/blob/master/doc/pacconf.pod.

- Show full parsed pacman configuration:

```
pacconf
```

- List configured repositories:

```
pacconf --repo-list
```

- Always show directive names even if only one directive is provided:

```
pacconf --verbose {{directive}}
```

- Display only first value of multi-value options:

```
pacconf --single
```

- Display help:

```
pacconf --help
```

- Display version:

```
pacconf --version
```

pacdiff

Maintenance utility for **`.pacorig`**, **`.pacnew`** and **`.pacsave`** files created by **`pacman`**.

More information: https://manned.org/pacdiff.

- Review files that need maintenance in interactive mode:

```
pacdiff
```

- Use sudo and sudoedit to remove and merge files:

```
pacdiff {{[-s|--sudo]}}
```

- Review files needing maintenance, creating `.bak`ups of the original if you `(O)verwrite`:

```
pacdiff {{[-s|--sudo]}} {{[-b|--backup]}}
```

- Use a specific editor to view and merge configuration files (default is `vim -d`):

```
DIFFPROG={{editor}} pacdiff
```

- Scan for configuration files with `locate` instead of using `pacman` database:

```
pacdiff {{[-l|--locate]}}
```

- Display help:

```
pacdiff {{[-h|--help]}}
```

pacfile

Display information about package files, including repository data, file system checks, and database comparisons.

More information: https://github.com/andrewgregory/pacutils/blob/master/doc/pacfile.pod.

- List all available packages:

```
pacfile {{path/to/file_or_directory}}
```

- Compare database values to the file system:

```
pacfile {{path/to/file_or_directory}} --check
```

- Display help:

```
pacfile --help
```

- Display version:

```
pacfile --version
```

pacgraph

Draw a graph of installed packages to PNG/SVG/GUI/console.

More information: https://github.com/keenerd/pacgraph.

- Produce an SVG and PNG graph:

```
pacgraph
```

- Produce an SVG graph:

```
pacgraph --svg
```

- Print summary to console:

```
pacgraph --console
```

- Override the default filename/location (Note: Do not specify the file extension):

```
pacgraph --file={{path/to/file}}
```

- Change the color of packages that are not dependencies:

```
pacgraph --top={{color}}
```

- Change the color of package dependencies:

```
pacgraph --dep={{color}}
```

- Change the background color of a graph:

```
pacgraph --background={{color}}
```

- Change the color of links between packages:

```
pacgraph --link={{color}}
```

pacinfo

Display information about installed packages.

More information: https://github.com/andrewgregory/pacutils/blob/master/doc/pacinfo.pod.

- Display information about a specific package:

```
pacinfo {{package_name}}
```

- Disable low-speed timeouts for downloads:

```
pacinfo --no-timeout {{package_name}}
```

- Display sizes in bytes and date values as Unix timestamps:

```
pacinfo --raw {{package_name}}
```

- Display additional package information:

```
pacinfo --verbose {{package_name}}
```

- Display help:

```
pacinfo --help
```

- Display version:

```
pacinfo --version
```

pacini

Query pacman-style configuration files.

More information: https://github.com/andrewgregory/pacutils/blob/master/doc/pacini.pod.

- Show the full parsed configuration file (default: `stdin`):

```
pacini {{path/to/file}}
```

- List configured sections:

```
pacini {{path/to/file}} --section-list
```

- Always show directive names even if only one directive is provided:

```
pacini {{path/to/file}} --verbose
```

- Display directives listed in a specific section:

```
pacini {{path/to/file}} --section {{section_name}}
```

- Display help:

```
pacini --help
```

- Display version:

```
pacini --version
```

paclock

Lock/unlock the libalpm database (used by **pacman**) to prevent or allow simultaneous package management operations.

More information: https://github.com/andrewgregory/pacutils/blob/master/doc/paclock.pod.

- Lock the database:

```
paclock
```

- Write the lock file path to `stdout` (without locking the database):

```
paclock --print
```

- Unlock the database:

```
paclock --unlock
```

- Display help:

```
paclock --help
```

- Display version:

```
paclock --version
```

paclog

Filter pacman log entries.

More information: https://github.com/andrewgregory/pacutils/blob/master/doc/paclog.pod.

- Display the entire pacman log:

```
paclog
```

- Display pacman-style logged commandline entries:

```
paclog --commandline
```

- Display only errors, warnings, and notes:

```
paclog --warnings
```

- Display package name and action type:

```
paclog --package {{package_name}} --action {{install|upgrade|remove|...}}
```

- Display the list of installed packages according to the log:

```
paclog --pkglist
```

- Display help:

```
paclog --help
```

- Display version:

```
paclog --version
```

pacman -D

This command is an alias of **`pacman --database`**.

- View documentation for the original command:

  ```
  tldr pacman database
  ```

pacman --database

Operate on the Arch Linux package database.

Modify certain attributes of the installed packages.

See also: **pacman**.

More information: https://manned.org/pacman.8.

- Mark a package as implicitly installed:

```
sudo pacman -D --asdeps {{package}}
```

- Mark a package as explicitly installed:

```
sudo pacman -D --asexplicit {{package}}
```

- Chec[k] that all the package dependencies are installed:

```
pacman -Dk
```

- Chec[k] the sync [D]atabase to ensure all specified dependencies of downloadable packages are available:

```
pacman -Dkk
```

- Chec[k] and display in [q]uiet mode (only error messages are displayed):

```
pacman -Dkq
```

- Display [h]elp:

```
pacman -Dh
```

pacman --deptest

Check each dependency specified and return a list of dependencies that are not currently satisfied on the system.

See also: **pacman**.

More information: https://manned.org/pacman.8.

- Print the package names of the dependencies that are not installed:

```
pacman -T {{package1 package2 ...}}
```

- Check if the installed package satisfies the given minimum version:

```
pacman -T "{{bash>=5}}"
```

- Check if a later version of a package is installed:

```
pacman -T "{{bash>5}}"
```

- Display [h]elp:

```
pacman -Th
```

pacman -F

This command is an alias of **`pacman --files`**.

- View documentation for the original command:

  ```
  tldr pacman files
  ```

pacman --files

Arch Linux package manager utility.

See also: **`pacman`**, **`pkgfile`**.

More information: https://manned.org/pacman.8.

- Update the package database:

```
sudo pacman -Fy
```

- Find the package that owns a specific [F]ile:

```
pacman -F {{filename}}
```

- Find the package that owns a specific [F]ile, using a `rege[x]`:

```
pacman -Fx '{{regex}}'
```

- List only the package names:

```
pacman -Fq {{filename}}
```

- [l]ist the [F]iles owned by a specific package:

```
pacman -Fl {{package}}
```

- Display [h]elp:

```
pacman -Fh
```

pacman-key

Wrapper script for GnuPG used to manage pacman's keyring.

See also: **pacman**.

More information: https://manned.org/pacman-key.

- Initialize the pacman keyring:

```
sudo pacman-key --init
```

- Add the default Arch Linux keys:

```
sudo pacman-key --populate
```

- List keys from the public keyring:

```
pacman-key {{[-l|--list-keys]}}
```

- Add the specified keys:

```
sudo pacman-key {{[-a|--add]}} {{path/to/keyfile.gpg}}
```

- Receive a key from a key server:

```
sudo pacman-key {{[-r|--recv-keys]}} "{{uid|name|email}}"
```

- Print the fingerprint of a specific key:

```
pacman-key {{[-f|--finger]}} "{{uid|name|email}}"
```

- Sign an imported key locally:

```
sudo pacman-key --lsign-key "{{uid|name|email}}"
```

- Remove a specific key:

```
sudo pacman-key {{[-d|--delete]}} "{{uid|name|email}}"
```

pacman-mirrors

Generate a **pacman** mirrorlist for Manjaro Linux.

Every run of **pacman-mirrors** requires you to synchronize your database and update your system using **sudo pacman -Syyu**.

See also: **pacman**.

More information: https://wiki.manjaro.org/index.php?title=Pacman-mirrors.

- Generate a mirrorlist using the default settings:

```
sudo pacman-mirrors --fasttrack
```

- Get the status of the current mirrors:

```
pacman-mirrors --status
```

- Display the current branch:

```
pacman-mirrors --get-branch
```

- Switch to a different branch:

```
sudo pacman-mirrors --api --set-branch {{stable|unstable|testing}}
```

- Generate a mirrorlist, only using mirrors in your country:

```
sudo pacman-mirrors --geoip
```

pacman -Q

This command is an alias of **`pacman --query`**.

- View documentation for the original command:

  ```
  tldr pacman query
  ```

pacman --query

Arch Linux package manager utility.

See also: **pacman**.

More information: https://manned.org/pacman.8.

- [Q]uery the local package database and list installed packages and versions:

```
pacman -Q
```

- List only packages and versions that were [e]xplicitly installed:

```
pacman -Qe
```

- Find which package [o]wns a file:

```
pacman -Qo {{filename}}
```

- Display information about an [i]nstalled package:

```
pacman -Qi {{package}}
```

- Display the [l]ist of files owned by a specific package:

```
pacman -Ql {{package}}
```

- List orphan packages (installed as [d]ependencies but unrequired ([t]) by any package and print in [q]uiet mode (only package name is displayed)):

```
pacman -Qdtq
```

- List installed packages foreign ([m]) to the repository database:

```
pacman -Qm
```

- List packages that can be [u]pgraded:

```
pacman -Qu
```

pacman -R

This command is an alias of **`pacman --remove`**.

- View documentation for the original command:

  ```
  tldr pacman remove
  ```

pacman --remove

Arch Linux package manager utility.

See also: **pacman**.

More information: https://manned.org/pacman.8.

- [R]emove a package and its dependencies recur[s]ively:

```
sudo pacman -Rs {{package}}
```

- [R]emove a package and its dependencies. Also do [n]ot save backups of configuration files:

```
sudo pacman -Rsn {{package}}
```

- [R]emove a package without prompting:

```
sudo pacman -R --noconfirm {{package}}
```

- [R]emove orphan packages (installed as [d]ependencies but no[t] required by any package):

```
sudo pacman -Rsn $(pacman -Qdtq)
```

- [R]emove a package and [c]ascade that to all packages that depend on it:

```
sudo pacman -Rc {{package}}
```

- List and [p]rint packages that would be affected (does not [R]emove any packages):

```
pacman -Rp {{package}}
```

- Display [h]elp:

```
pacman -Rh
```

pacman -S

This command is an alias of **`pacman --sync`**.

- View documentation for the original command:

  ```
  tldr pacman sync
  ```

pacman --sync

Arch Linux package manager utility.

See also: **`pacman`**.

More information: https://manned.org/pacman.8.

- Install a new package:

```
sudo pacman -S {{package}}
```

- [S]ynchronize and refresh ([y]) the package database along with a sys[u]pgrade (add `--downloadonly` to only download the packages and not update them):

```
sudo pacman -Syu
```

- Update and [u]pgrade all packages and install a new one without prompting:

```
sudo pacman -Syu --noconfirm {{package}}
```

- [s]earch the package database for a `regex` or keyword:

```
pacman -Ss "{{search_pattern}}"
```

- Display [i]nformation about a package:

```
pacman -Si {{package}}
```

- Overwrite conflicting files during a package update:

```
sudo pacman -Syu --overwrite {{path/to/file}}
```

- Remove not installed packages and unused repositories from the cache (use the flags `Scc` to [c]lean all packages):

```
sudo pacman -Sc
```

- Specify the package version that should be installed:

```
sudo pacman -S {{package}}={{version}}
```

pacman -T

This command is an alias of **`pacman --deptest`**.

- View documentation for the original command:

  ```
  tldr pacman deptest
  ```

pacman -U

This command is an alias of **`pacman --upgrade`**.

- View documentation for the original command:

  ```
  tldr pacman upgrade
  ```

pacman --upgrade

Arch Linux package manager utility.

See also: **pacman**.

More information: https://manned.org/pacman.8.

- Install one or more packages from files:

```
sudo pacman -U {{path/to/package1.pkg.tar.zst path/to/package2.pkg.tar.zst ...}}
```

- Install a package without prompting:

```
sudo pacman -U --noconfirm {{path/to/package.pkg.tar.zst}}
```

- Overwrite conflicting files during a package installation:

```
sudo pacman -U --overwrite {{path/to/file}} {{path/to/package.pkg.tar.zst}}
```

- Install a package, skipping the [d]ependency version checks:

```
sudo pacman -Ud {{path/to/package.pkg.tar.zst}}
```

- Fetch and [p]rint packages that would be affected by upgrade (does not install any packages):

```
pacman -Up {{path/to/package.pkg.tar.zst}}
```

- Display [h]elp:

```
pacman -Uh
```

pacman

Arch Linux package manager utility.

See also: **`pacman-sync`**, **`pacman-remove`**, **`pacman-query`**, **`pacman-upgrade`**, **`pacman-files`**, **`pacman-database`**, **`pacman-deptest`**, **`pacman-key`**, **`pacman-mirrors`**.

For equivalent commands in other package managers, see https://wiki.archlinux.org/title/Pacman/Rosetta.

More information: https://manned.org/pacman.8.

- [S]ynchronize and update all packages:

```
sudo pacman -Syu
```

- Install a new package:

```
sudo pacman -S {{package}}
```

- [R]emove a package and its dependencies:

```
sudo pacman -Rs {{package}}
```

- Search ([s]) the package database for a `regex` or keyword:

```
pacman -Ss "{{search_pattern}}"
```

- Search the database for packages containing a specific [F]ile:

```
pacman -F "{{file_name}}"
```

- List only the [e]xplicitly installed packages and versions:

```
pacman -Qe
```

- List orphan packages (installed as [d]ependencies but not actually required by any package):

```
pacman -Qtdq
```

- Empty the entire `pacman` cache:

```
sudo pacman -Scc
```

pacman4console

A text-based console game inspired by the original Pacman.

More information: https://github.com/YoctoForBeaglebone/pacman4console.

- Start a game at Level 1:

```
pacman4console
```

- Start a game on a certain level (there are nine official levels):

```
pacman4console --level={{level_number}}
```

- Start the pacman4console level editor, saving to a specified text file:

```
pacman4consoleedit {{path/to/level_file}}
```

- Play a custom level:

```
pacman4console --level={{path/to/level_file}}
```

pacrepairdb

Fix corrupted database entries in libalpm database.

More information: https://github.com/andrewgregory/pacutils/blob/master/doc/pacrepairdb.pod.

- Perform a basic repair on a specific package:

```
pacrepairdb {{package_name}}
```

- Update the database entries without extracting or removing any packages:

```
pacrepairdb {{package_name}} --dbonly
```

- Display the packages to be repaired and the cache packages to be used without making any changes:

```
pacrepairdb {{package_name}} --print-only
```

- Display additional progress and debug information:

```
pacrepairdb {{package_name}} --verbose
```

- Display help:

```
pacrepairdb --help
```

- Display version:

```
pacrepairdb --version
```

pacrepairfile

Reset properties on files managed by alpm.

More information: https://github.com/andrewgregory/pacutils/blob/master/doc/pacrepairfile.pod.

- Search for the package and reset the properties of a file:

```
pacrepairfile {{path/to/file}} --package
```

- Reset a file quietly:

```
pacrepairfile {{package_name}} --quiet --package
```

- Reset specific file properties (mode, owner UID, group GID, or modification time):

```
pacrepairfile {{package_name}} --{{mode|gid|mtime|uid}} --package
```

- Display help:

```
pacrepairfile --help
```

- Display version:

```
pacrepairfile --version
```

pacsift

Query and filter packages.

More information: https://github.com/andrewgregory/pacutils/blob/master/doc/pacsift.pod.

- List all available packages:

```
pacsift
```

- Filter packages that provide a given package:

```
pacsift --satisfies {{package}}
```

pacstall

An AUR package manager for Ubuntu.

More information: https://github.com/pacstall/pacstall.

- Search the package database for a package name:

```
pacstall --search {{query}}
```

- Install a package:

```
pacstall --install {{package}}
```

- Remove a package:

```
pacstall --remove {{package}}
```

- Add a repository to the database (only GitHub and GitLab are supported):

```
pacstall --add-repo {{remote_repository_location}}
```

- Update pacstall's scripts:

```
pacstall --update
```

- Update all packages:

```
pacstall --upgrade
```

- Display information about a package:

```
pacstall --cache-info {{package}}
```

- List all installed packages:

```
pacstall --list
```

pacstrap

Install Arch Linux packages in the specified new root directory.

More information: https://manned.org/pacstrap.

- Install the `base` package, the Linux kernel and firmware for common hardware:

```
pacstrap {{path/to/new_root}} {{base}} {{linux}} {{linux-firmware}}
```

- Install the `base` package, the Linux LTS kernel and `base-devel` build tools:

```
pacstrap {{path/to/new_root}} {{base}} {{base-devel}} {{linux-lts}}
```

- Install packages and copy the host's Pacman config to the target:

```
pacstrap -P {{path/to/new_root}} {{packages}}
```

- Install packages without copying the host's mirrorlist to the target:

```
pacstrap -M {{path/to/new_root}} {{packages}}
```

- Use an alternate configuration file for Pacman:

```
pacstrap -C {{path/to/pacman.conf}} {{path/to/new_root}} {{packages}}
```

- Install packages using the package cache on the host instead of on the target:

```
pacstrap -c {{path/to/new_root}} {{packages}}
```

- Initialize an empty `pacman` keyring in the target without copying it from the host:

```
pacstrap -K {{path/to/new_root}} {{packages}}
```

- Install packages in interactive mode (prompts for confirmation):

```
pacstrap -i {{path/to/new_root}} {{packages}}
```

pactree

Package dependency tree viewer for pacman.

More information: https://manned.org/pactree.8.

- Print the dependency tree of a specific package:

```
pactree {{package}}
```

- Print what packages depend on a specific package:

```
pactree {{[-r|--reverse]}} {{package}}
```

- Dump dependencies one per line, skipping duplicates:

```
pactree {{[-u|--unique]}} {{package}}
```

- Include optional dependencies of a specific package and colorize the output:

```
pactree {{[-co|--color --optional]}} {{package}}
```

- Display help:

```
pactree
```

Palworld

Create and start a headless Palworld server.

More information: https://docs.palworldgame.com/settings-and-operation/arguments.

- Start the server with default settings:

```
{{path/to/PalServer.sh}}
```

- Start the server with a specific port and set a maximum player count:

```
{{path/to/PalServer.sh}} -port={{8211}} -players={{1..32}}
```

- Start a public lobby server:

```
{{path/to/PalServer.sh}} -publiclobby
```

- Start the server with performance optimizations for multi-threaded CPUs:

```
{{path/to/PalServer.sh}} -useperfthreads -
NoAsyncLoadingThread -UseMultithreadForDS
```

- Start the server with a specific public IP and port for community servers:

```
{{path/to/PalServer.sh}} -publicip={{ip_address}} -
publicport={{port}}
```

pamac

A command-line utility for the GUI package manager pamac.

If you can't see the AUR packages, enable it in **`/etc/pamac.conf`** or in the GUI.

More information: https://wiki.manjaro.org/index.php/Pamac.

- Install a new package:

  ```
  pamac install {{package_name}}
  ```

- Remove a package and its no longer required dependencies (orphans):

  ```
  pamac remove --orphans {{package_name}}
  ```

- Search the package database for a package:

  ```
  pamac search {{package_name}}
  ```

- List installed packages:

  ```
  pamac list --installed
  ```

- Check for package updates:

  ```
  pamac checkupdates
  ```

- Upgrade all packages:

  ```
  pamac upgrade
  ```

partclone

Copy and restore partitions to and from an image while disregarding empty blocks.

More information: https://manned.org/partclone.

- Copy a partition into an image:

```
sudo partclone.{{ext4|btrfs|fat32|xfs|...}} {{[-c|--clone]}} {{[-s|--source]}} {{/dev/sdXY}} {{[-o|--output]}} {{path/to/backup.img}}
```

- Restore a partition from an image:

```
sudo partclone.{{ext4|btrfs|fat32|xfs|...}} {{[-c|--clone]}} {{[-s|--source]}} {{path/to/backup.img}} {{[-o|--output]}} {{/dev/sdXY}}
```

- Display help:

```
partclone.{{ext4|btrfs|fat32|xfs|...}} {{[-h|--help]}}
```

parted

A partition manipulation program.

See also: **`parted`**, **`partprobe`**.

More information: https://www.gnu.org/software/parted/parted.html.

- Start interactive mode with the specified disk selected:

```
sudo parted {{/dev/sdX}}
```

- Show partition information in interactive mode:

```
print
```

- Select a disk in interactive mode:

```
select {{/dev/sdX}}
```

- Create a 16 GB partition with the specified filesystem in interactive mode (`GPT` partition table):

```
mkpart {{partition_name}} {{btrfs|ext2|ext3|ext4|fat16|fat32|
hfs|hfs+|linux-swap|ntfs|reiserfs|udf|xfs}} {{0%}} {{16G}}
```

- Create a 16 GB partition with the specified filesystem in interactive mode (`MBR` partition table):

```
mkpart {{primary|logical|extended}} {{btrfs|ext2|ext3|ext4|
fat16|fat32|hfs|hfs+|linux-swap|ntfs|reiserfs|udf|xfs}}
{{0%}} {{16G}}
```

- Resize a partition in interactive mode:

```
resizepart {{/dev/sdXN}} {{end_position_of_partition}}
```

- Remove a partition in interactive mode:

```
rm {{/dev/sdXN}}
```

- Display help:

```
?
```

parted

A partition manipulation program.

See also: **`parted.interactive`**, **`partprobe`**.

More information: https://www.gnu.org/software/parted/manual/parted.html#Invoking-Parted.

- List partitions on all block devices:

```
sudo parted {{[-l|--list]}}
```

- Create a new partition table of the specified label-type:

```
sudo parted {{/dev/sdX}} mklabel {{aix|amiga|bsd|dvh|gpt|loop|mac|msdos|pc98|sun}}
```

- Create a new `gpt` partition table with a 500MiB boot partition and give the rest for the system partition (`--script` skips user intervention prompts):

```
sudo parted {{/dev/sdX}} {{[-s|--script]}} mklabel gpt mkpart "{{boot_partition_name}}" 0% 500MiB mkpart "{{system_partition_name}}" 500MiB 100%
```

- Set a partition to have its boot flag turned on:

```
sudo parted {{/dev/sdX}} set {{1}} boot on
```

- Start interactive mode with the specified disk selected:

```
sudo parted {{/dev/sdX}}
```

- Display help:

```
parted {{[-h|--help]}}
```

partprobe

Notify the operating system kernel of partition table changes.

More information: https://manned.org/partprobe.

- Notify the operating system kernel of partition table changes:

```
sudo partprobe
```

- Notify the kernel of partition table changes and show a summary of devices and their partitions:

```
sudo partprobe {{[-s|--summary]}}
```

- Show a summary of devices and their partitions but don't notify the kernel:

```
sudo partprobe {{[-s|--summary]}} {{[-d|--dry-run]}}
```

partx

Parse a partition table and tell the kernel about it.

More information: https://manned.org/partx.

- List the partitions on a block device or disk image:

```
sudo partx {{[-l|--list]}} {{path/to/device_or_disk_image}}
```

- Add all the partitions found in a given block device to the kernel:

```
sudo partx {{[-a|--add]}} {{[-v|--verbose]}} {{path/to/
device_or_disk_image}}
```

- Delete all the partitions found from the kernel (does not alter partitions on disk):

```
sudo partx {{[-d|--delete]}} {{path/to/device_or_disk_image}}
```

paru

An AUR helper and pacman wrapper.

See also: **`pacman`**, **`yay`**.

More information: https://github.com/Morganamilo/paru.

- Interactively search for and install a package:

```
paru {{package_name_or_search_term}}
```

- Synchronize and update all packages:

```
paru
```

- Upgrade AUR packages:

```
paru -Sua
```

- Remove an installed package, it's configuration files, and dependencies:

```
paru -Rns {{package}}
```

- Get information about a package:

```
paru -Si {{package}}
```

- Download `PKGBUILD` and other package source files from the AUR or ABS:

```
paru --getpkgbuild {{package}}
```

- Display the `PKGBUILD` file of a package:

```
paru --getpkgbuild --print {{package}}
```

pasuspender

Temporarily suspends **`pulseaudio`** while another command is running to allow access to alsa.

More information: https://manned.org/pasuspender.

- Suspend PulseAudio while running `jackd`:

```
pasuspender -- {{jackd --driver alsa --device hw:0}}
```

patool

Archive file manager.

Various archive formats can be created, extracted, tested, listed, searched, repacked, and compared.

More information: https://github.com/wummel/patool.

- Extract an archive:

```
patool extract {{path/to/archive}}
```

- Create an archive:

```
patool create {{path/to/archive}} {{path/to/
file_or_directory1 path/to/file_or_directory2 ...}}
```

- List contents of an archive:

```
patool list {{path/to/archive}}
```

- Compare the contents of two archives and display the differences in `stdout`:

```
patool diff {{path/to/archive1}} {{path/to/archive2}}
```

- Search for a string inside the contents of an archive:

```
patool search {{path/to/archive}}
```

paxs

Manage packages across Yay, Flatpak, and Snap.

Supports searching, installing, removing, and upgrading packages.

More information: https://github.com/zamhedonia/paxs.

- Search for a package:

```
paxs {{search_term}}
```

- Upgrade all packages:

```
paxs {{[-u|--upgrade-all]}}
```

- Install a package (prompting for the source):

```
paxs {{[-i|--install]}} {{package}}
```

- Remove a package (prompting for the source):

```
paxs {{[-r|--remove]}} {{package}}
```

- Check for updates across all package managers:

```
paxs {{[-c|--check-update]}}
```

- Display help:

```
paxs {{[-h|--help]}}
```

pct clone

Clone a container.

More information: https://pve.proxmox.com/pve-docs/pct.1.html.

- Clone a container:

```
pct clone {{template_id}} {{copy_id}}
```

- Clone a container with a custom name:

```
pct {{[cl|clone]}} {{template_id}} {{copy_id}} --hostname
{{host_name}}
```

pct config

Print the configuration of a container.

More information: https://pve.proxmox.com/pve-docs/pct.1.html.

- Display the saved configuration:

```
pct config {{100}}
```

- Display the running configuration without pending changes:

```
pct config {{100}} --current
```

- Display configuration of a specific snapshot:

```
pct config {{100}} --snapshot {{snapshot_name}}
```

pct console

Attach to a container TTY.

More information: https://pve.proxmox.com/pve-docs/pct.1.html.

- Attach to a terminal:

```
pct console {{100}}
```

- Detach from a terminal:

```
<Ctrl a><q>
```

pct create

Create LXC containers in Proxmox.

More information: https://pve.proxmox.com/pve-docs/pct.1.html.

- Create a container from a template with 4GB size, give it 512MiB of memory and unlimited access to CPU:

```
pct {{[cr|create]}} {{100}} {{local:vztmpl/distro-name.tar.zst}} --rootfs {{local-lvm}}:4
```

- Create a container from a template and give it a specific memory limit in megabytes:

```
pct {{[cr|create]}} {{100}} {{local:vztmpl/distro-name.tar.zst}} --rootfs {{local-lvm}}:{{4}} --memory {{8192}}
```

- Create a container from a template and give it a hostname and a password:

```
pct {{[cr|create]}} {{100}} {{local:vztmpl/distro-name.tar.zst}} --rootfs {{local-lvm}}:{{4}} --hostname {{hostname}} --password {{password}}
```

- Create a container from a template and give it network access:

```
pct {{[cr|create]}} {{100}} {{local:vztmpl/distro-name.tar.zst}} --rootfs {{local-lvm}}:{{4}} --net0 name={{eth0}},bridge={{vmbr0}},ip={{dhcp|manual|10.0.0.1/24}} --features nesting=1
```

- Start a container immediately after creation:

```
pct {{[cr|create]}} {{100}} {{local:vztmpl/distro-name.tar.zst}} --rootfs {{local-lvm}}:{{4}} --start
```

pct destroy

Destroy a container.

More information: https://pve.proxmox.com/pve-docs/pct.1.html.

- Destroy a container:

```
pct {{[des|destroy]}} {{100}}
```

- Destroy a container even if it's running:

```
pct {{[des|destroy]}} {{100}} --force
```

- Also delete all references to this container:

```
pct {{[des|destroy]}} {{100}} --purge
```

pct exec

Launch a command inside a specified container.

More information: https://pve.proxmox.com/pve-docs/pct.1.html.

- Launch a command in a container:

```
pct {{[ex|exec]}} {{100}} {{command}}
```

- Open a Bash shell in a container:

```
pct {{[ex|exec]}} {{100}} bash
```

- Pass arguments to the command:

```
pct {{[ex|exec]}} {{100}} -- {{command}} {{arguments}}
```

pct resize

Resize container storage.

More information: https://pve.proxmox.com/pve-docs/pct.1.html.

- Resize the container size to 20GB:

```
pct {{[resi|resize]}} {{100}} rootfs 20G
```

- Add 10GB to the container storage:

```
pct {{[resi|resize]}} {{100}} rootfs +10G
```

pct set

Set container options.

More information: https://pve.proxmox.com/pve-docs/pct.1.html.

- Set container to start automatically on boot:

```
pct set {{100}} --onboot
```

- Set a container to have a static IP:

```
pct set {{100}} --net0
name=eth0,bridge=vmbr0,ip={{10.0.0.100/24}},gw={{10.0.0.1}}
```

- Set container memory and CPU limit:

```
pct set {{100}} --memory {{8192}} --cpulimit {{4}}
```

- Mount a host file location in a guest:

```
pct set {{100}} --mp{{0}} /{{path/to/host_directory}},mp=/
{{path/to/guest_mount_point}}
```

- Set container tags:

```
pct set {{100}} --tags {{tag1,tag2,...}}
```

- Remove an option:

```
pct set {{100}} --delete {{net0,mp0,mp1,...}}
```

pct status

Display the status of a container.

More information: https://pve.proxmox.com/pve-docs/pct.1.html.

- Display if the container is running:

```
pct status {{100}}
```

- Display detailed container information:

```
pct status {{100}} --verbose
```

pct template

Convert a container into a template.

More information: https://pve.proxmox.com/pve-docs/pct.1.html.

- Convert a container into a template:

```
pct {{[t|template]}} {{100}}
```

pct

Manage LXC containers in Proxmox.

More information: https://pve.proxmox.com/pve-docs/pct.1.html.

- List all containers:

```
pct list
```

- Start/Stop/Reboot a specific container:

```
pct {{start|stop|reboot}} {{100}}
```

- Access a specific container's shell:

```
pct {{[en|enter]}} {{100}}
```

- Create a container from template with 4GB size:

```
pct {{[cr|create]}} {{100}} {{local:vztmpl/distro-
name.tar.zst}} --rootfs {{local-lvm}}:4
```

- Resize the container's disk to 20G:

```
pct {{[resi|resize]}} {{100}} {{rootfs|mpX}} {{20G}}
```

- Show the configuration of a container, specifying its ID:

```
pct {{[conf|config]}} {{100}}
```

- Snapshot a specific container with description:

```
pct {{[sn|snapshot]}} {{100}} {{my-snapshot}} --description
{{My snapshot description}}
```

- Destroy a container and remove all related resources:

```
pct {{[des|destroy]}} {{100}} --purge
```

pdbedit

Edit the Samba user database.

For simple user add/remove/password, you can also use **`smbpasswd`**.

More information: https://manned.org/pdbedit.

- List all Samba users (use verbose flag to show their settings):

```
sudo pdbedit --list --verbose
```

- Add an existing Unix user to Samba (will prompt for password):

```
sudo pdbedit --user {{username}} --create
```

- Remove a Samba user:

```
sudo pdbedit --user {{username}} --delete
```

- Reset a Samba user's failed password counter:

```
sudo pdbedit --user {{username}} --bad-password-count-reset
```

pdfattach

Add a new attachment (embedded file) to an existing PDF file.

See also: **`pdfdetach`**, **`pdfimages`**, **`pdfinfo`**.

More information: https://manned.org/pdfattach.

- Add a new attachment to an existing PDF file:

```
pdfattach {{path/to/input.pdf}} {{path/to/file_to_attach}}
{{path/to/output.pdf}}
```

- Replace attachment with same name if it exists:

```
pdfattach -replace {{path/to/input.pdf}} {{path/to/
file_to_attach}} {{path/to/output.pdf}}
```

- Display help:

```
pdfattach {{[-h|--help]}}
```

- Display version:

```
pdfattach -v
```

pdfbook2

Create a double sided printable PDF booklet from a PDF.

Note: The booklet needs to be printed double-sided in landscape mode, flipped on the long edge.

More information: https://github.com/jenom/pdfbook2#examples.

- Create a booklet named `file-book.pdf` with sane defaults:

```
pdfbook2 {{path/to/file.pdf}}
```

- Create a booklet with the paper size set to A4:

```
pdfbook2 {{[-p|--paper]}} a4paper {{path/to/file.pdf}}
```

- Create a booklet that has the inner margin reduced to 50 pixels (default = 150px):

```
pdfbook2 {{[-p|--paper]}} a4paper {{[-i|--inner-margin]}} 50 {{path/to/file.pdf}}
```

- Organize a large file with print signatures for binding into a larger booklet (signatures must be divisible by 4):

```
pdfbook2 {{[-p|--paper]}} a4paper --signature {{24}} {{path/to/file.pdf}}
```

pdfcrop

Detect and remove margins in each page in a PDF file.

More information: https://github.com/ho-tex/pdfcrop.

- Automatically detect and remove the margin for each page in a PDF file:

```
pdfcrop {{path/to/input_file.pdf}} {{path/to/
output_file.pdf}}
```

- Set the margins of each page to a specific value:

```
pdfcrop {{path/to/input_file.pdf}} --margins '{{left}}
{{top}} {{right}} {{bottom}}' {{path/to/output_file.pdf}}
```

- Set the margins of each page to a specific value, using the same value for left, top, right and bottom:

```
pdfcrop {{path/to/input_file.pdf}} --margins {{300}} {{path/
to/output_file.pdf}}
```

- Use a user-defined bounding box for cropping instead of automatically detecting it:

```
pdfcrop {{path/to/input_file.pdf}} --bbox '{{left}} {{top}}
{{right}} {{bottom}}' {{path/to/output_file.pdf}}
```

- Use different user-defined bounding boxes for odd and even pages:

```
pdfcrop {{path/to/input_file.pdf}} --bbox-odd '{{left}}
{{top}} {{right}} {{bottom}}' --bbox-even '{{left}} {{top}}
{{right}} {{bottom}}' {{path/to/output_file.pdf}}
```

- Automatically detect margins using a lower resolution for improved performance:

```
pdfcrop {{path/to/input_file.pdf}} --resolution {{72}}
{{path/to/output_file.pdf}}
```

pdfdetach

List or extract attachments (embedded files) from a PDF file.

See also: **`pdfattach`**, **`pdfimages`**, **`pdfinfo`**.

More information: https://manned.org/pdfdetach.

- List all attachments in a file with a specific text encoding:

```
pdfdetach list -enc {{UTF-8}} {{path/to/input.pdf}}
```

- Save specific embedded file by specifying its number:

```
pdfdetach -save {{number}} {{path/to/input.pdf}}
```

- Save specific embedded file by specifying its name:

```
pdfdetach -savefile {{name}} {{path/to/input.pdf}}
```

- Save the embedded file with a custom output filename:

```
pdfdetach -save {{number}} -o {{path/to/output}} {{path/to/
input.pdf}}
```

- Save the attachment from a file secured by owner/user password:

```
pdfdetach -save {{number}} {{-opw|-upw}} {{password}} {{path/
to/input.pdf}}
```

pdftohtml

Convert PDF files into HTML, XML and PNG images.

More information: https://manned.org/pdftohtml.

- Convert a PDF file to an HTML file:

```
pdftohtml {{path/to/file.pdf}} {{path/to/output_file.html}}
```

- Ignore images in the PDF file:

```
pdftohtml -i {{path/to/file.pdf}} {{path/to/
output_file.html}}
```

- Generate a single HTML file that includes all PDF pages:

```
pdftohtml -s {{path/to/file.pdf}} {{path/to/
output_file.html}}
```

- Convert a PDF file to an XML file:

```
pdftohtml -xml {{path/to/file.pdf}} {{path/to/
output_file.xml}}
```

pdftoppm

Convert PDF document pages to portable Pixmap (image formats).

More information: https://manned.org/pdftoppm.

- Specify the range of pages to convert (`n` - first page, `m` - last page):

```
pdftoppm -f {{n}} -l {{m}} {{path/to/file.pdf}}
{{image_name_prefix}}
```

- Convert only the first page of a PDF:

```
pdftoppm -singlefile {{path/to/file.pdf}}
{{image_name_prefix}}
```

- Generate a monochrome PBM file (instead of a color PPM file):

```
pdftoppm -mono {{path/to/file.pdf}} {{image_name_prefix}}
```

- Generate a grayscale PGM file (instead of a color PPM file):

```
pdftoppm -gray {{path/to/file.pdf}} {{image_name_prefix}}
```

- Generate a PNG file instead a PPM file:

```
pdftoppm -png {{path/to/file.pdf}} {{image_name_prefix}}
```

pdfxup

N-up PDF pages.

N-upping means putting multiple pages onto one page by scaling and rotating them into a grid.

More information: https://ctan.org/pkg/pdfxup.

- Create a 2-up PDF:

```
pdfxup {{[-o|--output]}} {{path/to/output.pdf}} {{path/to/input.pdf}}
```

- Create a PDF with 3 columns and 2 lines per page:

```
pdfxup {{[-x|--columns]}} {{3}} {{[-y|--rows]}} {{2}} {{[-o|--output]}} {{path/to/output.pdf}} {{path/to/input.pdf}}
```

- Create a PDF in booklet mode (2-up, and pages are sorted to form a book when folded):

```
pdfxup {{[-b|--booklet]}} {{[-o|--output]}} {{path/to/output.pdf}} {{path/to/input.pdf}}
```

perf

Framework for Linux performance counter measurements.

More information: https://perf.wiki.kernel.org.

- Display basic performance counter stats for a command:

```
perf stat {{gcc hello.c}}
```

- Display system-wide real-time performance counter profile:

```
sudo perf top
```

- Run a command and record its profile into `perf.data`:

```
sudo perf record {{command}}
```

- Record the profile of an existing process into `perf.data`:

```
sudo perf record {{[-p|--pid]}} {{pid}}
```

- Read `perf.data` (created by `perf record`) and display the profile:

```
sudo perf report
```

pg

View files one page at a time.

More information: https://manned.org/pg.

- View a file:

```
pg {{path/to/file}}
```

- Display help:

```
pg {{[-h|--help]}}
```

phar

Create, update or extract PHP archives (PHAR).

More information: https://manned.org/phar.

- Add one or more files or directories to a Phar file:

```
phar add -f {{path/to/phar_file}} {{path/to/
file_or_directory1 path/to/file_or_directory2 ...}}
```

- Display the contents of a Phar file:

```
phar list -f {{path/to/phar_file}}
```

- Delete the specified file or directory from a Phar file:

```
phar delete -f {{path/to/phar_file}} -e {{file_or_directory}}
```

- Compress or uncompress files and directories in a Phar file:

```
phar compress -f {{path/to/phar_file}} -c {{algorithm}}
```

- Get information about a Phar file:

```
phar info -f {{path/to/phar_file}}
```

- Sign a Phar file with a specific hash algorithm:

```
phar sign -f {{path/to/phar_file}} -h {{algorithm}}
```

- Sign a Phar file with an OpenSSL private key:

```
phar sign -f {{path/to/phar_file}} -h openssl -y {{path/to/
private_key}}
```

- Display help and available hashing/compression algorithms:

```
phar help
```

photorec

Deleted file recovery tool.

It is recommended to write recovered files to a disk separate to the one being recovered from.

More information: https://www.cgsecurity.org/wiki/PhotoRec_Step_By_Step.

- Run PhotoRec on a specific device:

```
sudo photorec {{/dev/sdb}}
```

- Run PhotoRec on a disk image (`image.dd`):

```
sudo photorec {{path/to/image.dd}}
```

phpdismod

Disable PHP extensions on Debian-based OSes.

More information: https://salsa.debian.org/php-team/php-defaults.

- Disable the JSON extension for every SAPI of every PHP version:

```
sudo phpdismod {{json}}
```

- Disable the JSON extension for PHP 7.3 with the cli SAPI:

```
sudo phpdismod -v {{7.3}} -s {{cli}} {{json}}
```

phpenmod

Enable PHP extensions on Debian-based OSes.

More information: https://salsa.debian.org/php-team/php-defaults.

- Enable the JSON extension for every SAPI of every PHP version:

```
sudo phpenmod {{json}}
```

- Enable the JSON extension for PHP 7.3 with the cli SAPI:

```
sudo phpenmod -v {{7.3}} -s {{cli}} {{json}}
```

phpquery

PHP extension manager for Debian-based OSes.

More information: https://code.google.com/archive/p/phpquery/wikis/CommandLineInterface.wiki.

- List available PHP versions:

```
sudo phpquery -V
```

- List available SAPIs for PHP 7.3:

```
sudo phpquery -v {{7.3}} -S
```

- List enabled extensions for PHP 7.3 with the cli SAPI:

```
sudo phpquery -v {{7.3}} -s {{cli}} -M
```

- Check if the JSON extension is enabled for PHP 7.3 with the apache2 SAPI:

```
sudo phpquery -v {{7.3}} -s {{apache2}} -m {{json}}
```

physlock

Lock all consoles and virtual terminals.

More information: https://github.com/muennich/physlock.

- Lock every console (require current user or root to unlock):

```
physlock
```

- Mute kernel messages on console while locked:

```
physlock -m
```

- Disable SysRq mechanism while locked:

```
physlock -s
```

- Display a message before the password prompt:

```
physlock -p "{{Locked!}}"
```

- Fork and detach physlock (useful for suspend or hibernate scripts):

```
physlock -d
```

pi

Compute decimal Archimedes' constant Pi.

More information: https://manned.org/pi.

- Display 100 decimal digits of Archimedes' constant Pi:

```
pi
```

- Display a specified number of decimal digits of Archimedes' constant Pi:

```
pi {{number}}
```

- Display recommended readings:

```
pi --bibliography
```

- Display help:

```
pi --help
```

- Display version:

```
pi --version
```

picom

Standalone compositor for Xorg.

More information: https://manned.org/picom.

- Enable `picom` during a session:

```
picom &
```

- Start `picom` as a background process:

```
picom {{[-b|--daemon]}}
```

- Use a custom configuration file:

```
picom --config {{path/to/config_file}}
```

pidof

Get the ID of a process using its name.

More information: https://manned.org/pidof.

- List all process IDs with given name:

```
pidof {{bash}}
```

- List a single process ID with given name:

```
pidof -s {{bash}}
```

- List process IDs including scripts with given name:

```
pidof -x {{script.py}}
```

- Kill all processes with given name:

```
kill $(pidof {{name}})
```

pidstat

Show system resource usage, including CPU, memory, IO etc.

More information: https://manned.org/pidstat.

- Show CPU statistics at a 2 second interval for 10 times:

```
pidstat {{2}} {{10}}
```

- Show page faults and memory utilization:

```
pidstat -r
```

- Show input/output usage per process ID:

```
pidstat -d
```

- Show information on a specific PID:

```
pidstat -p {{PID}}
```

- Show memory statistics for all processes whose command name include "fox" or "bird":

```
pidstat -C "{{fox|bird}}" -r -p ALL
```

pihole

Manage the Pi-hole ad-blocking DNS server.

More information: https://docs.pi-hole.net/main/pihole-command.

- Check Pi-hole's status:

```
pihole status
```

- Update Pi-hole and Gravity:

```
sudo pihole {{[-up|updatePihole]}}
```

- Start or stop the daemon:

```
pihole {{enable|disable}}
```

- Update the lists and flush the cache without restarting the DNS server:

```
pihole reloaddns
```

- Update the list of ad-serving domains:

```
pihole {{[-g|updateGravity]}}
```

- Allow or deny the specified domain:

```
pihole {{allow|deny}} {{example.com}}
```

- Search the lists for a domain:

```
pihole {{[-q|query]}} {{example.com}}
```

- Open a real-time log of connections:

```
pihole {{[-t|tail]}}
```

pinout

View the current Raspberry Pi's GPIO pin-out information on the terminal with an ASCII diagram.

More information: https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#view-a-gpio-pinout-for-your-raspberry-pi.

- View the pinout information and GPIO header diagram for the current Raspberry Pi:

  ```
  pinout
  ```

- Open https://pinout.xyz/ in the default browser:

  ```
  pinout {{[-x|--xyz]}}
  ```

pipewire-pulse

Start a PulseAudio-compatible daemon for PipeWire.

More information: https://docs.pipewire.org/page_man_pipewire-pulse_1.html.

- Start the daemon with default settings:

```
pipewire-pulse
```

- Start the daemon with a specific configuration file:

```
pipewire-pulse {{[-c|--config]}} {{path/to/pipewire-pulse.conf}}
```

- Increase the verbosity level (use multiple times for more verbosity, e.g. `-vvv`):

```
pipewire-pulse {{[-v|--verbose]}}
```

- Display help:

```
pipewire-pulse {{[-h|--help]}}
```

pipewire

Start the PipeWire daemon.

More information: https://docs.pipewire.org/page_man_pipewire_1.html.

- Start the PipeWire daemon:

```
pipewire
```

- Use a different configuration file:

```
pipewire --config {{path/to/file.conf}}
```

- Set the verbosity level (error, warn, info, debug or trace):

```
pipewire -{{v|vv|...|vvvvv}}
```

- Display help:

```
pipewire --help
```

pirut

Graphical frontend for **yum**.

See also: **yum**.

More information: https://manned.org/pirut.

- Launch `pirut`:

```
pirut
```

pivpn

Easy security-hardened OpenVPN setup and manager.

Originally designed for the Raspberry Pi, but works on other Linux devices too.

More information: https://docs.pivpn.io/openvpn/.

- Add a new client device:

  ```
  sudo pivpn add
  ```

- List all client devices:

  ```
  sudo pivpn list
  ```

- List currently connected devices and their statistics:

  ```
  sudo pivpn clients
  ```

- Revoke a previously authenticated device:

  ```
  sudo pivpn revoke
  ```

- Uninstall PiVPN:

  ```
  sudo pivpn uninstall
  ```

pkcon

PackageKit client used by Discover and Gnome software and alternative to **`apt`**.

More information: https://manned.org/pkcon.

- Install a package:

```
pkcon install {{package}}
```

- Remove a package:

```
pkcon remove {{package}}
```

- Refresh the package cache:

```
pkcon refresh
```

- Update packages:

```
pkcon update
```

- Search for a specific package:

```
pkcon search {{package}}
```

- List all available packages:

```
pkcon get-packages
```

pkexec

Execute commands as another user.

Asks for password in a GUI if available.

See also: **`sudo`**, **`run0`**, **`doas`**.

More information: https://polkit.pages.freedesktop.org/polkit/pkexec.1.html.

- Run command as root:

```
pkexec {{command}}
```

- Switch user to root:

```
pkexec
```

- Run command as a specific user:

```
pkexec --user {{username}} {{command}}
```

pkgadd

Add a package to a CRUX system.

More information: https://docs.oracle.com/cd/E88353_01/html/E72487/pkgadd-8.html.

- Install a local software package:

```
pkgadd {{package}}
```

- Update an already installed package from a local package:

```
pkgadd -u {{package}}
```

pkgctl auth

Authenticate **`pkgctl`** with services like GitLab.

More information: https://manned.org/pkgctl-auth.1.

- Authenticate `pkgctl` with the GitLab instance:

```
pkgctl auth login
```

- View authentication status:

```
pkgctl auth status
```

pkgctl build

Build packages inside a clean **`chroot`**.

More information: https://manned.org/pkgctl-build.1.

- Automatically choose the right build script to build packages in a clean `chroot`:

```
pkgctl build
```

- Manually build packages in a clean `chroot`:

```
pkgctl build --arch {{architecture}} --repo {{repository}} --clean
```

pkgctl db update

Update the **pacman** database as final release step for packages that have been transferred and staged on https://repos.archlinux.org.

More information: https://manned.org/pkgctl-db-update.1.

- Update the binary repository as final release step:

```
pkgctl db update
```

pkgctl diff

Compare package files using different modes.

See also: **`pkgctl`**.

More information: https://manned.org/pkgctl-diff.

- Compare package files in tar content list different mode (default):

```
pkgctl diff {{[-l|--list]}} {{path/to/file|pkgname}}
```

- Compare package files in diffoscope different mode:

```
pkgctl diff {{[-d|--diffoscope]}} {{path/to/file|pkgname}}
```

- Compare package files in `.PKGINFO` different mode:

```
pkgctl diff {{[-p|--pkginfo]}} {{path/to/file|pkgname}}
```

- Compare package files in `.BUILDINFO` different mode:

```
pkgctl diff {{[-b|--buildinfo]}} {{path/to/file|pkgname}}
```

pkgctl release

Release step to commit, tag and upload build artifacts.

More information: https://manned.org/pkgctl-release.1.

- Release a build artifact:

```
pkgctl release --repo {{repository}} --message
{{commit_message}}
```

pkgctl repo

Manage Git packaging repositories and their configuration for Arch Linux.

See also: **`pkgctl`**.

More information: https://manned.org/pkgctl-repo.1.

- Clone a package repository (requires setting an SSH key in your Arch Linux GitLab account):

```
pkgctl repo clone {{pkgname}}
```

- Clone a package repository over HTTPS:

```
pkgctl repo clone --protocol https {{pkgname}}
```

- Create a new GitLab package repository and clone it after creation (requires valid GitLab API authentication):

```
pkgctl repo create {{pkgbase}}
```

- Switch a package repository to a specified version:

```
pkgctl repo switch {{version}} {{pkgbase}}
```

- Open a package repository's website:

```
pkgctl repo web {{pkgbase}}
```

pkgctl

Unified command-line frontend for Arch Linux devtools.

More information: https://manned.org/pkgctl.1.

- View documentation for authenticating `pkgctl` with services like GitLab:

```
tldr pkgctl auth
```

- View documentation for building packages inside a clean `chroot`:

```
tldr pkgctl build
```

- View documentation for updating the binary repository as final release step:

```
tldr pkgctl db update
```

- View documentation for comparing package files using different modes:

```
tldr pkgctl diff
```

- View documentation for releasing build artifacts:

```
tldr pkgctl release
```

- View documentation for managing Git packaging repositories and their configuration:

```
tldr pkgctl repo
```

- Display version:

```
pkgctl version
```

pkgfile

Search files from packages in the official repositories on Arch-based systems.

See also: **`pacman files`** describing the usage of **`pacman --files`**.

More information: https://manned.org/pkgfile.

- Synchronize the pkgfile database:

```
sudo pkgfile --update
```

- Search for a package that owns a specific file:

```
pkgfile {{filename}}
```

- List all files provided by a package:

```
pkgfile --list {{package}}
```

- List executables provided by a package:

```
pkgfile --list --binaries {{package}}
```

- Search for a package that owns a specific file using case-insensitive matching:

```
pkgfile --ignorecase {{filename}}
```

- Search for a package that owns a specific file in the `bin` or `sbin` directory:

```
pkgfile --binaries {{filename}}
```

- Search for a package that owns a specific file, displaying the package version:

```
pkgfile --verbose {{filename}}
```

- Search for a package that owns a specific file in a specific repository:

```
pkgfile --repo {{repository_name}} {{filename}}
```

pkginfo

Query the package database on a CRUX system.

More information: https://crux.nu/Main/Handbook3-6#ntoc19.

- List installed packages and their versions:

```
pkginfo {{[-i|--installed]}}
```

- List files owned by a package:

```
pkginfo {{[-l|--list]}} {{package}}
```

- List the owner(s) of files matching a pattern:

```
pkginfo {{[-o|--owner]}} {{pattern}}
```

- Print the footprint of a file:

```
pkginfo -f {{path/to/file}}
```

pkgmk

Make a binary package for use with pkgadd on CRUX.

More information: https://docs.oracle.com/cd/E88353_01/html/E37839/pkgmk-1.html.

- Make and download a package:

```
pkgmk -d
```

- Install the package after making it:

```
pkgmk -d -i
```

- Upgrade the package after making it:

```
pkgmk -d -u
```

- Ignore the footprint when making a package:

```
pkgmk -d -if
```

- Ignore the MD5 sum when making a package:

```
pkgmk -d -im
```

- Update the package's footprint:

```
pkgmk -uf
```

pkgrm

Remove a package from a CRUX system.

More information: https://docs.oracle.com/cd/E88353_01/html/E72487/pkgrm-8.html.

- Remove an installed package:

```
pkgrm {{package}}
```

plasma-apply-colorscheme

Switch Plasma color scheme.

More information: https://invent.kde.org/plasma/plasma-workspace.

- List available color schemes:

```
plasma-apply-colorscheme --list-schemes
```

- Apply a color scheme:

```
plasma-apply-colorscheme {{BreezeLight}}
```

- Display help:

```
plasma-apply-colorscheme --help
```

plasma-apply-cursortheme

Switch Plasma cursor theme.

More information: https://invent.kde.org/plasma/plasma-workspace.

- List available cursor themes:

```
plasma-apply-cursortheme --list-themes
```

- Apply a cursor theme:

```
plasma-apply-cursortheme {{breeze_cursors}}
```

- Display help:

```
plasma-apply-cursortheme --help
```

plasma-apply-desktoptheme

Switch Plasma desktop theme.

More information: https://invent.kde.org/plasma/plasma-workspace.

- List available desktop themes:

```
plasma-apply-desktoptheme --list-themes
```

- Apply a desktop theme:

```
plasma-apply-desktoptheme {{default}}
```

- Display help:

```
plasma-apply-desktoptheme --help
```

plasma-apply-lookandfeel

Switch Plasma global themes.

More information: https://userbase.kde.org/System_Settings/Look_And_Feel.

- List available global themes:

```
plasma-apply-lookandfeel --list
```

- Apply a global theme:

```
plasma-apply-lookandfeel --apply
{{org.example.theme.desktop}}
```

- Operate `plasma-apply-lookandfeel` without a display server:

```
plasma-apply-lookandfeel --platform offscreen
```

- Display help:

```
plasma-apply-lookandfeel --help
```

plasma-apply-wallpaperimage

Switch Plasma wallpaper.

More information: https://invent.kde.org/plasma/plasma-workspace.

- Apply a wallpaper:

```
plasma-apply-wallpaperimage /{{path/to/image}}
```

- Apply a wallpaper with a fill mode:

```
plasma-apply-wallpaperimage --fill-mode {{stretch|
preserveAspectFit|preserveAspectCrop|tile|tileVertically|
tileHorizontally|pad}} /{{path/to/image}}
```

- Display help:

```
plasma-apply-wallpaperimage --help
```

plasmashell

Start and restart Plasma Desktop.

More information: https://invent.kde.org/plasma/plasma-desktop.

- Restart `plasmashell`:

```
systemctl restart --user plasma-plasmashell
```

- Restart `plasmashell` without systemd:

```
plasmashell --replace & disown
```

- Display help:

```
plasmashell {{[-h|--help]}}
```

- Display help, including Qt options:

```
plasmashell --help-all
```

playerctl

Control media players via MPRIS.

More information: https://github.com/altdesktop/playerctl.

- Toggle play:

```
playerctl play-pause
```

- Skip to the next track:

```
playerctl next
```

- Go back to the previous track:

```
playerctl previous
```

- List all players:

```
playerctl {{[-l|--list-all]}}
```

- Send a command to a specific player:

```
playerctl {{[-p|--player]}} {{player_name}} {{play-pause|
next|previous|...}}
```

- Send a command to all players:

```
playerctl {{[-a|--all-players]}} {{play-pause|next|
previous|...}}
```

- Display metadata about the current track:

```
playerctl metadata {{[-f|--format]}} "{{Now playing: \{\
{artist\}\} - \{\{album\}\} - \{\{title\}\}}}"
```

pluma

Edit files in MATE desktop environment.

More information: https://manned.org/pluma.

- Start the editor:

```
pluma
```

- Open specific documents:

```
pluma {{path/to/file1 path/to/file2 ...}}
```

- Open documents using a specific encoding:

```
pluma --encoding {{WINDOWS-1252}} {{path/to/file1 path/to/
file2 ...}}
```

- Print all supported encodings:

```
pluma --list-encodings
```

- Open document and go to a specific line:

```
pluma +{{10}} {{path/to/file}}
```

pmap

Report memory map of a process or processes.

More information: https://manned.org/pmap.

- Print memory map for a specific process ID (PID):

```
pmap {{pid}}
```

- Show the extended format:

```
pmap --extended {{pid}}
```

- Show the device format:

```
pmap --device {{pid}}
```

- Limit results to a memory address range specified by `low` and `high`:

```
pmap --range {{low}},{{high}}
```

- Print memory maps for multiple processes:

```
pmap {{pid1 pid2 ...}}
```

pmount

Mount arbitrary hotpluggable devices as a normal user.

More information: https://manned.org/pmount.

- Mount a device below `/media/` (using device as mount point):

```
pmount {{/dev/to/block/device}}
```

- Mount a device with a specific filesystem type to `/media/label`:

```
pmount {{[-t|--type]}} {{filesystem}} {{/dev/to/block/device}} {{label}}
```

- Mount a CD-ROM (filesystem type ISO9660) in read-only mode:

```
pmount {{[-t|--type]}} iso9660 {{[-r|--read-only]}} {{/dev/cdrom}}
```

- Mount an NTFS-formatted disk, forcing read-write access:

```
pmount {{[-t|--type]}} ntfs {{[-w|--read-write]}} {{/dev/sdX}}
```

- Display all mounted removable devices:

```
pmount
```

po4a-gettextize

Convert a file to a PO file.

More information: https://po4a.org/man/man1/po4a-gettextize.1.php.

- Convert a text file to PO file:

```
po4a-gettextize --format {{text}} --master {{path/to/master.txt}} --po {{path/to/result.po}}
```

- List all available formats:

```
po4a-gettextize --help-format
```

- Convert a text file along with a translated document to a PO file (`-l` option can be provided multiple times):

```
po4a-gettextize --format {{text}} --master {{path/to/master.txt}} --localized {{path/to/translated.txt}} --po {{path/to/result.po}}
```

po4a-translate

Convert a PO file back to documentation format.

The provided PO file should be the translation of the POT file which was produced by **po4a-gettextize**.

More information: https://po4a.org/man/man1/po4a-translate.1.php.

- Convert a translated PO file back to a document:

```
po4a-translate --format {{text}} --master {{path/to/master.doc}} --po {{path/to/result.po}} --localized {{path/to/translated.txt}}
```

- List all available formats:

```
po4a-translate --help-format
```

po4a-updatepo

Update the translation (in PO format) of a documentation.

More information: https://po4a.org/man/man1/po4a-updatepo.1.php.

- Update a PO file according to the modification of its origin file:

```
po4a-updatepo --format {{text}} --master {{path/to/
master.txt}} --po {{path/to/result.po}}
```

- List available formats:

```
po4a-updatepo --help-format
```

- Update several PO files according to the modification of their origin file:

```
po4a-updatepo --format {{text}} --master {{path/to/
master.txt}} --po {{path/to/po1.po}} --po {{path/to/po2.po}}
```

po4a

Update both PO files and translated documents.

More information: https://po4a.org/man/man1/po4a.1.php.

- Update PO files and documents according to the specified configuration file:

```
po4a {{path/to/config_file}}
```

pokego

Display Pokémon sprites in color directly in your terminal.

Inspired by Phoney badger's **`pokemon-colorscripts`** but offers enhanced speed and efficiency.

More information: https://github.com/rubiin/pokego.

- Print a specific Pokémon:

```
pokego --name charizard
```

- Print a specific shiny Pokémon:

```
pokego --name spheal -shiny
```

- Print an alternative form of a Pokémon:

```
pokego --name blastoise --form mega
```

- Print random Pokémon from generations 1-3 (range):

```
pokego --random 1-3
```

- Do not display Pokémon name (default: false):

```
pokego --random 1-3 --no-title
```

portablectl

A systemd utility for managing and deploying portable service images on Linux systems.

More information: https://www.freedesktop.org/software/systemd/man/portablectl.html.

- List available portable service images discovered in the portable image search paths:

```
portablectl list
```

- Attach a portable service image to the host system:

```
portablectl attach {{path/to/image}}
```

- Detach a portable service image from the host system:

```
portablectl detach {{path/to/image|image_name}}
```

- Display details and metadata about a specified portable service image:

```
portablectl inspect {{path/to/image}}
```

- Check if a portable service image is attached to the host system:

```
portablectl is-attached {{path/to/image|image_name}}
```

portageq

Query for information about Portage, the Gentoo Linux package manager.

Queryable Portage-specific environment variables are listed in **`/var/db/repos/gentoo/profiles/info_vars`**.

More information: https://wiki.gentoo.org/wiki/Portageq.

- Display the value of a Portage-specific environment variable:

```
portageq envvar {{variable}}
```

- Display a detailed list of repositories configured with Portage:

```
portageq repos_config /
```

- Display a list of repositories sorted by priority (highest first):

```
portageq get_repos /
```

- Display a specific piece of metadata about an atom (i.e. package name including the version):

```
portageq metadata / {{ebuild|porttree|binary|...}}
{{category}}/{{package}} {{BDEPEND|DEFINED_PHASES|
DEPEND|...}}
```

ports

Update/list the ports tree on a CRUX system.

More information: https://manned.org/ports.

- Update the ports tree:

```
ports -u
```

- List the ports in the current tree:

```
ports -l
```

- Check the differences between installed packages and the ports tree:

```
ports -d
```

postconf

Postfix configuration utility.

This command displays the values of the **`main.cf`** configuration parameters by default and warns about possible mistyped parameter names. It can also change the **`main.cf`** configuration parameter values.

More information: https://manned.org/postconf.

- Specify the directory of the `main.cf` configuration file instead of the default configuration directory:

```
postconf -c {{path/to/configuration_directory}}
```

- Edit the `main.cf` configuration file and update parameter settings with the "name=value" pairs:

```
postconf -e
```

- Print the default parameter settings of the `main.cf` instead of the actual settings:

```
postconf -d
```

- Display parameters only from the specified class. The class can be one of builtin, service, user or all:

```
postconf -C {{class}}
```

- List available SASL plug-in types for the Postfix SMTP server. The plug-in type is selected with the `smtpd_sasl_type` configuration parameter by specifying `cyrus` or `dovecot` as the name:

```
postconf -a
```

- List the names of all supported lookup table types. Lookup tables are specified as `type:name` in configuration files where the type can be `btree`, `cdb`, `hash`, `mysql`, etc:

```
postconf -m
```

postfix

Postfix mail transfer agent (MTA) control program.

See also: **dovecot**, a mail delivery agent (MDA) that integrates with Postfix.

More information: https://www.postfix.org/postfix.1.html.

- Check the configuration:

```
sudo postfix check
```

- Check the status of the Postfix daemon:

```
sudo postfix status
```

- Start Postfix:

```
sudo postfix start
```

- Gracefully stop Postfix:

```
sudo postfix stop
```

- Flush the mail queue:

```
sudo postfix flush
```

- Reload the configuration files:

```
sudo postfix reload
```

poweroff

Power off the system.

More information: https://manned.org/poweroff.

- Power off the system:

```
poweroff
```

- Halt the system (same as `halt`):

```
poweroff --halt
```

- Reboot the system (same as `reboot`):

```
poweroff --reboot
```

- Shut down immediately without contacting the system manager:

```
poweroff {{[-f|--force]}}
```

- Write the wtmp shutdown entry without shutting down the system:

```
poweroff {{[-w|--wtmp-only]}}
```

powerprofilesctl

Make power profiles handling available over D-Bus.

More information: https://gitlab.freedesktop.org/upower/power-profiles-daemon#how-to-use.

- List available power profiles:

```
powerprofilesctl list
```

- Set a specific power profile:

```
powerprofilesctl set {{profile_name}}
```

powerstat

Measures the power consumption of a computer that has a battery power source or supports the RAPL interface.

More information: https://manned.org/powerstat.

- Measure power with the default of 10 samples with an interval of 10 seconds:

```
powerstat
```

- Measure power with custom number of samples and interval duration:

```
powerstat {{interval}} {{number_of_samples}}
```

- Measure power using Intel's RAPL interface:

```
powerstat -R {{interval}} {{number_of_samples}}
```

- Show a histogram of the power measurements:

```
powerstat -H {{interval}} {{number_of_samples}}
```

- Enable all statistics gathering options:

```
powerstat -a {{interval}} {{number_of_samples}}
```

powertop

Optimize battery power usage.

More information: https://github.com/fenrus75/powertop.

- Calibrate power usage measurements:

```
sudo powertop --calibrate
```

- Generate HTML power usage report in the current directory:

```
sudo powertop --html={{power_report.html}}
```

- Tune to optimal settings:

```
sudo powertop --auto-tune
```

- Generate a report for a specified number of seconds (instead of 20 by default):

```
sudo powertop --time={{5}}
```

pridecat

Like cat but more colorful.

More information: https://github.com/lunasorcery/pridecat.

- Print the contents of a file in pride colors to `stdout`:

```
pridecat {{path/to/file}}
```

- Print contents of a file in trans colors:

```
pridecat {{path/to/file}} {{[--trans|--transgender]}}
```

- Alternate between lesbian and bisexual pride flags:

```
pridecat {{path/to/file}} --lesbian {{[--bi|--bisexual]}}
```

- Print contents of a file with the background colors changed:

```
pridecat {{path/to/file}} {{[-b|--background]}}
```

- List directory contents in pride flag colors:

```
ls | pridecat --{{flag}}
```

prime-run

Run a program using an alternative Nvidia graphics card.

More information: https://wiki.archlinux.org/title/PRIME#PRIME_render_offload.

- Run a program using a dedicated Nvidia GPU:

```
prime-run {{command}}
```

- Validate whether the Nvidia card is being used:

```
prime-run glxinfo | grep "OpenGL renderer"
```

print

An alias to a **`run-mailcap`**'s action print.

Originally **`run-mailcap`** is used to process mime-type/file.

More information: https://manned.org/print.

- Print action can be used to print any file on default run-mailcap tool:

```
print {{filename}}
```

- With `run-mailcap`:

```
run-mailcap --action=print {{filename}}
```

prlimit

Get or set process resource soft and hard limits.

Given a process ID and one or more resources, prlimit tries to retrieve and/or modify the limits.

More information: https://manned.org/prlimit.

- Display limit values for all current resources for the running parent process:

```
prlimit
```

- Display limit values for all current resources of a specified process:

```
prlimit {{[-p|--pid]}} {{pid_number}}
```

- Run a command with a custom number of open files limit:

```
prlimit {{[-n|--nofile=]}}{{10}} {{command}}
```

pro

Manage Ubuntu Pro services.

More information: https://manned.org/ubuntu-advantage.1.

- Connect your system to the Ubuntu Pro support contract:

```
sudo pro attach
```

- Display the status of Ubuntu Pro services:

```
pro status
```

- Check if the system is affected by a specific vulnerability (and apply a fix if possible):

```
pro fix {{CVE-number}}
```

- Display the number of unsupported packages:

```
pro security-status
```

- List packages that are no longer available for download:

```
pro security-status --unavailable
```

- List third-party packages:

```
pro security-status --thirdparty
```

proctl

Manage projects licenses and languages, switch between templated licenses.

More information: https://github.com/HeCodes2Much/proctl.

- List available licenses:

```
proctl {{[-ll|-list-licenses]}}
```

- List available languages:

```
proctl {{[-lL|-list-languages]}}
```

- Pick a license in a FZF menu:

```
proctl {{[-pl|-pick-license]}}
```

- Pick a language in a FZF menu:

```
proctl {{[-pL|-pick-language]}}
```

- Remove all licenses from the current project:

```
proctl {{[-r|-remove-license]}}
```

- Create a new license template:

```
proctl {{[-t|-new-template]}}
```

- Delete a license from templates:

```
proctl {{[-R|-delete-license]}} {{@license_name1
@license_name2 ...}}
```

- Display help:

```
proctl {{[-h|-help]}}
```

protontricks

A simple wrapper that runs Winetricks commands for Proton enabled games.

More information: https://github.com/Matoking/protontricks#usage.

- Run the protontricks GUI:

```
protontricks --gui
```

- Run Winetricks for a specific game:

```
protontricks {{appid}} {{winetricks_args}}
```

- Run a command within a game's installation directory:

```
protontricks {{[-c|--command]}} {{command}} {{appid}}
```

- List all installed games:

```
protontricks {{[-l|--list]}}
```

- Search for a game's App ID by name:

```
protontricks {{[-s|--search]}} {{game_name}}
```

- Run an executable in the proton environment of a specific game:

```
protontricks-launch --appid {{appid}} {{path/to/
executable.exe}}
```

- Display help:

```
protontricks {{[-h|--help]}}
```

protonvpn connect

Connect to ProtonVPN.

More information: https://github.com/Rafficer/linux-cli-community.

- Connect to ProtonVPN interactively:

```
protonvpn {{[c|connect]}}
```

- Connect to ProtonVPN using the fastest server available:

```
protonvpn {{[c|connect]}} {{[-f|--fastest]}}
```

- Connect to ProtonVPN using a specific server with a specific protocol:

```
protonvpn {{[c|connect]}} {{server_name}} -p {{udp|tcp}}
```

- Connect to ProtonVPN using a random server with a specific protocol:

```
protonvpn {{[c|connect]}} {{[-r|--random]}} -p {{udp|tcp}}
```

- Connect to ProtonVPN using the fastest Tor-supporting server:

```
protonvpn {{[c|connect]}} --tor
```

- Display help:

```
protonvpn {{[c|connect]}} --help
```

protonvpn

Unofficial third-party ProtonVPN client.

See also: **`protonvpn-connect`**.

More information: https://github.com/Rafficer/linux-cli-community.

- Initialize ProtonVPN profile:

```
protonvpn init
```

- Connect to ProtonVPN interactively:

```
protonvpn {{c|connect}}
```

- Display connection status:

```
protonvpn {{s|status}}
```

- Disconnect from ProtonVPN:

```
protonvpn {{d|disconnect}}
```

- Reconnect or connect to the last server used:

```
protonvpn {{r|reconnect}}
```

- Refresh OpenVPN configuration and server data:

```
protonvpn refresh
```

- Display help for a subcommand:

```
protonvpn {{subcommand}} --help
```

prt-get

The CRUX package manager.

More information: https://crux.nu/doc/prt-get%20-%20User%20Manual.html.

- Install a package:

```
prt-get install {{package}}
```

- Install a package with dependency handling:

```
prt-get depinst {{package}}
```

- Update a package manually:

```
prt-get upgrade {{package}}
```

- Remove a package:

```
prt-get remove {{package}}
```

- Upgrade the system from the local ports tree:

```
prt-get sysup
```

- Search the ports tree:

```
prt-get search {{query}}
```

- Search for a file in a package:

```
prt-get fsearch {{file}}
```

pstoedit

Convert PDF files into various image formats.

More information: http://www.pstoedit.net.

- Convert a PDF page to PNG or JPEG format:

```
pstoedit -page {{page_number}} -f magick {{path/to/file.pdf}} {{page.png|page.jpg}}
```

- Convert multiple PDF pages to numbered images:

```
pstoedit -f magick {{path/to/file}} {{page%d.png|page%d.jpg}}
```

pstree

A convenient tool to show running processes as a tree.

More information: https://manned.org/pstree.

- Display a tree of all processes (rooted at init):

```
pstree
```

- Display a tree of processes with PIDs:

```
pstree {{[-p|--show-pids]}}
```

- Display all process trees rooted at processes owned by specified user:

```
pstree {{user}}
```

- Display command line arguments:

```
pstree {{[-a|--arguments]}}
```

- Display children of a specified process:

```
pstree {{pid}}
```

- Display parents of a specified process:

```
pstree {{[-s|--show-parents]}} {{pid}}
```

ptx

Generate a permuted index of words from text files.

More information: https://www.gnu.org/software/coreutils/manual/html_node/ptx-invocation.html.

- Generate a permuted index where the first field of each line is an index reference:

```
ptx {{[-r|--references]}} {{path/to/file}}
```

- Generate a permuted index with automatically generated index references:

```
ptx {{[-A|--auto-reference]}} {{path/to/file}}
```

- Generate a permuted index with a fixed width:

```
ptx {{[-w|--width]}} {{width_in_columns}} {{path/to/file}}
```

- Generate a permuted index with a list of filtered words:

```
ptx {{[-o|--only-file]}} {{path/to/filter}} {{path/to/file}}
```

- Generate a permuted index with SYSV-style behaviors:

```
ptx {{[-G|--traditional]}} {{path/to/file}}
```

ptyxis

A container-oriented terminal for GNOME.

More information: https://gitlab.gnome.org/chergert/ptyxis#basic-usage--command-line-options.

- Open a new Ptyxis window:

```
ptyxis --new-window
```

- Execute a specific command in a new terminal window:

```
ptyxis {{[-x|--execute]}} {{command}}
```

- Open new tab in the last opened window:

```
ptyxis --tab
```

- Set the title for a new:

```
ptyxis --tab {{[-T|--title]}} {{title}}
```

- Specify the working directory for a new tab, window, or command execution:

```
ptyxis {{[-d|--working-directory]}} {{path/to/directory}} --tab
```

pulseaudio-ctl

Control PulseAudio volume.

More information: https://github.com/graysky2/pulseaudio-ctl.

- Increase volume by 5%:

```
pulseaudio-ctl up
```

- Increase volume by a specific amount:

```
pulseaudio-ctl up {{amount}}
```

- Decrease volume by 5%:

```
pulseaudio-ctl down
```

- Decrease volume by a specific amount:

```
pulseaudio-ctl down {{amount}}
```

- Set volume to a specific percentage:

```
pulseaudio-ctl set {{percentage}}
```

- Set volume to a specific percentage if the current volume is higher than the provided value:

```
pulseaudio-ctl atmost {{percentage}}
```

- Toggle mute:

```
pulseaudio-ctl mute
```

- Toggle microphone mute:

```
pulseaudio-ctl mute-input
```

pulseaudio

The PulseAudio sound system daemon and manager.

More information: https://manned.org/pulseaudio.

- Check if PulseAudio is running (a non-zero exit code means it is not running):

```
pulseaudio --check
```

- Start the PulseAudio daemon in the background:

```
pulseaudio --start
```

- Kill the running PulseAudio daemon:

```
pulseaudio {{[-k|--kill]}}
```

- List available modules:

```
pulseaudio --dump-modules
```

- Load a module into the currently running daemon with the specified arguments:

```
pulseaudio {{[-L|--load]}} "{{module_name}} {{arguments}}"
```

pvchange

Change attributes of physical volume(s).

More information: https://manned.org/pvchange.

- Allow allocation on a physical volume:

```
pvchange {{[-x|--allocatable]}} y {{/dev/sdXN}}
```

- Disallow allocation on a physical volume:

```
pvchange {{[-x|--allocatable]}} n {{/dev/sdXN}}
```

- Ignore metadata areas on a physical volume:

```
pvchange --metadataignore y {{/dev/sdXN}}
```

- Stop ignoring metadata areas on a physical volume:

```
pvchange --metadataignore n {{/dev/sdXN}}
```

- Add a tag to a physical volume:

```
pvchange --addtag {{tag}} {{/dev/sdXN}}
```

- Generate a new UUID for a physical volume (use with care):

```
pvchange --uuid {{/dev/sdXN}}
```

- Change all visible physical volumes (combine with other options such as allocatable):

```
pvchange {{[-a|--all]}} {{[-x|--allocatable]}} y
```

pvck

Check and repair LVM metadata on physical volumes.

More information: https://manned.org/pvck.

- Print LVM on-disk headers and structures (label, PV header, MDA header, metadata area):

```
pvck {{/dev/sdXN}} --dump headers
```

- Print the current VG metadata text:

```
pvck {{/dev/sdXN}} --dump metadata
```

- List all metadata versions found in the metadata area:

```
pvck {{/dev/sdXN}} --dump metadata_all
```

- Search common locations for metadata when headers may be damaged, and save it to a file:

```
pvck {{/dev/sdXN}} --dump metadata_search {{[-f|--file]}}
{{path/to/metadata.txt}}
```

- Select the second metadata area (mda2) when printing metadata:

```
pvck {{/dev/sdXN}} --dump metadata --settings "mda_num=2"
```

- Repair headers and metadata using a metadata input file (use with care):

```
pvck {{/dev/sdXN}} --repair {{[-f|--file]}} {{path/to/
metadata_file}}
```

- Repair only the PV header and label header:

```
pvck {{/dev/sdXN}} --repairtype pv_header
```

pvcreate

Initialize a disk or partition for use as a physical volume.

See also: **`lvm`**.

More information: https://manned.org/pvcreate.

- Initialize the `/dev/sda1` volume for use by LVM:

```
pvcreate {{/dev/sdXY}}
```

- Force the creation without any confirmation prompts:

```
pvcreate {{[-f|--force]}} {{/dev/sdXY}}
```

pvdisplay

Display information about Logical Volume Manager (LVM) physical volumes.

See also: **`lvm`**.

More information: https://manned.org/pvdisplay.

- Display information about all physical volumes:

```
sudo pvdisplay
```

- Display information about the physical volume on drive `/dev/sdXY`:

```
sudo pvdisplay {{/dev/sdXY}}
```

pve-firewall

Manage Proxmox VE Firewall.

More information: https://pve.proxmox.com/wiki/Firewall.

- Compile and print all firewall rules:

  `pve-firewall {{[c|compile]}}`

- Show information about the local network:

  `pve-firewall {{[l|localnet]}}`

- Restart the Proxmox VE Firewall service:

  `pve-firewall {{[r|restart]}}`

- Start the Proxmox VE Firewall service:

  `pve-firewall start`

- Stop the Proxmox VE Firewall service:

  `pve-firewall stop`

- Simulate all firewall rules:

  `pve-firewall {{[si|simulate]}}`

- Show the status of Proxmox VE Firewall:

  `pve-firewall status`

pveam

Manage LXC container templates.

More information: https://pve.proxmox.com/pve-docs/pveam.1.html.

- Update container template database:

```
pveam {{[u|update]}}
```

- List available templates:

```
pveam {{[a|available]}}
```

- Download a template:

```
pveam {{[d|download]}} {{local}} {{template_name}}
```

- List downloaded templates:

```
pveam {{[l|list]}} {{local}}
```

- List available templates in a specific section:

```
pveam {{[a|available]}} --section {{system|turnkeylinux|
mail}}
```

- Remove a template:

```
pveam {{[r|remove]}} {{local}}:{{vztmpl}}/{{template_name}}
```

pvecm

Proxmox VE Cluster Manager.

More information: https://pve.proxmox.com/pve-docs/pvecm.1.html.

- Add the current node to an existing cluster:

```
pvecm add {{hostname_or_ip}}
```

- Add a node to the cluster configuration (internal use):

```
pvecm {{[addn|addnode]}} {{node}}
```

- Display the version of the cluster join API available on this node:

```
pvecm {{[ap|apiver]}}
```

- Generate new cluster configuration:

```
pvecm {{[c|create]}} {{clustername}}
```

- Remove a node from the cluster configuration:

```
pvecm {{[d|delnode]}} {{node}}
```

- Display the local view of the cluster nodes:

```
pvecm {{[n|nodes]}}
```

- Display the local view of the cluster status:

```
pvecm {{[s|status]}}
```

pveperf

A benchmarking tool in Proxmox Server. Gather CPU and hard disk performance data for the hard disk.

More information: https://pve.proxmox.com/pve-docs/pveperf.1.html.

- Show CPU and hard disk performance data for the hard disk mounted at `/`:

```
pveperf
```

pvesh

Interface with the Proxmox VE API.

More information: https://pve.proxmox.com/pve-docs/pvesh.1.html.

- List available nodes:

```
pvesh {{[g|get]}} /nodes
```

- Display detailed information about containers or virtual machines:

```
pvesh {{[g|get]}} /nodes/{{node_name}}/{{lxc|qemu}}
```

- Discover API paths:

```
pvesh {{[l|ls]}} {{/}}
```

- Display API path usage instructions:

```
pvesh {{[u|usage]}} {{/pools}}
```

pvesm

Manage Proxmox storage.

More information: https://pve.proxmox.com/pve-docs/pvesm.1.html.

- Get status for all datastores:

```
pvesm {{[st|status]}}
```

- List storage contents:

```
pvesm {{[l|list]}} {{storage_name}}
```

- Add a directory storage:

```
pvesm add {{[d|dir]}} {{storage_name}} --path {{path/to/
directory}}
```

- Set a storage to contain specific content:

```
pvesm set {{storage_name}} --content
{{iso,images,backup,vztmpl,...}}
```

- Delete a file from storage:

```
pvesm free {{local:iso/archlinux-2025.08.01-x86_64.iso}}
```

- Remove a storage:

```
pvesm {{[r|remove]}} {{storage_name}}
```

pveum

Manage proxmox users.

More information: https://pve.proxmox.com/pve-docs/pveum.1.html.

- List users:

```
pveum {{[u|user]}} {{[l|list]}}
```

- Add a user:

```
pveum {{[u|user]}} {{[a|add]}} {{username}}@pve
```

- Delete a user:

```
pveum {{[u|user]}} {{[d|delete]}} {{username}}@pve
```

pveversion

Proxmox VE version info.

More information: https://pve.proxmox.com/wiki/Command_Line_Tools#Software_version_check.

- Print system version:

```
pveversion
```

- Print Proxmox subpackage versions:

```
pveversion {{[-v|--verbose]}}
```

pvremove

Remove LVM labels from physical volume(s).

More information: https://manned.org/pvremove.

- Remove a LVM label from a physical volume:

```
sudo pvremove {{/dev/sdXY}}
```

- Display detailed output during the operation:

```
sudo pvremove {{[-v|--verbose]}} {{/dev/sdXY}}
```

- Remove a LVM label without asking for confirmation:

```
sudo pvremove {{[-y|--yes]}} {{/dev/sdXY}}
```

- Forcefully remove a LVM label:

```
sudo pvremove {{[-f|--force]}} {{/dev/sdXY}}
```

- Display output in JSON format:

```
sudo pvremove --reportformat json {{/dev/sdXY}}
```

pvs

Display information about physical volumes.

See also: **`lvm`**.

More information: https://manned.org/pvs.

- Display information about physical volumes:

```
sudo pvs
```

- Display non-physical volumes:

```
sudo pvs {{[-a|--all]}}
```

- Change default display to show more details:

```
sudo pvs {{[-v|--verbose]}}
```

- Display only specific fields:

```
sudo pvs {{[-o|--options]}} {{field_name_1}},{{field_name_2}}
```

- Append field to default display:

```
sudo pvs {{[-o|--options]}} +{{field_name}}
```

- Suppress heading line:

```
sudo pvs --noheadings
```

- Use separator to separate fields:

```
sudo pvs --separator {{special_character}}
```

pvscan

List all physical volumes and manage their online status.

More information: https://manned.org/pvscan.

- List all physical volumes:

```
pvscan
```

- Show the volume group that uses a specific physical volume:

```
pvscan --cache --listvg {{/dev/sdX}}
```

- Show logical volumes that use a specific physical volume:

```
pvscan --cache --listlvs {{/dev/sdX}}
```

- Display detailed information in JSON format:

```
pvscan --reportformat json
```

pw-cat

Play and record audio files through PipeWire.

See also: **`wpctl`**, **`pw-cli`**.

More information: https://docs.pipewire.org/page_man_pw-cat_1.html.

- Play a WAV file over the default target:

```
pw-cat {{[-p|--playback]}} {{path/to/file.wav}}
```

- Play a WAV file with a specified resampler quality (4 by default):

```
pw-cat {{[-q|--quality]}} {{0..15}} {{[-p|--playback]}} {{path/to/file.wav}}
```

- Record a sample recording at a volume level of 125%:

```
pw-cat {{[-r|--record]}} --volume {{1.25}} {{path/to/file.wav}}
```

- Record a sample recording using a different sample rate:

```
pw-cat {{[-r|--record]}} --rate {{6000}} {{path/to/file.wav}}
```

- Display help:

```
pw-cat {{[-h|--help]}}
```

pw-cli

Manage a PipeWire instance's modules, objects, nodes, devices, links and much more.

See also: **`wpctl`**.

More information: https://docs.pipewire.org/page_man_pw-cli_1.html.

- Print information of all object of a specific type:

```
pw-cli {{[ls|list-objects]}} {{Node|Link|Port|Client|Device|Metadata|Factory|Module|Profiler|SecurityContext|Core}}
```

- Print information about an object with a specific ID:

```
pw-cli {{[i|info]}} {{4}}
```

- Print all objects' information:

```
pw-cli {{[i|info]}} all
```

- Monitor for object changes:

```
pw-cli {{[-m|--monitor]}}
```

- Display help:

```
pw-cli {{[h|help]}}
```

pw-config

List configuration paths and sections that will be used by the PipeWire server and clients.

More information: https://docs.pipewire.org/page_man_pw-config_1.html.

- List all configuration files that will be used:

```
pw-config
```

- List all configuration files that will be used by the PipeWire PulseAudio server:

```
pw-config {{[-n|--name]}} pipewire-pulse.conf
```

- List all configuration sections used by the PipeWire PulseAudio server:

```
pw-config {{[-n|--name]}} pipewire-pulse.conf list
```

- List the `context.properties` fragments used by the JACK clients:

```
pw-config {{[-n|--name]}} jack.conf list context.properties
```

- List the merged `context.properties` used by the JACK clients:

```
pw-config {{[-n|--name]}} jack.conf merge context.properties
```

- List the merged `context.modules` used by the PipeWire server and reformat:

```
pw-config {{[-n|--name]}} pipewire.conf {{[-r|--recurse]}} merge context.modules
```

- Display help:

```
pw-config {{[-h|--help]}}
```

pw-container

Run a program in a new security context.

More information: https://docs.pipewire.org/page_man_pw-container_1.html.

- Create a new security context and print its socket address to `stdout`:

```
pw-container
```

- Run a specific program within a new security context:

```
pw-container {{command}} {{argument1 argument2 ...}}
```

- Run a program, connecting to a specific remote PipeWire instance:

```
pw-container {{[-r|--remote]}} {{remote_instance_name}}
{{command}}
```

- Run a program in a new context with specific properties using a JSON object:

```
pw-container {{[-P|--properties]}} '{{{"key":
"value"}}}' {{command}}
```

- Display help:

```
pw-container {{[-h|--help]}}
```

pw-dot

Create **`.dot`** files of the PipeWire graph.

See also: **`dot`** for rendering graph.

More information: https://docs.pipewire.org/page_man_pw-dot_1.html.

- Generate a graph to `pw.dot` file:

```
pw-dot
```

- Read objects from `pw-dump` JSON file:

```
pw-dot {{[-j|--json]}} {{path/to/file.json}}
```

- Specify an output file, showing all object types:

```
pw-dot {{[-o|--output]}} {{path/to/file.dot}} {{[-a|--all]}}
```

- Print `.dot` graph to `stdout`, showing all object properties:

```
pw-dot {{[-o|--output]}} - {{[-d|--detail]}}
```

- Generate a graph from a remote instance, showing only linked objects:

```
pw-dot {{[-r|--remote]}} {{remote_name}} {{[-s|--smart]}}
```

- Lay the graph from left to right, instead of dot's default top to bottom:

```
pw-dot {{[-L|--lr]}}
```

- Lay the graph using 90-degree angles in edges:

```
pw-dot {{[-9|--90]}}
```

- Display help:

```
pw-dot {{[-h|--help]}}
```

pw-dump

Dump PipeWire's current state as JSON, including the information on nodes, devices, modules, ports, and other objects.

See also: **pw-mon**.

More information: https://docs.pipewire.org/page_man_pw-dump_1.html.

- Print a JSON representation of the default PipeWire instance's current state:

```
pw-dump
```

- Print a JSON representation of an object:

```
pw-dump {{object_id}}
```

- Dump the current state monitoring changes, printing it again:

```
pw-dump {{[-m|--monitor]}}
```

- Dump the current state of a remote instance to a file:

```
pw-dump {{[-r|--remote]}} {{remote_name}} > {{path/to/dump_file.json}}
```

- Set a color configuration:

```
pw-dump {{[-C|--color]}} {{never|always|auto}}
```

- Display help:

```
pw-dump {{[-h|--help]}}
```

pw-jack

Run a JACK application with PipeWire.

More information: https://docs.pipewire.org/page_man_pw-jack_1.html.

- Run a command with its arguments, using PipeWire:

```
pw-jack {{command}} {{argument1 argument2 ...}}
```

- Run a command in verbose mode:

```
pw-jack -v {{command}}
```

- Connect to a specific remote PipeWire instance:

```
pw-jack -r {{remote_instance_name}} {{command}}
```

- Display help:

```
pw-jack -h
```

pw-link

Manage links between ports in PipeWire.

More information: https://gitlab.freedesktop.org/pipewire/pipewire/-/wikis/Virtual-Devices.

- List all audio output and input ports with their IDs:

```
pw-link {{[-oiI|--output --input --id]}}
```

- Create a link between an output and an input port:

```
pw-link {{output_port_name}} {{input_port_name}}
```

- Disconnect two ports:

```
pw-link {{[-d|--disconnect]}} {{output_port_name}} {{input_port_name}}
```

- List all links with their IDs:

```
pw-link {{[-lI|--links --id]}}
```

- Display help:

```
pw-link {{[-h|--help]}}
```

pw-loopback

Create loopback devices in PipeWire.

More information: https://docs.pipewire.org/page_man_pw-loopback_1.html.

- Create a loopback device with the default loopback behavior:

```
pw-loopback
```

- Create a loopback device that automatically connects to the speakers:

```
pw-loopback {{[-m|--channel-map]}} '{{[FL FR]}}' {{[-i|--
capture-props]}} '{{media.class=Audio/Sink}}'
```

- Create a loopback device that automatically connects to the microphone:

```
pw-loopback {{[-m|--channel-map]}} '{{[FL FR]}}' {{[-o|--
playback-props]}} '{{media.class=Audio/Source}}'
```

- Create a dummy loopback device that doesn't automatically connect to anything:

```
pw-loopback {{[-m|--channel-map]}} '{{[FL FR]}}' {{[-i|--
capture-props]}} '{{media.class=Audio/Sink}}' {{[-o|--
playback-props]}} '{{media.class=Audio/Source}}'
```

- Create a loopback device that automatically connects to the speakers and swaps the left and right channels between the sink and source:

```
pw-loopback {{[-i|--capture-props]}} '{{media.class=Audio/
Sink audio.position=[FL FR]}}' {{[-o|--playback-props]}}
'{{audio.position=[FR FL]}}'
```

- Create a loopback device that automatically connects to the microphone and swaps the left and right channels between the sink and source:

```
pw-loopback {{[-i|--capture-props]}} '{{audio.position=[FR
FL]}}' {{[-o|--playback-props]}} '{{media.class=Audio/Source
audio.position=[FL FR]}}'
```

pw-metadata

Monitor, set, and delete metadata on PipeWire objects.

See also: **`pipewire`**, **`pw-mon`**, **`pw-cli`**.

More information: https://docs.pipewire.org/page_man_pw-metadata_1.html.

- Show metadata in `default` name:

```
pw-metadata
```

- Show metadata with ID 0 in `settings`:

```
pw-metadata {{[-n|--name]}} {{settings}} {{0}}
```

- List all available metadata objects:

```
pw-metadata {{[-l|--list]}}
```

- Keep running and log the changes to the metadata:

```
pw-metadata {{[-m|--monitor]}}
```

- Delete all metadata:

```
pw-metadata {{[-d|--delete]}}
```

- Set `log.level` to 1 in `settings`:

```
pw-metadata {{[-n|--name]}} {{settings}} {{0}} {{log.level}} {{1}}
```

- Display help:

```
pw-metadata {{[-h|--help]}}
```

pw-mididump

Dump MIDI messages to **`stdout`**.

More information: https://docs.pipewire.org/page_man_pw-mididump_1.html.

- Listen for and dump all incoming MIDI events:

```
pw-mididump
```

- Dump MIDI events from a specific file:

```
pw-mididump {{path/to/file.mid}}
```

- Connect to a specific remote PipeWire instance:

```
pw-mididump {{[-r|--remote]}} {{remote_instance_name}}
```

- Display help:

```
pw-mididump {{[-h|--help]}}
```

pw-mon

Monitor objects on the PipeWire instance.

More information: https://docs.pipewire.org/page_man_pw-mon_1.html.

- Monitor the default PipeWire instance:

```
pw-mon
```

- Monitor a specific remote instance:

```
pw-mon {{[-r|--remote]}} {{remote_name}}
```

- Monitor the default instance specifying a color configuration:

```
pw-mon {{[-N|--color]}} {{never|always|auto}}
```

- Display help:

```
pw-mon {{[-h|--help]}}
```

pw-play

This command is an alias of **`pw-cat --playback`**.

- View documentation for the original command:

  ```
  tldr pw-cat
  ```

pw-profiler

Profile a local or remote instance.

More information: https://docs.pipewire.org/page_man_pw-profiler_1.html.

- Profile the default instance, logging to `profile.log` (`gnuplot` files and a HTML file for result visualizing are also generated):

```
pw-profiler
```

- Change the log output file:

```
pw-profiler {{[-o|--output]}} {{path/to/file.log}}
```

- Profile a remote instance:

```
pw-profiler {{[-r|--remote]}} {{remote_name}}
```

- Display help:

```
pw-profiler {{[-h|--help]}}
```

pw-record

This command is an alias of **`pw-cat --record`**.

- View documentation for the original command:

  ```
  tldr pw-cat
  ```

pw-reserve

Reserve a device for PipeWire.

More information: https://docs.pipewire.org/page_man_pw-reserve_1.html.

- Reserve a device (currently only supports audio devices):

```
pw-reserve {{[-n|--name]}} {{audioN}}
```

- Monitor a device instead of reserving it:

```
pw-reserve {{[-n|--name]}} {{audioN}} {{[-m|--monitor]}}
```

- Display help:

```
pw-reserve {{[-h|--help]}}
```

pw-top

View the PipeWire nodes and devices statistics in real-time.

See also: **`pipewire`**, **`pw-dump`**, **`pw-cli`**, **`pw-profiler`**.

More information: https://docs.pipewire.org/page_man_pw-top_1.html.

- Display an interactive view of PipeWire nodes and devices:

```
pw-top
```

- Monitor a remote instance:

```
pw-top {{[-r|--remote]}} {{remote_name}}
```

- Print information periodically instead of running in interactive mode:

```
pw-top {{[-b|--batch-mode]}}
```

- Print information periodically for a specific number of times:

```
pw-top {{[-b|--batch-mode]}} {{[-n|--iterations]}} {{3}}
```

pw-v4l2

Use v4l2 programs within PipeWire.

More information: https://docs.pipewire.org/page_man_pw-v4l2_1.html.

- Run a program:

```
pw-v4l2 {{program}}
```

- Display help:

```
pw-v4l2 -h
```

pwdx

Print working directory of a process.

More information: https://manned.org/pwdx.

- Print current working directory of a process:

```
pwdx {{process_id}}
```

pwn

Exploit Development Library designed for rapid prototyping.

More information: https://docs.pwntools.com/en/stable/commandline.html.

- Convert the given assembly code to `bytes`:

```
pwn asm "{{xor edi, edi}}"
```

- Create a cyclic pattern of the specific number of characters:

```
pwn cyclic {{number}}
```

- Encode the given data into the hexadecimal system:

```
pwn hex {{deafbeef}}
```

- Decode the given data from hexadecimal:

```
pwn unhex {{6c4f7645}}
```

- Print a x64 Linux shellcode for running a shell:

```
pwn shellcraft {{amd64.linux.sh}}
```

- Check the binary security settings for the given ELF file:

```
pwn checksec {{path/to/file}}
```

- Check for Pwntools updates:

```
pwn update
```

- Display version:

```
pwn version
```

pyrit

WPA/WPA2 cracking tool using computational power.

More information: https://manned.org/pyrit.

- Display system cracking speed:

```
pyrit benchmark
```

- List available cores:

```
pyrit list_cores
```

- Set [e]SSID:

```
pyrit -e "{{ESSID}}" create_essid
```

- [r]ead and analyze a specific packet capture file:

```
pyrit -r {{path/to/file.cap|path/to/file.pcap}} analyze
```

- Read and [i]mport passwords to the current database:

```
pyrit -i {{path/to/file}} {{import_unique_passwords|
unique_passwords|import_passwords}}
```

- Exp[o]rt passwords from database to a specific file:

```
pyrit -o {{path/to/file}} export_passwords
```

- Translate passwords with Pired Master Keys:

```
pyrit batch
```

- [r]ead the capture file and crack the password:

```
pyrit -r {{path/to/file}} attack_db
```

qjoypad

Translate input from gamepads or joysticks into keyboard strokes or mouse actions.

More information: https://qjoypad.sourceforge.net/doc/doc_index.html.

- Start QJoyPad:

```
qjoypad
```

- Start QJoyPad and look for devices in a specific directory:

```
qjoypad --device={{path/to/directory}}
```

- Start QJoyPad but don't show a system tray icon:

```
qjoypad --notray
```

- Start QJoyPad and force the window manager to use a system tray icon:

```
qjoypad --force-tray
```

- Force a running instance of QJoyPad to update its list of devices and layouts:

```
qjoypad --update
```

- Load the given layout in an already running instance of QJoyPad, or start QJoyPad using the given layout:

```
qjoypad "{{layout}}"
```

qm agent

This command is an alias of **`qm guest cmd`**.

- View documentation for the original command:

```
tldr qm guest cmd
```

qm cleanup

Clean up resources on QEMU/KVM Virtual Machine Manager like tap devices, VGPUs, etc.

Called after a VM shuts down, crashes, etc.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Clean up resources:

```
qm {{[cl|cleanup]}} {{vm_id}} {{clean-shutdown}} {{guest-requested}}
```

qm clone

Create a copy of virtual machine on QEMU/KVM Virtual Machine Manager.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Copy a virtual machine:

```
qm clone {{vm_id}} {{new_vm_id}}
```

- Copy a virtual machine using a specific name:

```
qm clone {{vm_id}} {{new_vm_id}} --name {{name}}
```

- Copy a virtual machine using a specific descriptionn:

```
qm clone {{vm_id}} {{new_vm_id}} --description
{{description}}
```

- Copy a virtual machine creating a full copy of all disks:

```
qm clone {{vm_id}} {{new_vm_id}} --full
```

- Copy a virtual machine using a specific format for file storage (requires `--full`):

```
qm clone {{vm_id}} {{new_vm_id}} --full --format {{qcow2|raw|
vmdk}}
```

- Copy a virtual machine then add it to a specific pool:

```
qm clone {{vm_id}} {{new_vm_id}} --pool {{pool_name}}
```

qm cloudinit dump

Generate cloudinit configuration files.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Generate a cloudinit file for a specific configuration type:

```
qm {{[clou|cloudinit]}} {{[d|dump]}} {{virtual_machine_id}}
{{meta|network|user}}
```

qm config

Display the virtual machine configuration with pending configuration changes applied.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Display the virtual machine configuration:

  ```
  qm {{[co|config]}} {{vm_id}}
  ```

- Display the current configuration values instead of pending values for the virtual machine:

  ```
  qm {{[co|config]}} --current {{true}} {{vm_id}}
  ```

- Fetch the configuration values from the given snapshot:

  ```
  qm {{[co|config]}} --snapshot {{snapshot_name}} {{vm_id}}
  ```

qm create

Create or restore a virtual machine on QEMU/KVM Virtual Machine Manager.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Create a virtual machine with access to 512MiB of memory and 1 CPU core:

```
qm {{[cr|create]}} {{100}}
```

- Give the virtual machine a name and automatically start the machine after creation:

```
qm {{[cr|create]}} {{100}} --name {{vm_name}} --start
```

- Give a virtual machine specified amount of memory and CPUs:

```
qm {{[cr|create]}} {{100}} --memory {{8192}} --cores {{4}}
```

- Specify the type of operating system on the machine:

```
qm {{[cr|create]}} {{100}} --ostype {{win10}}
```

- Replace an existing machine (requires archiving it):

```
qm {{[cr|create]}} {{100}} --archive {{path/to/
backup_file.tar}} --force 1
```

- Specify a script that is executed automatically depending on the state of the virtual machine:

```
qm {{[cr|create]}} {{100}} --hookscript {{path/to/script.pl}}
```

- Specify the install media:

```
qm {{[cr|create]}} {{100}} --cdrom {{local:iso/install-
media.iso}}
```

- Create a VM that bridges itself to the host network:

```
qm {{[cr|create]}} {{100}} --net{{0}} virtio,bridge=vmbr{{0}}
```

qm delsnapshot

Delete virtual machine snapshots.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Delete a snapshot:

```
qm {{[del|delsnapshot]}} {{vm_id}} {{snapshot_name}}
```

- Delete a snapshot from a configuration file (even if removing the disk snapshot fails):

```
qm {{[del|delsnapshot]}} {{vm_id}} {{snapshot_name}} --force 1
```

qm destroy

Destroy a virtual machine in QEMU/KVM Virtual Machine Manager.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Destroy a specific virtual machine:

```
qm {{[des|destroy]}} {{vm_id}}
```

- Destroy all disks that are not explicitly referenced in a specific virtual machine's configuration:

```
qm {{[des|destroy]}} {{vm_id}} --destroy-unreferenced-disks
```

- Destroy a virtual machine and remove from all locations (inventory, backup jobs, high availability managers, etc.):

```
qm {{[des|destroy]}} {{vm_id}} --purge
```

- Destroy a specific virtual machine ignoring locks and forcing destroy:

```
sudo qm {{[des|destroy]}} {{vm_id}} --skiplock
```

qm disk import

Import a disk image to a virtual machine as an unused disk.

The supported image formats for **qemu-img**, such as raw, qcow2, qed, vdi, vmdk, and vhd must be used.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Import a VMDK/qcow2/raw disk image using a specific storage name:

```
qm {{[di|disk]}} {{[i|import]}} {{vm_id}} {{path/to/disk}} {{storage_name}} --format {{qcow2|raw|vmdk}}
```

qm disk move

Move a virtual disk from one storage to another within the same Proxmox cluster.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Move a virtual disk:

```
qm {{[di|disk]}} {{[m|move]}} {{vm_id}} {{destination}}
{{index}}
```

- Delete the previous copy of the virtual disk:

```
qm {{[di|disk]}} {{[m|move]}} --delete {{vm_id}}
{{destination}} {{index}}
```

qm disk rescan

Rescan all storages and update disk sizes and unused disk images of virtual machines.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Rescan all storages and update disk sizes and unused disk images:

```
qm {{[di|disk]}} {{[resc|rescan]}}
```

- Perform a dry-run of a rescan and do not write any changes to configurations:

```
qm {{[di|disk]}} {{[resc|rescan]}} --dryrun
```

- Specify a virtual machine by its ID:

```
qm {{[di|disk]}} {{[resc|rescan]}} --vmid {{100}}
```

qm disk resize

Resize a virtual machine disk in the Proxmox Virtual Environment (PVE).

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Add n gigabytes to a virtual disk:

```
qm {{[di|disk]}} {{[resi|resize]}} {{vm_id}} {{disk_name}} +
{{n}}G
```

qm disk

Manage disk images.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Add n gigabytes to a virtual disk:

```
qm {{[di|disk]}} {{[resi|resize]}} {{vm_id}} {{disk_name}} +
{{n}}G
```

- Move a virtual disk:

```
qm {{[di|disk]}} {{[m|move]}} {{vm_id}} {{destination}}
{{index}}
```

- Delete the previous copy of the virtual disk:

```
qm {{[di|disk]}} {{[m|move]}} --delete {{vm_id}}
{{destination}} {{index}}
```

- Import a VMDK/qcow2/raw disk image using a specific storage name:

```
qm {{[di|disk]}} {{[i|import]}} {{vm_id}} {{path/to/disk}}
{{storage_name}} --format {{qcow2|raw|vmdk}}
```

qm guest cmd

Execute QEMU Guest Agent commands.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Execute a specific QEMU Guest Agent command:

```
qm {{[g|guest]}} {{[c|cmd]}} {{virtual_machine_id}}
{{fsfreeze-freeze|fsfreeze-status|fsfreeze-thaw|fstrim|get-
fsinfo|...}}
```

qm guest exec-status

Print the status of a pid started by the guest-agent on QEMU/KVM Virtual Machine Manager.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Print the status of a specific PID:

```
qm {{[g|guest]}} {{[exec-s|exec-status]}} {{vm_id}} {{pid}}
```

qm guest exec

Execute a specific command via a guest agent.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Execute a specific command via a guest agent:

```
qm {{[g|guest]}} exec {{vm_id}} {{command}} {{argument1 argument2 ...}}
```

- Execute a specific command via a guest agent asynchronously:

```
qm {{[g|guest]}} exec {{vm_id}} {{argument1 argument2 ...}} --synchronous 0
```

- Execute a specific command via a guest agent with a specified timeout of 10 seconds:

```
qm {{[g|guest]}} exec {{vm_id}} {{argument1 argument2...}} --timeout {{10}}
```

- Execute a specific command via a guest agent and forward input from `stdin` until EOF to the guest agent:

```
qm {{[g|guest]}} exec {{vm_id}} {{argument1 argument2 ...}} --pass-stdin 1
```

qm guest passwd

Set the password for a user on QEMU/KVM Virtual Machine Manager.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Set a password for a specific user in a virtual machine interactively:

```
qm {{[g|guest]}} {{[p|passwd]}} {{vm_id}} {{username}}
```

- Set an already hashed password for a specific user in a virtual machine interactively:

```
qm {{[g|guest]}} {{[p|passwd]}} {{vm_id}} {{username}} --crypted 1
```

qm guest

Manage a VM guest agent.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Print the status of a specific PID:

```
qm {{[g|guest]}} {{[exec-s|exec-status]}} {{vm_id}} {{pid}}
```

- Set a password for a specific user in a virtual machine interactively:

```
qm {{[g|guest]}} {{[p|passwd]}} {{vm_id}} {{username}}
```

- Execute a specific QEMU Guest Agent command:

```
qm {{[g|guest]}} {{[c|cmd]}} {{virtual_machine_id}}
{{fsfreeze-freeze|fsfreeze-status|fsfreeze-thaw|fstrim|get-
fsinfo|...}}
```

- Execute a specific command via a guest agent:

```
qm {{[g|guest]}} exec {{vm_id}} {{command}} {{argument1
argument2 ...}}
```

qm help

Display help for a command.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Display help for a specific command:

```
qm {{[h|help]}} {{command}}
```

- Display help for a specific command with detailed information:

```
qm {{[h|help]}} {{command}} --verbose {{true|false}}
```

qm importdisk

This command is an alias of **`qm disk import`**.

- View documentation for the original command:

```
tldr qm disk import
```

qm list

List all virtual machines.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- List all virtual machines:

```
qm list
```

- List all virtual machines with a full status about the ones which are currently running:

```
qm list --full 1
```

qm listsnapshot

List snapshots of virtual machines.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- List all snapshots of a specific virtual machine:

```
qm {{[lists|listsnapshot]}} {{vm_id}}
```

qm migrate

Migrate a virtual machine.

Used to create a new migration task.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Migrate a specific virtual machine:

```
qm {{[mi|migrate]}} {{vm_id}} {{target}}
```

- Override the current I/O bandwidth limit with 10 KiB/s:

```
qm {{[mi|migrate]}} {{vm_id}} {{target}} --bwlimit 10
```

- Allow migration of virtual machines using local devices (root only):

```
qm {{[mi|migrate]}} {{vm_id}} {{target}} --force true
```

- Use online/live migration if a virtual machine is running:

```
qm {{[mi|migrate]}} {{vm_id}} {{target}} --online true
```

- Enable live storage migration for local disks:

```
qm {{[mi|migrate]}} {{vm_id}} {{target}} --with-local-disks
true
```

qm monitor

Enter the QEMU Monitor interface.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Enter the QEMU Monitor interface of a specific virtual machine:

```
qm {{[mo|monitor]}} {{vm_id}}
```

qm move-disk

This command is an alias of **`qm disk move`**.

- View documentation for the original command:

```
tldr qm disk move
```

qm move_disk

This command is an alias of **`qm disk move`**.

- View documentation for the original command:

```
tldr qm disk move
```

qm mtunnel

Used by **`qmigrate`**.

It should not be invoked manually.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Command used by `qmigrate` during data migration from a VM to another host:

  ```
  qm {{[mt|mtunnel]}}
  ```

qm nbdstop

Stop embedded nbd server.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Stop embedded nbd server:

```
qm {{[n|nbdstop]}} {{VM_ID}}
```

qm pending

Get the virtual machine configuration with both current and pending values.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Get the virtual machine configuration of a specific virtual machine:

```
qm {{[p|pending]}} {{vm_id}}
```

qm reboot

Reboot a virtual machine by shutting it down, and starting it again after applying pending changes.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Reboot a virtual machine:

```
qm {{[reb|reboot]}} {{vm_id}}
```

- Reboot a virtual machine after wait for at most 10 seconds:

```
qm {{[reb|reboot]}} --timeout {{10}} {{vm_id}}
```

qm rescan

This command is an alias of **`qm disk rescan`**.

- View documentation for the original command:

```
tldr qm disk rescan
```

qm reset

Reset a virtual machine on QEMU/KVM Virtual Machine Manager.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Reset a virtual machine:

```
qm reset {{vm_id}}
```

- Reset a virtual machine and skip lock (only root can use this option):

```
qm reset --skiplock {{true}} {{vm_id}}
```

qm resize

This command is an alias of **`qm disk resize`**.

- View documentation for the original command:

```
tldr qm disk resize
```

qm resume

Resume a virtual machine.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Resume a specific virtual machine:

```
qm {{[resu|resume]}} {{vm_id}}
```

- Resume a specific virtual machine ignoring locks (requires root):

```
sudo qm {{[resu|resume]}} {{vm_id}} --skiplock true
```

qm rollback

Rollback the VM state to a specified snapshot.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Rollback the state of a specific VM to a specified snapshot:

```
qm {{[ro|rollback]}} {{vm_id}} {{snap_name}}
```

qm sendkey

Send QEMU monitor encoding key event to a virtual machine.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Send the specified key event to a specific virtual machine:

```
qm {{[sen|sendkey]}} {{vm_id}} {{key}}
```

- Allow root user to send key event and ignore locks:

```
qm {{[sen|sendkey]}} --skiplock {{true}} {{vm_id}} {{key}}
```

qm set

Set virtual machine options.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Set a name for a VM in the GUI:

```
qm set {{100}} --name {{vm_name}}
```

- Set a VM to autostart on boot:

```
qm set {{100}} --autostart {{0|1}}
```

- Set the allotted core count of a VM:

```
qm set {{100}} --cores {{4}}
```

- Set the allotted amount of memory:

```
qm set {{100}} --memory {{8192}}
```

- Give a VM a network device and bridge it to the host network:

```
qm set {{100}} --net{{0}} {{virtio|e1000|rtl8139|
vmxnet3}},bridge=vmbr{{0}}
```

- Delete a device:

```
qm set {{100}} --delete {{device_name0,device_name1,...}}
```

- Passthrough a GPU device to the guest:

```
qm set {{100}} --hostpci{{0}} {{0000:00:02}},x-vga=1 --bios
ovmf
```

qm showcmd

Show command-line which is used to start the VM (debug info).

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Show command-line for a specific virtual machine:

```
qm {{[sho|showcmd]}} {{vm_id}}
```

- Put each option on a new line to enhance human readability:

```
qm {{[sho|showcmd]}} --pretty {{true}} {{vm_id}}
```

- Fetch configuration values from a specific snapshot:

```
qm {{[sho|showcmd]}} --snapshot {{string}} {{vm_id}}
```

qm shutdown

Shutdown a virtual machine on QEMU/KVM Virtual Machine Manager.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Shutdown a virtual machine:

```
qm {{[shu|shutdown]}} {{VM_ID}}
```

- Shutdown a virtual machine after wait for at most 10 seconds:

```
qm {{[shu|shutdown]}} --timeout {{10}} {{VM_ID}}
```

- Shutdown a virtual machine and do not deactivate storage volumes:

```
qm {{[shu|shutdown]}} --keepActive {{true}} {{VM_ID}}
```

- Shutdown a virtual machine and skip lock (only root can use this option):

```
qm {{[shu|shutdown]}} --skiplock {{true}} {{VM_ID}}
```

- Stop and shutdown a virtual machine:

```
qm {{[shu|shutdown]}} --forceStop {{true}} {{VM_ID}}
```

qm snapshot

Create virtual machine snapshots.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Create a snapshot of a specific virtual machine (name must start with a letter):

```
qm {{[sn|snapshot]}} {{vm_id}} {{snapshot_name}}
```

- Create a snapshot with a specific description:

```
qm {{[sn|snapshot]}} {{vm_id}} {{snapshot_name}} --
description {{description}}
```

- Create a snapshot including the vmstate:

```
qm {{[sn|snapshot]}} {{vm_id}} {{snapshot_name}} --
description {{description}} --vmstate 1
```

- List snapshots of a VM:

```
qm {{[lists|listsnapshot]}} {{100}}
```

- Rollback the state of a specific VM to a specified snapshot:

```
qm {{[ro|rollback]}} {{vm_id}} {{snap_name}}
```

qm start

Start a virtual machine on QEMU/KVM Virtual Machine Manager.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Start a specific virtual machine:

```
qm start {{100}}
```

- Specify the QEMU machine type (i.e. the CPU to emulate):

```
qm start {{100}} --machine {{q35}}
```

- Start a specific virtual machine with a timeout in 60 seconds:

```
qm start {{100}} --timeout {{60}}
```

qm status

Show virtual machine status.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Display the status of a specific virtual machine:

```
qm {{[stat|status]}} {{vm_id}}
```

- Display detailed status of a specific virtual machine:

```
qm {{[stat|status]}} --verbose {{true}} {{vm_id}}
```

qm stop

Stop a virtual machine.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Stop a virtual machine immediately:

```
qm stop {{VM_ID}}
```

- Stop a virtual machine and wait for at most 10 seconds:

```
qm stop --timeout {{10}} {{VM_ID}}
```

- Stop a virtual machine and skip lock (only root can use this option):

```
qm stop --skiplock {{true}} {{VM_ID}}
```

- Stop a virtual machine and don't deactivate storage volumes:

```
qm stop --keepActive {{true}} {{VM_ID}}
```

qm suspend

Suspends a virtual machine (VM) in the Proxmox Virtual Environment (PVE).

Use **`--skiplock`** and **`--skiplockstorage`** flags with caution, as they may lead to data corruption in certain situations.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Suspend a virtual machine by ID:

```
qm {{[su|suspend]}} {{vm_id}} {{integer}}
```

- Skip the lock check when suspending the VM:

```
qm {{[su|suspend]}} {{vm_id}} {{integer}} --skiplock
```

- Skip the lock check for storage when suspending the VM:

```
qm {{[su|suspend]}} {{vm_id}} {{integer}} --skiplockstorage
```

qm template

Create a Proxmox VM template.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Create a template out of a specific virtual machine:

```
qm {{[tem|template]}} {{vm_id}}
```

qm terminal

Attach to the serial console of a VM.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Attach to a terminal:

```
qm {{[ter|terminal]}} {{100}}
```

- Detach from a terminal:

```
<Ctrl o>
```

qm unlink

This command is an alias of **`qm disk unlink`**.

- View documentation for the original command:

```
tldr qm disk unlink
```

qm unlock

Unlock a virtual machine in QEMU/KVM Virtual Machine Manager.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Unlock a specific virtual machine:

```
qm {{[u|unlock]}} {{vm_id}}
```

qm vncproxy

Proxy Virtual Machine VNC (Virtual network computing) traffic to **`stdin`** or **`stdout`**.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Proxy a specific virtual machine:

```
qm {{[v|vncproxy]}} {{vm_id}}
```

qm wait

Wait until the virtual machine is stopped.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- Wait until the virtual machine is stopped:

```
qm {{[w|wait]}} {{vm_id}}
```

- Wait until the virtual machine is stopped with a 10 second timeout:

```
qm {{[w|wait]}} --timeout {{10}} {{vm_id}}
```

- Send a shutdown request, then wait until the virtual machine is stopped with a 10 second timeout:

```
qm {{[shu|shutdown]}} {{vm_id}} && qm {{[w|wait]}} --timeout {{10}} {{vm_id}}
```

qm

QEMU/KVM Virtual Machine Manager.

Some subcommands such as **`list`**, **`start`**, **`stop`**, **`clone`**, etc. have their own usage documentation.

More information: https://pve.proxmox.com/pve-docs/qm.1.html.

- List all virtual machines:

```
qm list
```

- Using an ISO file uploaded on the local storage, create a virtual machine with a 4 GB SCSI disk on the `local-lvm` storage and an ID of 100:

```
qm {{[cr|create]}} {{100}} --scsi0 {{local-lvm:4}} --net0
{{e1000}} --cdrom {{local:iso/proxmox-mailgateway_2.1.iso}}
```

- Show the configuration of a virtual machine, specifying its ID:

```
qm {{[co|config]}} {{100}}
```

- Start a specific virtual machine:

```
qm start {{100}}
```

- Send a shutdown request, then wait until the virtual machine is stopped:

```
qm {{[shu|shutdown]}} {{100}} && qm {{[w|wait]}} {{100}}
```

- Destroy a virtual machine and remove all related resources:

```
qm {{[des|destroy]}} {{100}} --purge
```

qmrestore

Restore QemuServer **vzdump** backups.

More information: https://pve.proxmox.com/pve-docs/qmrestore.1.html.

- Restore virtual machine from given backup file on the original storage:

```
qmrestore {{path/to/vzdump-qemu-100.vma.lzo}} {{100}}
```

- Overwrite existing virtual machine from a given backup file on the original storage:

```
qmrestore {{path/to/vzdump-qemu-100.vma.lzo}} {{100}} --force true
```

- Restore the virtual machine from a given backup file on specific storage:

```
qmrestore {{path/to/vzdump-qemu-100.vma.lzo}} {{100}} --storage {{local}}
```

- Start virtual machine immediately from the backup while restoring in the background (only on Proxmox Backup Server):

```
qmrestore {{path/to/vzdump-qemu-100.vma.lzo}} {{100}} --live-restore true
```

qrcp

A file transfer tool.

More information: https://github.com/claudiodangelis/qrcp.

- Send a file or directories:

```
qrcp send {{path/to/file_or_directory1 path/to/
file_directory2 ...}}
```

- Receive files:

```
qrcp receive
```

- Compress content before transferring:

```
qrcp send --zip {{path/to/file_or_directory}}
```

- Use a specific port:

```
qrcp {{send|receive}} {{[-p|--port]}} {{port_number}}
```

- Use a specific network interface:

```
qrcp {{send|receive}} {{[-i|--interface]}} {{interface}}
```

- Keep the server alive:

```
qrcp {{send|receive}} --keep-alive
```

qsub

Submits a script to the queue management system TORQUE.

More information: https://manned.org/qsub.1.

- Submit a script with default settings (depends on TORQUE settings):

```
qsub {{script.sh}}
```

- Submit a script with a specified wallclock runtime limit of 1 hour, 2 minutes and 3 seconds:

```
qsub -l walltime={{1}}:{{2}}:{{3}} {{script.sh}}
```

- Submit a script that is executed on 2 nodes using 4 cores per node:

```
qsub -l nodes={{2}}:ppn={{4}} {{script.sh}}
```

- Submit a script to a specific queue. Note that different queues can have different maximum and minimum runtime limits:

```
qsub -q {{queue_name}} {{script.sh}}
```

qtchooser

A wrapper used to select between Qt development binary versions.

More information: https://manned.org/qtchooser.

- List available Qt versions from the configuration files:

```
qtchooser --list-versions
```

- Print environment information:

```
qtchooser --print-env
```

- Run the specified tool using the specified Qt version:

```
qtchooser --run-tool={{tool}} --qt={{version_name}}
```

- Add a Qt version entry to be able to choose from:

```
qtchooser --install {{version_name}} {{path/to/qmake}}
```

- Display help:

```
qtchooser --help
```

qtile

A full-featured, hackable tiling window manager written and configured in Python.

More information: https://docs.qtile.org/en/latest/manual/commands/shell/index.html.

- Start the window manager, if it is not running already (should ideally be run from `.xsession` or similar):

```
qtile start
```

- Check the configuration file for any compilation errors (default location is `~/.config/qtile/config.py`):

```
qtile check
```

- Show current resource usage information:

```
qtile top --force
```

- Open the program `xterm` as a floating window on the group named `test-group`:

```
qtile run-cmd --group {{test-group}} --float {{xterm}}
```

- Restart the window manager:

```
qtile cmd-obj --object cmd --function restart
```

quickemu

Build and manage highly optimised desktop virtual machines quickly.

See also: **`quickget`** for preparing VM configurations.

More information: https://github.com/quickemu-project/quickemu.

- Create and run a virtual machine from a configuration file:

```
quickemu --vm {{path/to/file.conf}}
```

- Do not commit any changes to disk/snapshot but write any changes to temporary files:

```
quickemu --status-quo --vm {{path/to/file.conf}}
```

- Start the virtual machine in full-screen mode (`<Ctrl Alt f>` to exit) and select the display backend (`sdl` by default):

```
quickemu --fullscreen --display {{sdl|gtk|spice|spice-app|
none}} --vm {{path/to/file.conf}}
```

- Select a virtual audio device to emulate and create a desktop shortcut:

```
quickemu --sound-card {{intel-hda|ac97|es1370|sb16|none}} --
shortcut --vm {{path/to/file.conf}}
```

- Create a snapshot:

```
quickemu --snapshot create {{tag}} --vm {{path/to/file.conf}}
```

- Restore a snapshot:

```
quickemu --snapshot apply {{tag}} --vm {{path/to/file.conf}}
```

- Delete a snapshot:

```
quickemu --snapshot delete {{tag}} --vm {{path/to/file.conf}}
```

quickget

Download and prepare materials for building a Quickemu virtual machine.

Note: The parameter "edition" is always optional.

See also: **`quickemu`**.

More information: https://github.com/quickemu-project/quickemu.

- Display the list of all supported guest operating systems, versions and variants:

```
quickget list
```

- Download and create the virtual machine configuration for building a Quickemu virtual machine for an OS:

```
quickget {{os}} {{release}} {{edition}}
```

- Download configuration for a Windows 11 VM with VirtIO drivers for Windows:

```
quickget windows 11
```

- Download a macOS recovery image and creates a virtual machine configuration:

```
quickget macos {{mojave|catalina|big-sur|monterey|ventura|sonoma}}
```

- Show an ISO URL for an operating system:

```
quickget --url fedora {{release}} {{edition}}
```

- Test if an ISO file is available for an operating system:

```
quickget --check nixos {{release}} {{edition}}
```

- Download an image without building any VM configuration:

```
quickget --download {{os}} {{release}} {{edition}}
```

- Create a VM configuration for an OS image:

```
quickget --create-config {{os}} {{path/to/iso}}
```

quotacheck

Scan a filesystem for disk usage; create, check and repair quota files.

It is best to run quota check with quotas turned off to prevent damage or loss to quota files.

More information: https://manned.org/quotacheck.

- Check quotas on all mounted non-NFS filesystems:

```
sudo quotacheck --all
```

- Force check even if quotas are enabled (this can cause damage or loss to quota files):

```
sudo quotacheck --force {{mountpoint}}
```

- Check quotas on a given filesystem in debug mode:

```
sudo quotacheck --debug {{mountpoint}}
```

- Check quotas on a given filesystem, displaying the progress:

```
sudo quotacheck --verbose {{mountpoint}}
```

- Check user quotas:

```
sudo quotacheck --user {{user}} {{mountpoint}}
```

- Check group quotas:

```
sudo quotacheck --group {{group}} {{mountpoint}}
```

radeontop

Show utilization of AMD GPUs.

May require root privileges depending on your system.

More information: https://github.com/clbr/radeontop.

- Show the utilization of the default AMD GPU:

```
radeontop
```

- Enable colored output:

```
radeontop --color
```

- Select a specific GPU (the bus number is the first number in the output of `lspci`):

```
radeontop --bus {{bus_number}}
```

- Specify the display refresh rate (higher means more GPU overhead):

```
radeontop --ticks {{samples_per_second}}
```

rankmirrors

Rank a list of Pacman mirrors by connection and opening speed.

Writes the new mirrorlist to **`stdout`**.

More information: https://manned.org/rankmirrors.

- Rank a mirror list:

```
rankmirrors {{/etc/pacman.d/mirrorlist}}
```

- Output only a given number of the top ranking servers:

```
rankmirrors -n {{number}} {{/etc/pacman.d/mirrorlist}}
```

- Be verbose when generating the mirrorlist:

```
rankmirrors {{[-v|--verbose]}} {{/etc/pacman.d/mirrorlist}}
```

- Test only a specific URL:

```
rankmirrors {{[-u|--url]}} {{url}}
```

- Output only the response times instead of a full mirrorlist:

```
rankmirrors {{[-t|--times]}} {{/etc/pacman.d/mirrorlist}}
```

raspi-config

An **`ncurses`** terminal GUI to config a Raspberry Pi.

More information: https://www.raspberrypi.com/documentation/computers/configuration.html.

- Start `raspi-config`:

```
sudo raspi-config
```

raspinfo

Display Raspberry Pi system information.

More information: https://github.com/raspberrypi/utils/tree/master/raspinfo.

- Display system information:

```
raspinfo
```

raw

Bind a Unix raw character device.

More information: https://manned.org/raw.8.

- Bind a raw character device to a block device:

```
raw /dev/raw/raw{{1}} {{/dev/block_device}}
```

- Query an existing binding instead of setting a new one:

```
raw /dev/raw/raw{{1}}
```

- Query all bound raw devices:

```
raw {{[-qa|--query --all]}}
```

rc-service

Locate and run OpenRC services with arguments.

See also: **`openrc`**.

More information: https://manned.org/rc-service.

- Show a service's status:

```
rc-service {{service_name}} status
```

- Start a service:

```
sudo rc-service {{service_name}} start
```

- Stop a service:

```
sudo rc-service {{service_name}} stop
```

- Restart a service:

```
sudo rc-service {{service_name}} restart
```

- Simulate running a service's custom command:

```
sudo rc-service {{[-Z|--dry-run]}} {{service_name}}
{{command_name}}
```

- Actually run a service's custom command:

```
sudo rc-service {{service_name}} {{command_name}}
```

- Resolve the location of a service definition on disk:

```
sudo rc-service {{[-r|--resolve]}} {{service_name}}
```

rc-status

Show status info about runlevels.

See also: **`openrc`**.

More information: https://manned.org/rc-status.

- Show a summary of services and their status:

```
rc-status
```

- Include services in all runlevels in the summary:

```
rc-status {{[-a|--all]}}
```

- List services that have crashed:

```
rc-status {{[-c|--crashed]}}
```

- List manually started services:

```
rc-status {{[-m|--manual]}}
```

- List supervised services:

```
rc-status {{[-S|--supervised]}}
```

- Display the current runlevel:

```
rc-status {{[-r|--runlevel]}}
```

- List all runlevels:

```
rc-status {{[-l|--list]}}
```

rc-update

Add and remove OpenRC services to and from runlevels.

See also: **`openrc`**.

More information: https://manned.org/rc-update.

- List enabled services and the runlevels they are added to:

```
rc-update
```

- List all services:

```
rc-update {{[-v|--verbose]}}
```

- Add a service to a runlevel:

```
sudo rc-update add {{service_name}} {{runlevel}}
```

- Delete a service from a runlevel:

```
sudo rc-update {{[del|delete]}} {{service_name}} {{runlevel}}
```

- Delete a service from all runlevels:

```
sudo rc-update {{[-a|--all]}} {{[del|delete]}}
{{service_name}}
```

rcp

Copy files between local and remote systems.

It mimics the behavior of the **cp** command but operates across different machines.

More information: https://www.gnu.org/software/inetutils/manual/inetutils.html#rcp-invocation.

- Copy a file to a remote host:

```
rcp {{path/to/local_file}} {{username}}@{{remote_host}}:/{{path/to/destination}}/
```

- Copy a directory recursively:

```
rcp {{[-r|--recursive]}} {{path/to/local_directory}} {{username}}@{{remote_host}}:/{{path/to/destination}}/
```

- Preserve the file attributes:

```
rcp {{[-p|--preserve]}} {{path/to/local_file}} {{username}}@{{remote_host}}:/{{path/to/destination}}/
```

- Force copy without a confirmation:

```
rcp {{[-f|--from]}} {{path/to/local_file}} {{username}}@{{remote_host}}:/{{path/to/destination}}/
```

rdesktop

Remote Desktop Protocol client.

It can be used to connect the remote computer using the RDP protocol.

More information: https://manned.org/rdesktop.

- Connect to a remote computer (default port is 3389):

```
rdesktop -u {{username}} -p {{password}} {{host:port}}
```

- Simple Examples:

```
rdesktop -u Administrator -p passwd123 192.168.1.111:3389
```

- Connect to a remote computer with full screen (press `<Ctrl Alt Enter>` to exist):

```
rdesktop -u {{username}} -p {{password}} -f {{host:port}}
```

- Use a custom resolution (use the letter `x` between the numbers):

```
rdesktop -u {{username}} -p {{password}} -g {{1366}}x{{768}} {{host:port}}
```

- Connect to a remote computer using domain user:

```
rdesktop -u {{username}} -p {{password}} -d {{domainname}} {{host:port}}
```

- Use the 16-bit color (speed up):

```
rdesktop -u {{username}} -p {{password}} -a 16 {{host:port}}
```

readcd

Read or write Compact Disc media data.

More information: https://manned.org/readcd.

- Read a cd and copy it to a file:

```
readcd dev={{/dev/srX}} f={{path/to/file.iso}}
```

readelf

Display information about ELF files.

More information: https://manned.org/readelf.1.

- Display all information about the ELF file:

```
readelf -all {{path/to/binary}}
```

- Display all the headers present in the ELF file:

```
readelf --headers {{path/to/binary}}
```

- Display the entries in symbol table section of the ELF file, if it has one:

```
readelf --symbols {{path/to/binary}}
```

- Display ELF header information:

```
readelf --file-header {{path/to/binary}}
```

- Display ELF section header information:

```
readelf --section-headers {{path/to/binary}}
```

readpe

Display information about PE files.

More information: https://manned.org/readpe.

- Display all information about a PE file:

```
readpe {{path/to/executable}}
```

- Display all the headers present in a PE file:

```
readpe --all-headers {{path/to/executable}}
```

- Display all the sections present in a PE file:

```
readpe --all-sections {{path/to/executable}}
```

- Display a specific header from a PE file:

```
readpe --header {{dos|coff|optional}} {{path/to/executable}}
```

- List all imported functions:

```
readpe --imports {{path/to/executable}}
```

- List all exported functions:

```
readpe --exports {{path/to/executable}}
```

reboot

Reboot the system.

More information: https://manned.org/reboot.8.

- Reboot the system:

  `reboot`

- Power off the system (same as `poweroff`):

  `reboot {{[-p|--poweroff]}}`

- Halt (terminates all processes and shuts down the CPU) the system (same as `halt`):

  `reboot --halt`

- Reboot immediately without contacting the system manager:

  `reboot {{[-f|--force]}}`

- Write the wtmp shutdown entry without rebooting the system:

  `reboot {{[-w|--wtmp-only]}}`

reflector

Arch script to fetch and sort mirrorlists.

More information: https://manned.org/reflector.

- Get all mirrors, sort for download speed and save them:

```
sudo reflector --sort {{rate}} --save {{/etc/pacman.d/mirrorlist}}
```

- Only get German HTTPS mirrors:

```
reflector {{[-c|--country]}} {{Germany}} {{[-p|--protocol]}} {{https}}
```

- Only get the 10 recently sync'd mirrors:

```
reflector {{[-l|--latest]}} {{10}}
```

- Use a configuration file to fetch mirrors:

```
sudo reflector @{{/etc/xdg/reflector/reflector.conf}}
```

- Display help:

```
reflector {{[-h|--help]}}
```

register_new_matrix_user

Register new users in a home server when registration has been disabled.

More information: https://manned.org/register_new_matrix_user.

- Create a user interactively:

```
register_new_matrix_user --config {{path/to/homeserver.yaml}}
```

- Create an admin user interactively:

```
register_new_matrix_user --config {{path/to/homeserver.yaml}}
--admin
```

- Create an admin user non-interactively (not recommended):

```
register_new_matrix_user --config {{path/to/homeserver.yaml}}
--user {{username}} --password {{password}} --admin
```

remote-viewer

View desktops remotely.

More information: https://manned.org/remote-viewer.

- View a remote using a `virt-viewer` file:

```
remote-viewer {{path/to/virt_viewer_file}}
```

removepkg

Remove a specified Slackware package.

More information: http://www.slackware.com/config/packages.php.

- Remove a package:

```
sudo removepkg {{package_name}}
```

- Generate a report of a simulated removal to `stdout`:

```
removepkg -warn {{package_name}}
```

- Reconstruct the package subtree in `/tmp/preserved_packages/package_name` and remove the package:

```
sudo removepkg -preserve {{package_name}}
```

- Copy package under `/tmp/preserved_packages/package_name` without removing:

```
removepkg -copy {{package_name}}
```

- Save temporary files created by `removepkg` for debugging:

```
sudo removepkg -keep {{package_name}}
```

rename

rename can refer to multiple commands with the same name.

- View documentation for the Perl version:

  `tldr {{[-p|--platform]}} common rename`

- View documentation for the `util-linux` version:

  `tldr rename.util`

rename

Rename multiple files.

WARNING: This command will overwrite files without prompting unless the dry-run option is used.

Note: This page refers to the command from the **`util-linux`** package.

More information: https://manned.org/rename.

- Rename files using simple substitutions (substitute `foo` with `bar` wherever found):

```
rename {{foo}} {{bar}} {{*}}
```

- Dry-run - display which renames would occur without performing them:

```
rename {{[-vn|--verbose --no-act]}} {{foo}} {{bar}} {{*}}
```

- Do not overwrite existing files:

```
rename {{[-o|--no-overwrite]}} {{foo}} {{bar}} {{*}}
```

- Change file extensions:

```
rename {{.ext}} {{.bak}} {{*.ext}}
```

- Prepend "foo" to all filenames in the current directory:

```
rename '' '{{foo}}' {{*}}
```

- Rename a group of increasingly numbered files zero-padding the numbers up to 3 digits:

```
rename {{foo}} {{foo00}} {{foo?}} && rename {{foo}} {{foo0}} {{foo??}}
```

renice

Alter the scheduling priority/niceness of running processes.

Niceness values range from -20 (most favorable to the process) to 19 (least favorable to the process).

See also: **nice**.

More information: https://manned.org/renice.

- Set the absolute priority of a running process:

```
renice --priority {{3}} {{[-p|--pid]}} {{pid}}
```

- Increase the priority of a running process:

```
sudo renice --relative {{-4}} {{[-p|--pid]}} {{pid}}
```

- Decrease the priority of all processes owned by a user:

```
renice --relative {{4}} {{[-u|--user]}} {{uid|user}}
```

- Set the priority of all processes that belong to a process group:

```
sudo renice {{-5}} {{[-g|--pgrp]}} {{process_group}}
```

repo-add

Package database maintenance utility which enables installation of said package via Pacman.

See also: **`repo-remove`**.

More information: https://manned.org/repo-add.

- Create an empty repository:

```
repo-add {{path/to/database.db.tar.gz}}
```

- Add packages to the repository:

```
repo-add {{path/to/database.db.tar.gz}}
{{package1.pkg.tar.zst package2.pkg.tar.zst ...}}
```

- Add all package binaries in the current directory and remove any outdated package files:

```
repo-add {{[-R|--remove]}} {{path/to/database.db.tar.gz}}
{{*.pkg.tar.zst}}
```

- Add all package binaries in the current directory in silent mode except for warning and error messages:

```
repo-add {{[-q|--quiet]}} {{path/to/database.db.tar.gz}}
{{*.pkg.tar.zst}}
```

- Add all package binaries in the current directory without showing color:

```
repo-add --nocolor {{path/to/database.db.tar.gz}}
{{*.pkg.tar.zst}}
```

repo-remove

Package database maintenance utility which removes packages from a local repository.

See also: **repo-add**.

More information: https://manned.org/repo-add.

- Remove a package from a local repository:

```
repo-remove {{path/to/database.db.tar.gz}} {{package}}
```

reportbug

Bug report tool of Debian distribution.

More information: https://manned.org/reportbug.

- Generate a bug report about a specific package, then send it by e-mail:

```
reportbug {{package}}
```

- Report a bug that is not about a specific package (general problem, infrastructure, etc.):

```
reportbug other
```

- Write the bug report to a file instead of sending it by e-mail:

```
reportbug {{[-o|--output]}} {{filename}} {{package}}
```

repquota

Display a summary of existing file quotas for a filesystem.

More information: https://manned.org/repquota.

- Report stats for all quotas in use:

```
sudo repquota {{[-a|--all]}}
```

- Report quota stats for all users, even those who aren't using any of their quota:

```
sudo repquota {{[-v|--verbose]}} {{filesystem}}
```

- Report on quotas for users only:

```
repquota {{[-u|--user]}} {{filesystem}}
```

- Report on quotas for groups only:

```
sudo repquota {{[-g|--group]}} {{filesystem}}
```

- Report on used quota and limits in a human-readable format:

```
sudo repquota {{[-s|--human-readable]}} {{filesystem}}
```

- Report on all quotas for users and groups in a human-readable format:

```
sudo repquota {{[-augs|--all --user --group --human-readable]}}
```

reptyr

Move a running process to a new terminal.

Best used when you forget to start a long running task in **`screen`**.

More information: https://github.com/nelhage/reptyr.

- Move a running process to your current terminal:

```
reptyr {{pid}}
```

- Attach to a process using its name:

```
reptyr $(pidof {{htop}})
```

reset

Reinitialize the current terminal. Clears the entire terminal screen.

More information: https://manned.org/reset.

- Reinitialize the current terminal:

```
reset
```

- Display the terminal type instead:

```
reset -q
```

resize

Resize terminal size to the window size.

More information: https://manned.org/resize.

- Resize the terminal:

```
resize
```

- Print terminal size:

```
resize -s
```

resize2fs

Resize an ext2, ext3 or ext4 filesystem.

Does not resize the underlying partition. The filesystem may have to be unmounted first, read the man page for more details.

More information: https://manned.org/resize2fs.

- Automatically resize a filesystem to its maximum possible size:

```
resize2fs {{/dev/sdXN}}
```

- Resize the filesystem to a size of 40G, displaying a progress bar:

```
resize2fs -p {{/dev/sdXN}} {{40G}}
```

- Shrink the filesystem to its minimum possible size:

```
resize2fs -M {{/dev/sdXN}}
```

resolvectl

Resolve domain names, IPv4 and IPv6 addresses, DNS resource records, and services.

Introspect and reconfigure the DNS resolver.

More information: https://www.freedesktop.org/software/systemd/man/resolvectl.html.

- Show DNS settings:

```
resolvectl status
```

- Resolve the IPv4 and IPv6 addresses for one or more domains:

```
resolvectl query {{domain1 domain2 ...}}
```

- Retrieve the domain of a specified IP address:

```
resolvectl query {{ip_address}}
```

- Flush all local DNS caches:

```
resolvectl flush-caches
```

- Display DNS statistics (transactions, cache, and DNSSEC verdicts):

```
resolvectl statistics
```

- Retrieve an MX record of a domain:

```
resolvectl --legend {{no}} {{[-t|--type]}} {{MX}} query {{domain}}
```

- Resolve an SRV record, for example _xmpp-server._tcp gmail.com:

```
resolvectl service _{{service}}._{{protocol}} {{name}}
```

- Retrieve a TLS key:

```
resolvectl tlsa tcp {{domain}}:443
```

resolveip

Resolve hostnames to their IP addresses and vice versa.

More information: https://mariadb.com/kb/en/resolveip/.

- Resolve a hostname to an IP address:

```
resolveip {{example.org}}
```

- Resolve an IP address to a hostname:

```
resolveip {{1.1.1.1}}
```

- Resolve a hostname to an IP address with less output:

```
resolveip {{[-s|--silent]}} {{example.org}}
```

restorecon

Restore SELinux security context on files/directories according to persistent rules.

See also: **`semanage-fcontext`**.

More information: https://manned.org/restorecon.

- View the current security context of a file or directory:

```
ls {{[-dlZ|--directory -l --context]}} {{path/to/file_or_directory}}
```

- Restore the security context of a file or directory:

```
restorecon {{path/to/file_or_directory}}
```

- Restore the security context of a directory recursively, and show all changed labels:

```
restorecon -R -v {{path/to/directory}}
```

- Restore the security context of a directory recursively, using all available threads, and show progress:

```
restorecon -R -T {{0}} -p {{path/to/directory}}
```

- Preview the label changes that would happen without applying them:

```
restorecon -R -n -v {{path/to/directory}}
```

restorecond

Daemon that monitors file creation and automatically restores SELinux contexts.

Useful for directories where files are frequently created with incorrect contexts.

See also: **`restorecon`**, **`semanage-fcontext`**.

More information: https://manned.org/restorecond.

- Start the `restorecond` daemon:

```
sudo restorecond
```

- Run `restorecond` in [v]erbose mode to see restoration events:

```
sudo restorecond -v
```

- Run `restorecond` in [d]ebug mode:

```
sudo restorecond -d
```

- Use alternative restorecond.conf file:

```
sudo restorecond -f restorecond_file
```

- Check the status of the restorecond service:

```
sudo systemctl status restorecond
```

- Enable restorecond to start at boot:

```
sudo systemctl enable restorecond --now
```

retroarch

A frontend for emulators, game engines and media players.

The reference implementation of the libretro API.

More information: https://github.com/libretro/RetroArch.

- Start in the menu mode:

```
retroarch
```

- Start in full screen mode:

```
retroarch --fullscreen
```

- List all compiled features:

```
retroarch --features
```

- Set the path of a configuration file:

```
retroarch --config={{path/to/config_file}}
```

- Display help:

```
retroarch --help
```

- Display version:

```
retroarch --version
```

rev

Reverse a line of text or a file.

More information: https://manned.org/rev.

- Reverse text typed into terminal:

```
rev
```

- Reverse the text string "hello":

```
echo "hello" | rev
```

- Reverse an entire file and print to `stdout`:

```
rev {{path/to/file}}
```

- Use '\0' as a line separator (zero termination):

```
rev {{[-0|--zero]}} {{path/to/file}}
```

- Display help:

```
rev {{[-h|--help]}}
```

- Display version:

```
rev {{[-V|--version]}}
```

rexec

Execute a command on a remote host.

Note: Use **rexec** with caution, as it transmits data in plain text. Consider secure alternatives like SSH for encrypted communication.

More information: https://www.gnu.org/software/inetutils/manual/inetutils.html#rexec-invocation.

- Execute a command on a remote host:

```
rexec {{[-h|--host]}} {{remote_host}} {{ls -l}}
```

- Specify the remote username on a remote host:

```
rexec {{[-u|--username]}} {{username}} {{[-h|--host]}} {{remote_host}} {{ps aux}}
```

- Redirect `stdin` from `/dev/null` on a remote host:

```
rexec {{[-n|--noerr]}} {{[-h|--host]}} {{remote_host}} {{ls -l}}
```

- Specify the remote port on a remote host:

```
rexec {{[-P|--port]}} {{1234}} {{[-h|--host]}} {{remote_host}} {{ls -l}}
```

rfkill

Enable and disable wireless devices.

More information: https://manned.org/rfkill.

- List devices:

```
rfkill
```

- Filter by columns:

```
rfkill {{[-o|--output]}} {{ID,TYPE,DEVICE}}
```

- Block devices by type (e.g. bluetooth, wlan):

```
rfkill block {{bluetooth}}
```

- Unblock devices by type (e.g. bluetooth, wlan):

```
rfkill unblock {{wlan}}
```

- Output in JSON format:

```
rfkill {{[-J|--json]}}
```

rig

Utility to piece together a random first name, last name, street number and address, along with a geographically consistent (ie, they all match the same area) city, state, ZIP code, and area code.

More information: https://manned.org/rig.

- Display a random name (male or female) and address:

```
rig
```

- Display a [m]ale (or [f]emale) random name and address:

```
rig -{{m|f}}
```

- Use [d]ata files from a specific directory (default is `/usr/share/rig`):

```
rig -d {{path/to/directory}}
```

- Display a specific number of identities:

```
rig -c {{number}}
```

- Display a specific number of [f]emale identities:

```
rig -f -c {{number}}
```

ripmime

Extract attachments out of a MIME encoded email package.

More information: https://pldaniels.com/ripmime.

- Extract file contents in the current directory:

```
ripmime -i {{path/to/file}}
```

- Extract file contents in a specific directory:

```
ripmime -i {{path/to/file}} -d {{path/to/directory}}
```

- Extract file contents and print verbose output:

```
ripmime -i {{path/to/file}} -v
```

- Get detailed information about the whole decoding process:

```
ripmime -i {{path/to/file}} --debug
```

rkhunter

Searches for rootkits and malware.

More information: https://manned.org/rkhunter.

- Check a system for rootkits and malware:

```
sudo rkhunter --check
```

- Update rkhunter:

```
sudo rkhunter --update
```

- Print all available tests:

```
sudo rkhunter --list
```

- Display version:

```
sudo rkhunter --versioncheck
```

- Display help:

```
sudo rkhunter --help
```

rlogin

Log in to a remote host.

More information: https://www.gnu.org/software/inetutils/manual/inetutils.html#rlogin-invocation.

- Log in to a remote host:

```
rlogin {{remote_host}}
```

- Log in to a remote host with a specific username:

```
rlogin {{[-l|--user]}} {{username}} {{remote_host}}
```

rmmod

Remove modules from the Linux kernel.

See also: **kmod** for other module management commands.

More information: https://manned.org/rmmod.

- Remove a module from the kernel:

```
sudo rmmod {{module_name}}
```

- Remove a module from the kernel and display verbose information:

```
sudo rmmod --verbose {{module_name}}
```

- Remove a module from the kernel and send errors to syslog instead of `stderr`:

```
sudo rmmod --syslog {{module_name}}
```

- Display help:

```
rmmod --help
```

- Display version:

```
rmmod --version
```

rnm

Bulk Rename Utility.

More information: https://github.com/neurobin/rnm#basic-options.

- Replace a search string with a replacement string in filenames:

```
rnm -ss {{old}} -rs {{new}} {{path/to/directory}}
```

- Use a fixed (literal) search and replace string instead of `regex`:

```
rnm -ssf {{old}} -rs {{new}} {{path/to/files}}
```

- Add an auto-incremented index to filenames starting from 1:

```
rnm -i 1 -inc 1 -rs {{_}} {{path/to/files}}
```

- Rename files using a list of new names from a text file:

```
rnm -ns/f {{path/to/names.txt}} {{path/to/files}}
```

- Rename only files (ignoring directories and links):

```
rnm -fo -ss {{pattern}} -rs {{replacement}} {{path/to/files}}
```

- Sort input files by modification time before renaming:

```
rnm -s/mt -ss {{pattern}} -rs {{replacement}} {{path/to/
files}}
```

- Run a simulation without making actual changes:

```
rnm -sim -ss {{pattern}} -rs {{replacement}} {{path/to/
files}}
```

- Undo the last renaming operation:

```
rnm -u
```

rofi

An application launcher and window switcher.

More information: https://github.com/davatorium/rofi#manpage.

- Show the list of apps:

```
rofi -show drun
```

- Show the list of all commands:

```
rofi -show run
```

- Switch between windows:

```
rofi -show window
```

- Pipe a list of items to `stdin` and print the selected item to `stdout`:

```
printf "{{Choice1\nChoice2\nChoice3}}" | rofi -dmenu
```

rolldice

Roll virtual dice.

More information: https://manned.org/rolldice.

- Roll a single 20 sided dice:

```
rolldice d{{20}}
```

- Roll two six sided dice and drop the lowest roll:

```
rolldice {{2}}d{{6}}s{{1}}
```

- Roll two 20 sided dice and add a modifier value:

```
rolldice {{2}}d{{20}}{{+5}}
```

- Roll a 20 sided dice two times:

```
rolldice {{2}}xd{{20}}
```

routel

List IP routing in a human readable format.

See also: **`ip route`**, **`route`**.

More information: https://manned.org/man/routel.

- Display the default routing table:

```
routel
```

- Display a specific routing table:

```
routel {{table_number|main|local|default}}
```

- Display only IPv4 routes:

```
routel {{[-4|--family inet]}}
```

- Display only IPv6 routes:

```
routel {{[-6|--family inet6]}}
```

rpcclient

MS-RPC client tool (part of the samba suite).

More information: https://www.samba.org/samba/docs/current/man-html/rpcclient.1.html.

- Connect to a remote host:

```
rpcclient {{[-U|--user]}} {{domain}}\{{username}}%{{password}} {{ip}}
```

- Connect to a remote host on a domain without a password:

```
rpcclient {{[-U|--user]}} {{username}} {{[-W|--workgroup]}} {{domain}} {{[-N|--no-pass]}} {{ip}}
```

- Connect to a remote host, passing the password hash:

```
rpcclient {{[-U|--user]}} {{domain}}\{{username}} --pw-nt-hash {{ip}}
```

- Execute shell commands on a remote host:

```
rpcclient {{[-U|--user]}} {{domain}}\{{username}}%{{password}} {{[-c|--command]}} {{semicolon_separated_commands}} {{ip}}
```

- Display domain users:

```
rpcclient $> enumdomusers
```

- Display privileges:

```
rpcclient $> enumprivs
```

- Display information about a specific user:

```
rpcclient $> queryuser {{username|rid}}
```

- Create a new user in the domain:

```
rpcclient $> createdomuser {{username}}
```

rpcinfo

Make an RPC call to an RPC server and reports what it finds.

More information: https://manned.org/rpcinfo.

- Show full table of all RPC services registered on localhost:

```
rpcinfo
```

- Show concise table of all RPC services registered on localhost:

```
rpcinfo -s {{localhost}}
```

- Display table of statistics of rpcbind operations on localhost:

```
rpcinfo -m
```

- Display list of entries of given service name (mountd) and version number (2) on a remote nfs share:

```
rpcinfo -l {{remote_nfs_server_ip}} {{mountd}} {{2}}
```

- Delete the registration for version 1 of the mountd service for all transports:

```
rpcinfo -d {{mountd}} {{1}}
```

rpi-eeprom-update

Update EEPROM and view other EEPROM information.

More information: https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#rpi-eeprom-update.

- Print information about the current raspberry pi EEPROM installed:

```
sudo rpi-eeprom-update
```

- Update a raspberry pi EEPROM:

```
sudo rpi-eeprom-update -a
```

- Cancel the pending update:

```
sudo rpi-eeprom-update -r
```

- Display help:

```
rpi-eeprom-update -h
```

rpi-otp-private-key

Display the One-Time Programmable (OTP) private key of a Raspberry Pi.

More information: https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#program-a-key-into-otp-with-rpi-otp-private-key.

- Read the OTP private key:

```
rpi-otp-private-key
```

rpicam-hello

View a live camera stream using a Raspberry Pi camera.

More information: https://www.raspberrypi.com/documentation/computers/camera_software.html#rpicam-hello.

- Display a camera preview stream for a specific amount of time (in milliseconds):

```
rpicam-hello {{[-t|--timeout]}} {{time}}
```

- Tune the configuration for a particular camera sensor:

```
rpicam-hello --tuning-file {{/usr/share/libcamera/ipa/rpi/path/to/config.json}}
```

rpicam-jpeg

Capture and store a JPEG image using a Raspberry Pi camera.

More information: https://www.raspberrypi.com/documentation/computers/camera_software.html#rpicam-jpeg.

- Capture an image and name the file:

```
rpicam-jpeg {{[-o|--output]}} {{path/to/file.jpg}}
```

- Capture an image with set dimensions:

```
rpicam-jpeg {{[-o|--output]}} {{path/to/file.jpg}} --width {{1920}} --height {{1080}}
```

- Capture an image with an exposure of 20 seconds and a gain of 150%:

```
rpicam-jpeg {{[-o|--output]}} {{path/to/file.jpg}} --shutter 20000 --gain 1.5
```

rpicam-raw

Capture a raw video on a Raspberry Pi camera.

More information: https://www.raspberrypi.com/documentation/computers/camera_software.html#rpicam-raw.

- Capture a video for a specific amount of seconds:

```
rpicam-raw {{[-t|--timeout]}} {{2000}} {{[-o|--output]}} {{path/to/file.raw}}
```

- Change video dimensions and framerate:

```
rpicam-raw {{[-t|--timeout]}} {{5000}} --width {{4056}} --height {{3040}} {{[-o|--output]}} {{path/to/file.raw}} --framerate {{8}}
```

rpicam-still

Capture and store a photo using a Raspberry Pi camera with legacy features missing from **`rpicam-jpeg`**.

More information: https://www.raspberrypi.com/documentation/computers/camera_software.html#rpicam-still.

- Capture a photo with different encoding:

```
rpicam-still {{[-e|--encoding]}} {{bmp|png|rgb|yuv420}} {{[-o|--output]}} {{path/to/file.[bmp|png|rgb|yuv420]}}
```

- Capture a raw image:

```
rpicam-still {{[-r|--raw]}} {{[-o|--output]}} {{path/to/file.jpg}}
```

- Capture a 100 second exposure image:

```
rpicam-still {{[-o|--output]}} {{path/to/file.jpg}} --shutter 100000
```

rpicam-vid

Capture a video using a Raspberry Pi camera.

See also: **`vlc`**.

More information: https://www.raspberrypi.com/documentation/computers/camera_software.html#rpicam-vid.

- Capture a 10 second video:

```
rpicam-vid {{[-t|--timeout]}} 10000 {{[-o|--output]}} {{path/to/file.h264}}
```

rpm-ostree

A hybrid image/package system.

Manage ostree deployments, package layers, filesystem overlays, and boot configuration.

More information: https://coreos.github.io/rpm-ostree/administrator-handbook/.

- Show rpm-ostree deployments in the order they will appear in the bootloader:

```
rpm-ostree status
```

- Show packages which are outdated and can be updated:

```
rpm-ostree upgrade --preview
```

- Prepare a new ostree deployment with upgraded packages and reboot into it:

```
rpm-ostree upgrade {{[-r|--reboot]}}
```

- Reboot into the previous ostree deployment:

```
rpm-ostree rollback {{[-r|--reboot]}}
```

- Install a package into a new ostree deployment and reboot into it:

```
rpm-ostree install {{package}} {{[-r|--reboot]}}
```

rpm

RPM Package Manager.

For equivalent commands in other package managers, see https://wiki.archlinux.org/title/Pacman/Rosetta.

More information: https://rpm-software-management.github.io/rpm/man/rpm.8.

- Show version of httpd package:

```
rpm {{[-q|--query]}} httpd
```

- List versions of all matching packages:

```
rpm {{[-qa|--query --all]}} '{{mariadb*}}'
```

- Forcibly install a package regardless of currently installed versions:

```
rpm {{[-U|--upgrade]}} {{path/to/package.rpm}} --force
```

- Identify owner of a file and show version of the package:

```
rpm {{[-qf|--query --file]}} {{/etc/postfix/main.cf}}
```

- List package-owned files:

```
rpm {{[-ql|--query --list]}} {{kernel}}
```

- Show scriptlets from an RPM file:

```
rpm {{[-qp|--query --package]}} --scripts {{package.rpm}}
```

- Show changed, missing and/or incorrectly installed files of matching packages:

```
rpm {{[-Va|--verify --all]}} '{{php-*}}'
```

- Display the changelog of a specific package:

```
rpm {{[-q|--query]}} --changelog {{package}}
```

rpm2cpio

Convert an RPM package to a **`cpio`** archive.

More information: http://ftp.rpm.org/max-rpm/s1-rpm-miscellania-rpm2cpio.html.

- Convert an RPM package to a `cpio` archive and save it as `file.cpio` in the current directory:

```
rpm2cpio {{path/to/file.rpm}}
```

rpmbuild

RPM Package Build tool.

More information: https://manned.org/rpmbuild.

- Build binary and source packages:

```
rpmbuild -ba {{path/to/spec_file}}
```

- Build a binary package without source package:

```
rpmbuild -bb {{path/to/spec_file}}
```

- Specify additional variables when building a package:

```
rpmbuild -bb {{path/to/spec_file}} --define "{{variable1}} {{value1}}" --define "{{variable2}} {{value2}}"
```

rpmconf

Handle RPMNEW, RPMSAVE and RPMORIG files left over by package upgrades.

See also: **rpm**.

More information: https://github.com/xsuchy/rpmconf.

- List leftover files and interactively choose what to do with each of them:

```
sudo rpmconf --all
```

- Delete orphaned RPMNEW and RPMSAVE files:

```
sudo rpmconf --all --clean
```

rpmkeys

Tool to import and remove RPM keys for RPM repositories.

When adding an RPM repository, you must also import the corresponding RPM key.

More information: https://rpm-software-management.github.io/rpm/man/rpmkeys.8.

- List all imported RPM keys. Also outputs its Key ID needed for deleting a imported RPM key:

```
sudo rpmkeys --list
```

- Remove/Delete a previously imported RPM key, given by its 16-Number/Letters Key ID:

```
sudo rpmkeys --delete {{5a278d9c-5bbc73cb}}
```

- Import an RPM key of repository:

```
sudo rpmkeys --import {{path/to/rpm_key}}
```

rpmspec

Query a RPM spec file.

More information: https://manned.org/rpmspec.

- List binary packages which would be generated from a RPM spec file:

```
rpmspec --query {{path/to/rpm.spec}}
```

- List all options for `--queryformat`:

```
rpmspec --querytags
```

- Get summary information for single binary packages generated from a RPM spec file:

```
rpmspec --query --queryformat "{{%{name}: %{summary}
\n}}" {{path/to/rpm.spec}}
```

- Get the source package which would be generated from a RPM spec file:

```
rpmspec --query --srpm {{path/to/rpm.spec}}
```

- Parse a RPM spec file to `stdout`:

```
rpmspec --parse {{path/to/rpm.spec}}
```

rsh

Execute commands on a remote host.

More information: https://www.gnu.org/software/inetutils/manual/inetutils.html#rsh-invocation.

- Execute a command on a remote host:

```
rsh {{remote_host}} {{ls -l}}
```

- Execute a command on a remote host with a specific username:

```
rsh {{remote_host}} {{[-l|--user]}} {{username}} {{ls -l}}
```

- Redirect `stdin` to `/dev/null` when executing a command on a remote host:

```
rsh {{remote_host}} --no-err {{ls -l}}
```

rspamc

Client for **`rspamd`** servers.

More information: https://manned.org/rspamc.

- Train the bayesian filter to recognise an email as spam:

  ```
  rspamc learn_spam {{path/to/email_file}}
  ```

- Train the bayesian filter to recognise an email as ham:

  ```
  rspamc learn_ham {{path/to/email_file}}
  ```

- Generate a manual report on an email:

  ```
  rspamc symbols {{path/to/email_file}}
  ```

- Show server statistics:

  ```
  rspamc stat
  ```

rtcwake

Enter a system sleep state until specified wakeup time relative to your BIOS clock.

More information: https://manned.org/rtcwake.

- Show whether an alarm is set or not:

```
sudo rtcwake {{[-m|--mode]}} show {{[-v|--verbose]}}
```

- Suspend to RAM and wakeup after 10 seconds:

```
sudo rtcwake {{[-m|--mode]}} mem {{[-s|--seconds]}} {{10}}
```

- Suspend to disk (higher power saving) and wakeup 15 minutes later:

```
sudo rtcwake {{[-m|--mode]}} disk --date +{{15}}min
```

- Freeze the system (more efficient than suspend-to-RAM but version 3.9 or newer of the Linux kernel is required) and wakeup at a given date and time:

```
sudo rtcwake {{[-m|--mode]}} freeze --date {{YYYYMMDDhhmm}}
```

- Disable a previously set alarm:

```
sudo rtcwake {{[-m|--mode]}} disable
```

- Perform a dry run to wakeup the computer at a given time. (Press `<Ctrl c>` to abort):

```
sudo rtcwake {{[-m|--mode]}} on --date {{hh:ss}}
```

rtmon

Save network state changes to a file.

More information: https://manned.org/rtmon.

- Save all network state changes to a file:

```
sudo rtmon {{[f|file]}} {{path/to/file}}
```

- Specify the type of change to log:

```
sudo rtmon {{[f|file}} {{link|address|route}}
```

rtorrent

Download torrents.

More information: https://github.com/rakshasa/rtorrent.

- Add a torrent file or magnet to be downloaded:

```
rtorrent {{torrent_or_magnet}}
```

- Start the download:

```
<Ctrl s>
```

- View details about downloading torrent:

```
<ArrowRight>
```

- Close rtorrent safely:

```
<Ctrl q>
```

ruget

Alternative to wget written in Rust.

More information: https://github.com/ksk001100/ruget.

- Download the contents of a URL to a file:

```
ruget {{https://example.com/file}}
```

- Download the contents of a URL to a specified output file:

```
ruget {{[-o|--output]}} {{file_name}} {{https://example.com/file}}
```

run-mailcap

Run MailCap Programs.

Run mailcap view, see, edit, compose, print - execute programs via entries in the mailcap file (or any of its aliases) will use the given action to process each mime-type/file.

More information: https://manned.org/run-mailcap.

- Invoke individual actions/programs on run-mailcap:

```
run-mailcap --action={{view|cat|compose|composetyped|edit|print}} {{path/to/file}}
```

- Turn on extra information:

```
run-mailcap --action={{action}} --debug {{path/to/file}}
```

- Ignore any "copiousoutput" directive and forward output to `stdout`:

```
run-mailcap --action={{action}} --nopager {{path/to/file}}
```

- Display the found command without actually executing it:

```
run-mailcap --action={{action}} --norun {{path/to/file}}
```

run0

Elevate privileges interactively.

Similar to **sudo**, but it's not a SUID binary, authentication takes place via polkit, and commands are invoked from a **systemd** service.

See also: **sudo**, **pkexec**, **doas**.

More information: https://www.freedesktop.org/software/systemd/man/run0.html.

- Run a command as root:

```
run0 {{command}}
```

- Run a command as another user and/or group:

```
run0 {{[-u|--user]}} {{username|uid}} {{[-g|--group]}}
{{group_name|gid}} {{command}}
```

run_init

Run init scripts in the proper SELinux context.

Typically used to run system service scripts with correct SELinux domains.

See also: **runcon**, **semanage**.

More information: https://manned.org/run_init.

- Run a script in the init script context:

```
sudo run_init {{path/to/script}}
```

- Run a script with arguments:

```
sudo run_init {{path/to/script}} {{start|stop|restart}}
```

- Run a script and specify the init script context explicitly:

```
sudo run_init {{[-t|--type]}} {{context_type}} {{path/to/
script}}
```

- Display the context that would be used without running the script:

```
sudo run_init {{[-n|--dry-run]}} {{path/to/script}}
```

runcon

Run a program in a different SELinux security context.

See also: **secon**.

More information: https://www.gnu.org/software/coreutils/manual/html_node/runcon-invocation.html.

- Print the security context of the current execution context:

```
runcon
```

- Specify the domain to run a command in:

```
runcon {{[-t|--type]}} {{domain}}_t {{command}}
```

- Specify the context role to run a command with:

```
runcon {{[-r|--role]}} {{role}}_r {{command}}
```

- Specify the full context to run a command with:

```
runcon {{user}}_u:{{role}}_r:{{domain}}_t {{command}}
```

runlevel

SysV runlevel utility.

See also: **`init`**.

More information: https://manned.org/runlevel.

- Return previous and current SysV runlevel:

```
runlevel
```

- Display help:

```
runlevel --help
```

runlim

Sample and limit time and memory usage of a program and its child processes using the proc file system on Linux.

More information: https://fmv.jku.at/runlim.

- Print the time and memory usage of a command:

```
runlim {{command}} {{command_arguments}}
```

- Log statistics to a file instead of `stdout`:

```
runlim --output-file={{path/to/file}} {{command}}
{{command_arguments}}
```

- Limit time to an upper bound (in seconds):

```
runlim --time-limit={{number}} {{command}}
{{command_arguments}}
```

- Limit real-time to an upper bound (in seconds):

```
runlim --real-time-limit={{number}} {{command}}
{{command_arguments}}
```

- Limit space to an upper bound (in MB):

```
runlim --space-limit={{number}} {{command}}
{{command_arguments}}
```

runuser

Run commands as a user and group without asking for password (needs root privileges).

More information: https://manned.org/runuser.

- Run command as a different user:

```
runuser {{user}} {{[-c|--command]}} '{{command}}'
```

- Run command as a different user and group:

```
runuser {{user}} {{[-g|--group]}} {{group}} {{[-c|--command]}} '{{command}}'
```

- Start a login shell as a specific user:

```
runuser {{user}} {{[-l|--login]}}
```

- Specify a shell for running instead of the default shell (also works for login):

```
runuser {{user}} {{[-s|--shell]}} {{/bin/sh}}
```

- Preserve the entire environment of root (only if `--login` is not specified):

```
runuser {{user}} {{[-p|--preserve-environment]}} {{[-c|--command]}} '{{command}}'
```

rusnapshot

BTRFS snapshotting utility written in Rust.

More information: https://github.com/Edu4rdSHL/rusnapshot.

- Create a snapshot using a configuration file:

```
sudo rusnapshot {{[-c|--config]}} {{path/to/config.toml}} --cr
```

- List created snapshots:

```
sudo rusnapshot {{[-c|--config]}} {{path/to/config.toml}} {{[-l|--list]}}
```

- Delete a snapshot by ID or the name of the snapshot:

```
sudo rusnapshot {{[-c|--config]}} {{path/to/config.toml}} --del --id {{snapshot_id}}
```

- Delete all `hourly` snapshots:

```
sudo rusnapshot {{[-c|--config]}} {{path/to/config.toml}} {{[-l|--list]}} {{[-k|--keep]}} {{0}} --clean --kind {{hourly}}
```

- Create a read-write snapshot:

```
sudo rusnapshot {{[-c|--config]}} {{path/to/config.toml}} --cr {{[-r|--rw]}}
```

- Restore a snapshot:

```
sudo rusnapshot {{[-c|--config]}} {{path/to/config.toml}} --id {{snapshot_id}} {{[-r|--restore]}}
```

sa

Summarize accounting information about commands called by users, including basic information on CPU time spent processing and I/O rates.

Part of the **acct** package.

More information: https://manned.org/sa.8.

- Display executable invocations per user (username not displayed):

```
sudo sa
```

- Display executable invocations per user, showing responsible usernames:

```
sudo sa --print-users
```

- List resources used recently per user:

```
sudo sa --user-summary
```

sacct

Display accounting data from the Slurm service.

More information: https://slurm.schedmd.com/sacct.html.

- Display job ID, job name, partition, account, number of allocated cpus, job state, and job exit codes for recent jobs:

```
sacct
```

- Display job ID, job state, job exit code for recent jobs:

```
sacct {{[-b|--brief]}}
```

- Display the allocations of a job:

```
sacct {{[-j|--jobs]}} {{job_id}} {{[-X|--allocations]}}
```

- Display elapsed time, job name, number of requested CPUs, and memory requested of a job:

```
sacct {{[-j|--jobs]}} {{job_id}} {{[-o|--format]}}
Elapsed,JobName,ReqCPUS,ReqMem
```

- Display recent jobs that occurred from one week ago up to the present day:

```
sacct {{[-S|--starttime]}} $(date {{[-d|--date]}} "1 week
ago" +'%F')
```

- Output a larger number of characters for an attribute:

```
sacct {{[-o|--format]}} JobID,JobName%100
```

sacctmgr

View, setup, and manage Slurm accounts.

More information: https://slurm.schedmd.com/sacctmgr.html.

- Show current configuration:

```
sacctmgr show configuration
```

- Add a cluster to the slurm database:

```
sacctmgr add cluster {{cluster_name}}
```

- Add an account to the slurm database:

```
sacctmgr add account {{account_name}}
cluster={{cluster_of_account}}
```

- Show details of user/association/cluster/account using a specific format:

```
sacctmgr show {{user|association|cluster|account}}
format="Account%10" format="GrpTRES%30"
```

salloc

Start an interactive shell session or execute a command by allocating one or more nodes in a cluster.

More information: https://slurm.schedmd.com/salloc.html.

- Start an interactive shell session on a node in the cluster:

```
salloc
```

- Execute the specified command synchronously on a node in the cluster:

```
salloc {{ls --all}}
```

- Only allocate nodes fulfilling the specified constraints:

```
salloc {{[-C|--constraint]}} {{(amd|intel)&gpu}}
```

sam

AWS Serverless Application Model (SAM) CLI.

More information: https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/using-sam-cli-corecommands.html.

- Initialize a serverless application:

```
sam init
```

- Initialize a serverless application with a specific runtime:

```
sam init {{[-r|--runtime]}} {{python3.7}}
```

- Package a SAM application:

```
sam package
```

- Build your Lambda function code:

```
sam build
```

- Run your serverless application locally:

```
sam local start-api
```

- Deploy an AWS SAM application:

```
sam deploy
```

sar

Monitor performance of various Linux subsystems.

More information: https://manned.org/sar.

- Report I/O and transfer rate issued to physical devices, one per second (press `<Ctrl c>` to quit):

```
sar -b {{1}}
```

- Report a total of 10 network device statistics, one per 2 seconds:

```
sar -n DEV {{2}} {{10}}
```

- Report CPU utilization, one per 2 seconds:

```
sar -u ALL {{2}}
```

- Report a total of 20 memory utilization statistics, one per second:

```
sar -r ALL {{1}} {{20}}
```

- Report the run queue length and load averages, one per second:

```
sar -q {{1}}
```

- Report paging statistics, one per 5 seconds:

```
sar -B {{5}}
```

sattach

Attach to a Slurm job step.

More information: https://slurm.schedmd.com/sattach.html.

- Redirect the IO streams (`stdout`, `stderr`, and `stdin`) of a Slurm job step to the current terminal:

```
sattach {{jobid}}.{{stepid}}
```

- Use the current console's input as `stdin` to the specified task:

```
sattach --input-filter {{task_number}}
```

- Only redirect `stdin`/`stderr` of the specified task:

```
sattach --{{output|error}}-filter {{task_number}}
```

sbatch

Submit a batch job to the SLURM scheduler.

More information: https://manned.org/sbatch.

- Submit a batch job:

```
sbatch {{path/to/job.sh}}
```

- Submit a batch job with a custom name:

```
sbatch --job-name={{myjob}} {{path/to/job.sh}}
```

- Submit a batch job with a time limit of 30 minutes:

```
sbatch --time={{00:30:00}} {{path/to/job.sh}}
```

- Submit a job and request multiple nodes:

```
sbatch --nodes={{3}} {{path/to/job.sh}}
```

sbcast

Send a file to a job's allocated nodes.

This command should only be used from within a Slurm batch job.

More information: https://slurm.schedmd.com/sbcast.html.

- Send a file to all nodes allocated to the current job:

```
sbcast {{path/to/file}} {{path/to/destination}}
```

- Autodetect shared libraries the transmitted file depends upon and transmit them as well:

```
sbcast --send-libs={{yes}} {{path/to/executable}} {{path/to/destination}}
```

sbctl

A user-friendly secure boot key manager.

Note: Not enrolling Microsoft's certificates can brick your system. See https://github.com/Foxboron/sbctl/wiki/FAQ#option-rom.

More information: https://github.com/Foxboron/sbctl#usage.

- Show the current secure boot status:

```
sbctl status
```

- Create custom secure boot keys (by default, everything is stored in `/var/lib/sbctl`):

```
sbctl create-keys
```

- Enroll the custom secure boot keys and Microsoft's UEFI vendor certificates:

```
sbctl enroll-keys {{[-m|--microsoft]}}
```

- Automatically run `create-keys` and `enroll-keys` based on the settings in `/etc/sbctl/sbctl.conf`:

```
sbctl setup --setup
```

- Sign an EFI binary with the created key and save the file to the database:

```
sbctl sign {{[-s|--save]}} {{path/to/efi_binary}}
```

- Re-sign all the saved files:

```
sbctl sign-all
```

- Verify that all EFI executables on the EFI system partition have been signed:

```
sbctl verify
```

scancel

Cancel a Slurm job.

More information: https://slurm.schedmd.com/scancel.html.

- Cancel a job using its ID:

```
scancel {{job_id}}
```

- Cancel all jobs from a user:

```
scancel {{user_name}}
```

scanimage

Scan images with the Scanner Access Now Easy API.

More information: http://sane-project.org/man/scanimage.1.html.

- List available scanners to ensure the target device is connected and recognized:

```
scanimage {{[-L|--list-devices]}}
```

- Scan an image and save it to a file:

```
scanimage --format {{pnm|tiff|png|jpeg|pdf|...}} > {{path/to/new_image}}
```

- Specify the device to scan from:

```
scanimage {{[-d|--device]}} {{device_name}} > {{path/to/new_image}}
```

- Specify resolution for the scanned image (default resolution is 75dpi):

```
scanimage --resolution {{300}} > {{path/to/new_image}}
```

schroot

Run a command or start an interactive shell with a different root directory. More customizable than **`chroot`**.

More information: https://wiki.debian.org/Schroot.

- List available chroots:

```
schroot --list
```

- Run a command in a specific chroot:

```
schroot --chroot {{chroot}} {{command}}
```

- Run a command with options in a specific chroot:

```
schroot --chroot {{chroot}} {{command}} --
{{command_options}}
```

- Run a command in all available chroots:

```
schroot --all {{command}}
```

- Start an interactive shell within a specific chroot as a specific user:

```
schroot --chroot {{chroot}} --user {{user}}
```

- Begin a new session (a unique session ID is returned on `stdout`):

```
schroot --begin-session --chroot {{chroot}}
```

- Connect to an existing session:

```
schroot --run-session --chroot {{session_id}}
```

- End an existing session:

```
schroot --end-session --chroot {{session_id}}
```

scontrol

View information about and modify jobs.

More information: https://slurm.schedmd.com/scontrol.html.

- Show information for job:

```
scontrol show job {{job_id}}
```

- Suspend a comma-separated list of running jobs:

```
scontrol suspend {{job_id1,job_id2,...}}
```

- Resume a comma-separated list of suspended jobs:

```
scontrol resume {{job_id1,job_id2,...}}
```

- Hold a comma-separated list of queued jobs (Use `release` command to permit the jobs to be scheduled):

```
scontrol hold {{job_id1,job_id2,...}}
```

- Release a comma-separated list of suspended job:

```
scontrol release {{job_id1,job_id2,...}}
```

screenkey

A screencast tool to display keys pressed.

More information: https://www.thregr.org/~wavexx/software/screenkey/.

- Display keys which are currently being pressed on the screen:

```
screenkey
```

- Display keys and mouse buttons which are currently being pressed on the screen:

```
screenkey {{[-M|--mouse]}}
```

- Launch the settings menu of screenkey:

```
screenkey --show-settings
```

- Launch screenkey at a specific position:

```
screenkey {{[-p|--position]}} {{top|center|bottom|fixed}}
```

- Change the format of the key modifiers displayed on screen:

```
screenkey --mods-mode {{normal|emacs|mac|win|tux}}
```

- Change the appearance of screenkey:

```
screenkey --bg-color "{{#a1b2c3}}" {{[-f|--font]}} {{Hack}}
--font-color {{yellow}} --opacity {{0.8}}
```

- Drag and select a window on screen to display screenkey:

```
screenkey {{[-p|--position]}} fixed {{[-g|--geometry]}} {{$
(slop --nodecorations --format '%g')}}
```

script

Record all terminal output to a typescript file.

More information: https://manned.org/script.

- Record a new session to a file named `typescript` in the current directory:

```
script
```

- Stop recording:

```
exit
```

- Record a new session to a custom filepath:

```
script {{path/to/session.out}}
```

- Append to an existing file:

```
script {{[-a|--append]}} {{logfile.log}}
```

- Record timing information (data is outputted to `stderr`):

```
script {{[-t|--timing]}} 2> {{path/to/timing_file}}
```

- Write out data as soon as it happens:

```
script {{[-f|--flush]}} {{path/to/file}}
```

- Execute quietly without start and done messages:

```
script {{[-q|--quiet]}} {{logfile.log}}
```

- Display help:

```
script {{[-h|--help]}}
```

scriptlive

Execute a typescript created by the **`script`** command in real-time.

See also: **`script`**.

More information: https://manned.org/scriptlive.

- Execute a typescript in real-time:

```
scriptlive {{path/to/timing_file}} {{path/to/typescript}}
```

- Execute a typescript at double the original speed:

```
scriptlive {{path/to/timing_file}} {{path/to/typescript}} --
divisor 2
```

- Execute a typescript created using `--log-in` option of `script`:

```
scriptlive --log-in {{path/to/stdin_log_file}} {{path/to/
typescript}}
```

- Execute a typescript waiting at most 2 seconds between each command:

```
scriptlive {{path/to/timing_file}} {{path/to/typescript}} --
maxdelay 2
```

scriptreplay

Replay a typescript created by the **`script`** command to **`stdout`**.

More information: https://manned.org/scriptreplay.

- Replay a typescript at the speed it was recorded:

```
scriptreplay {{path/to/timing_file}} {{path/to/typescript}}
```

- Replay a typescript at double the original speed:

```
scriptreplay {{path/to/timingfile}} {{path/to/typescript}} 2
```

- Replay a typescript at half the original speed:

```
scriptreplay {{path/to/timingfile}} {{path/to/typescript}}
0.5
```

scrontab

Manage Slurm crontab files.

More information: https://slurm.schedmd.com/scrontab.html.

- Install a new crontab from the specified file:

```
scrontab {{path/to/file}}
```

- [e]dit the crontab of the current user:

```
scrontab -e
```

- [e]dit the crontab of the specified user:

```
scrontab --user={{user_id}} -e
```

- [r]emove the current crontab:

```
scrontab -r
```

- Print the crontab of the current user to `stdout`:

```
scrontab -l
```

scrot

Screen capture utility.

More information: https://github.com/resurrecting-open-source-projects/scrot.

- Capture a screenshot and save it to the current directory with the current date as the filename:

```
scrot
```

- Capture a screenshot and save it as `capture.png`:

```
scrot {{capture.png}}
```

- Capture a screenshot interactively:

```
scrot --select
```

- Capture a screenshot interactively without exiting on keyboard input, press `<Esc>` to exit:

```
scrot --select --ignorekeyboard
```

- Capture a screenshot interactively delimiting the region with a colored line:

```
scrot --select --line color={{x11_color|rgb_color}}
```

- Capture a screenshot from the currently focused window:

```
scrot --focused
```

- Display a countdown of 10 seconds before taking a screenshot:

```
scrot --count --delay {{10}}
```

scrun

An OCI runtime proxy for Slurm that runs containers as jobs.

More information: https://slurm.schedmd.com/scrun.html.

- Create a new container with a specific ID:

```
scrun create {{container_id}}
```

- Start a previously created container:

```
scrun start {{container_id}}
```

- Query the state of a container:

```
scrun state {{container_id}}
```

- Send a signal to a container (default: SIGTERM):

```
scrun kill {{container_id}}
```

- Send a specific signal to a container:

```
scrun kill {{container_id}} {{SIGKILL}}
```

- Delete a container and release its resources:

```
scrun delete {{container_id}}
```

- Enable debug logging:

```
scrun {{create|start|kill|delete}} {{container_id}} --debug
```

- Display version:

```
scrun --version
```

sdiag

Show information about the execution of **`slurmctld`**.

More information: https://slurm.schedmd.com/sdiag.html.

- Show all performance counters related to the execution of `slurmctld`:

```
sdiag {{[-a|--all]}}
```

- Reset performance counters related to the execution of `slurmctld`:

```
sdiag {{[-r|--reset]}}
```

- Specify the output format:

```
sdiag {{[-a|--all]}} --{{json|yaml}}
```

- Specify the cluster to send commands to:

```
sdiag {{[-a|--all]}} {{[-M|--cluster]}} {{cluster_name}}
```

sealert

Analyze and explain SELinux AVC denial messages.

Part of the **`setroubleshoot-server`** package.

See also: **`audit2why`**, **`ausearch`**, **`audit2allow`**.

More information: https://manned.org/sealert.

- Analyze all recent SELinux denials:

```
sudo sealert {{[-a|--analyze]}} {{/var/log/audit/audit.log}}
```

- Analyze a specific alert ID from system logs:

```
sudo sealert {{[-l|--lookupid]}} {{alert_id}}
```

- Display a summary of recent SELinux alerts:

```
sudo sealert {{[-b|--browser]}}
```

- Monitor audit log in real-time for new alerts:

```
sudo tail {{[-f|--follow]}} {{/var/log/audit/audit.log}} |
sealert {{[-l|--lookupid]}} -
```

secon

Get the SELinux security context of a file, pid, current execution context, or a context specification.

See also: **`semanage`**, **`runcon`**, **`chcon`**.

More information: https://manned.org/secon.

- Get the security context of the current execution context:

```
secon
```

- Get the current security context of a process:

```
secon --pid {{1}}
```

- Get the current security context of a file, resolving all intermediate symlinks:

```
secon --file {{path/to/file_or_directory}}
```

- Get the current security context of a symlink itself (i.e. do not resolve):

```
secon --link {{path/to/symlink}}
```

- Parse and explain a context specification:

```
secon {{system_u:system_r:container_t:s0:c899,c900}}
```

secret-tool

Store and retrieve passwords, part of the **`libsecret`** package.

Communicates with Freedesktop secret service implementations such as **`gnome-keyring`**.

More information: https://gnome.pages.gitlab.gnome.org/libsecret/.

- Store a secret with an optional label:

```
secret-tool store --label={{label}} {{key}} {{value}}
```

- Retrieve a secret:

```
secret-tool lookup key {{key}}
```

- Get more information about a secret:

```
secret-tool search key {{key}}
```

- Delete a stored secret:

```
secret-tool clear key {{key}}
```

sed

GNU stream editor for filtering and transforming text.

See also: **awk**, **ed**.

More information: https://www.gnu.org/software/sed/manual/sed.html.

- Replace `apple` with `mango` on all lines using basic `regex`, print to `stdout`:

```
{{command}} | sed 's/apple/mango/g'
```

- Replace `apple` with `APPLE` on all lines using extended `regex`, print to `stdout`:

```
{{command}} | sed {{[-E|--regexp-extended]}} 's/(apple)/\U\1/g'
```

- Use basic `regex` to replace `apple` with `mango` and `orange` with `lime` in-place in a file (overwriting original file):

```
sed {{[-i|--in-place]}} -e 's/apple/mango/g' -e 's/orange/lime/g' {{path/to/file}}
```

- Execute a specific `sed` script file and print the result to `stdout`:

```
{{command}} | sed {{[-f|--file]}} {{path/to/script.sed}}
```

- [p]rint only the first line to `stdout`:

```
{{command}} | sed {{[-n|--quiet]}} '1p'
```

- [d]elete lines 1 to 5 of a file and back up the original file with a `.orig` extension:

```
sed {{[-i|--in-place=]}}{{.orig}} '1,5d' {{path/to/file}}
```

- [i]nsert a new line at the beginning of a file, overwriting the original file in-place:

```
sed {{[-i|--in-place]}} '1i\your new line text\' {{path/to/file}}
```

- Delete blank lines (with or without spaces/tabs) from a file, overwriting the original file in-place:

```
sed {{[-i|--in-place]}} '/^[[:space:]]*$/d' {{path/to/file}}
```

see

Alias to **`run-mailcap`**'s view.

An alias to a **`run-mailcap`**'s action print.

More information: https://manned.org/see.

- See action can be used to view any file (usually image) on default mailcap explorer:

  ```
  see {{filename}}
  ```

- Using with `run-mailcap`:

  ```
  run-mailcap --action=view {{filename}}
  ```

selinuxenabled

Check whether SELinux is enabled.

Returns exit code 0 if SELinux is enabled, and 1 if it is not.

See also: **`getenforce`**, **`setenforce`**, **`sestatus`**.

More information: https://manned.org/selinuxenabled.

- Check if SELinux is enabled (no output; check exit code with `echo $?`):

```
selinuxenabled
```

- Check if SELinux is enabled and print the result:

```
selinuxenabled && echo "SELinux is enabled" || echo "SELinux is disabled"
```

- Use in a shell script to conditionally execute commands:

```
if selinuxenabled; then echo "SELinux is running"; fi
```

semanage boolean

Manage persistent SELinux boolean settings.

See also: **semanage** for managing SELinux policies, **getsebool** for checking boolean values, and **setsebool** for applying non-persistent boolean settings.

More information: https://manned.org/semanage-boolean.

- List all booleans settings:

```
sudo semanage boolean {{[-l|--list]}}
```

- List all user-defined boolean settings without headings:

```
sudo semanage boolean {{[-l|--list]}} {{[-C|--locallist]}}
{{[-n|--noheading]}}
```

- Set or unset a boolean persistently:

```
sudo semanage boolean {{[-m|--modify]}} {{-1|--on|-0|--off}}
{{haproxy_connect_any}}
```

semanage fcontext

Manage persistent SELinux security context rules on files/directories.

See also: **`semanage`**, **`matchpathcon`**, **`secon`**, **`chcon`**, **`restorecon`**.

More information: https://manned.org/semanage-fcontext.

- List all file labelling rules:

```
sudo semanage fcontext {{[-l|--list]}}
```

- List all user-defined file labelling rules without headings:

```
sudo semanage fcontext {{[-l|--list]}} {{[-C|--locallist]}}
{{[-n|--noheading]}}
```

- Add a user-defined rule that labels any path which matches a PCRE `regex`:

```
sudo semanage fcontext {{[-a|--add]}} {{[-t|--type]}}
{{samba_share_t}} '{{/mnt/share(/.*)?}}'
```

- Delete a user-defined rule using its PCRE `regex`:

```
sudo semanage fcontext {{[-d|--delete]}} '{{/mnt/
share(/.*)?}}'
```

- Relabel a directory recursively by applying the new rules:

```
restorecon -R -v {{path/to/directory}}
```

semanage interface

Manage SELinux network interface type definitions.

See also: **semanage**, **semanage-port**.

More information: https://manned.org/semanage-interface.

- List all interface type definitions:

```
sudo semanage interface {{[-l|--list]}}
```

- Add a network interface type definition:

```
sudo semanage interface {{[-a|--add]}} {{[-t|--type]}}
{{type_name}} {{interface_name}}
```

- Delete a network interface type definition:

```
sudo semanage interface {{[-d|--delete]}} {{interface_name}}
```

- Modify a network interface type definition:

```
sudo semanage interface {{[-m|--modify]}} {{[-t|--type]}}
{{type_name}} {{interface_name}}
```

- List interface type definitions in a customized format:

```
sudo semanage interface {{[-l|--list]}} {{[-C|--locallist]}}
```

semanage-login

Manage SELinux login mappings between Linux users and SELinux users.

See also: **`semanage`**, **`semanage-user`**.

More information: https://manned.org/semanage-login.

- List all login mappings:

```
sudo semanage login {{[-l|--list]}}
```

- Add a login mapping (map Linux user to SELinux user):

```
sudo semanage login {{[-a|--add]}} {{[-s|--seuser]}}
{{selinux_user}} {{linux_username}}
```

- Delete a login mapping:

```
sudo semanage login {{[-d|--delete]}} {{linux_username}}
```

- Modify an existing login mapping:

```
sudo semanage login {{[-m|--modify]}} {{[-s|--seuser]}}
{{selinux_user}} {{linux_username}}
```

- Add a login mapping with a specific MLS/MCS range:

```
sudo semanage login {{[-a|--add]}} {{[-s|--seuser]}}
{{user_u}} {{[-r|--range]}} {{s0-s0:c0.c1023}}
{{linux_username}}
```

- List only customized login mappings:

```
sudo semanage login {{[-l|--list]}} {{[-C|--locallist]}}
```

semanage permissive

Manage persistent SELinux permissive domains.

Note that this effectively makes the process unconfined. For long-term use, it is recommended to configure SELiunx properly.

See also: **`semanage`**, **`getenforce`**, **`setenforce`**.

More information: https://manned.org/semanage-permissive.

- List all process types (a.k.a domains) that are in permissive mode:

```
sudo semanage permissive {{[-l|--list]}}
```

- Set permissive mode for a domain:

```
sudo semanage permissive {{[-a|--add]}} {{httpd_t}}
```

- Unset permissive mode for a domain:

```
sudo semanage permissive {{[-d|--delete]}} {{httpd_t}}
```

semanage port

Manage persistent SELinux port definitions.

See also: **semanage**.

More information: https://manned.org/semanage-port.

- List all port labeling rules:

```
sudo semanage port {{[-l|--list]}}
```

- List all user-defined port labeling rules without headings:

```
sudo semanage port {{[-l|--list]}} {{[-C|--locallist]}} {{[-
n|--noheading]}}
```

- Add a user-defined rule that assigns a label to a protocol-port pair:

```
sudo semanage port {{[-a|--add]}} {{[-t|--type]}}
{{ssh_port_t}} {{[-p|--proto]}} {{tcp}} {{22000}}
```

- Add a user-defined rule that assigns a label to a protocol-port-range pair:

```
sudo semanage port {{[-a|--add]}} {{[-t|--type]}}
{{http_port_t}} {{[-p|--proto]}} {{tcp}} {{80-88}}
```

- Delete a user-defined rule using its protocol-port pair:

```
sudo semanage port {{[-d|--delete]}} {{[-p|--proto]}} {{udp}}
{{11940}}
```

semanage-user

Manage SELinux user mappings.

See also: **`semanage`**, **`semanage-login`**.

More information: https://manned.org/semanage-user.

- List all SELinux users:

```
sudo semanage user {{[-l|--list]}}
```

- Add a new SELinux user:

```
sudo semanage user {{[-a|--add]}} {{[-R|--roles]}}
{{role_name}} {{selinux_user}}
```

- Delete a SELinux user:

```
sudo semanage user {{[-d|--delete]}} {{selinux_user}}
```

- Modify an existing SELinux user's roles:

```
sudo semanage user {{[-m|--modify]}} {{[-R|--roles]}}
{{role_name}} {{selinux_user}}
```

- Add a SELinux user with a specific default level:

```
sudo semanage user {{[-a|--add]}} {{[-R|--roles]}}
{{role_name}} {{[-L|--level]}} {{s0}} {{selinux_user}}
```

- Add a SELinux user with a specific MLS/MCS range:

```
sudo semanage user {{[-a|--add]}} {{[-R|--roles]}}
{{role_name}} {{[-r|--range]}} {{s0-s0:c0.c1023}}
{{selinux_user}}
```

- List only customized SELinux users:

```
sudo semanage user {{[-l|--list]}} {{[-C|--locallist]}}
```

semanage

SELinux persistent policy management tool.

Some subcommands such as **`boolean`**, **`fcontext`**, **`port`**, etc. have their own usage documentation.

More information: https://manned.org/semanage.

- Set or unset a SELinux boolean. Booleans allow the administrator to customize how policy rules affect confined process types (a.k.a domains):

```
sudo semanage boolean {{[-m|--modify]}} {{--on|--off}} {{haproxy_connect_any}}
```

- Add a user-defined file context labeling rule. File contexts define what files confined domains are allowed to access:

```
sudo semanage fcontext {{[-a|--add]}} {{[-t|--type]}} {{samba_share_t}} '/mnt/share(/.*)?'
```

- Add a user-defined port labeling rule. Port labels define what ports confined domains are allowed to listen on:

```
sudo semanage port {{[-a|--add]}} {{[-t|--type]}} {{ssh_port_t}} {{[-p|--proto]}} {{tcp}} {{22000}}
```

- Set or unset permissive mode for a confined domain. Per-domain permissive mode allows more granular control compared to `setenforce`:

```
sudo semanage permissive {{--add|--delete}} {{httpd_t}}
```

- Output local customizations in the default store:

```
sudo semanage export {{[-f|--output_file]}} {{path/to/file}}
```

- Import a file generated by `semanage export` into local customizations (CAREFUL: may remove current customizations!):

```
sudo semanage import {{[-f|--input_file]}} {{path/to/file}}
```

semodule

Manage SELinux policy modules.

See also: **`audit2allow`**, **`semanage`**.

More information: https://manned.org/semodule.

- List all installed policy modules:

```
sudo semodule {{[-l|--list]}}
```

- Install a new policy module:

```
sudo semodule {{[-i|--install]}} {{path/to/module.pp}}
```

- Remove a policy module:

```
sudo semodule {{[-r|--remove]}} {{module_name}}
```

- Enable a policy module:

```
sudo semodule {{[-e|--enable]}} {{module_name}}
```

- Disable a policy module:

```
sudo semodule {{[-d|--disable]}} {{module_name}}
```

- Reload all policy modules:

```
sudo semodule {{[-R|--reload]}}
```

- Display the version of installed policy modules:

```
sudo semodule {{[-l|--list]}} {{[-v|--verbose]}}
```

sensible-browser

Open the default browser.

More information: https://manned.org/sensible-browser.

- Open a new window of the default browser:

```
sensible-browser
```

- Open a URL in the default browser:

```
sensible-browser {{url}}
```

sensible-editor

Open the default editor.

More information: https://manned.org/sensible-editor.

- Open a file in the default editor:

```
sensible-editor {{path/to/file}}
```

- Open a file in the default editor, with the cursor at the end of the file:

```
sensible-editor + {{path/to/file}}
```

- Open a file in the default editor, with the cursor at the beginning of line 10:

```
sensible-editor +10 {{path/to/file}}
```

- Open 3 files in vertically split editor windows at the same time:

```
sensible-editor -O3 {{path/to/file1 path/to/file2 path/to/file3}}
```

sensors

Report sensors information.

More information: https://manned.org/sensors.

- Show the current readings of all sensor chips:

```
sensors
```

- Show temperatures in degrees Fahrenheit:

```
sensors --fahrenheit
```

service

Manage services by running init scripts.

The full script path should be omitted (**`/etc/init.d/`** is assumed).

More information: https://manned.org/service.

- List the name and status of all services:

```
service --status-all
```

- Start/Stop/Restart/Reload service (start/stop should always be available):

```
service {{service_name}} {{start|stop|restart|reload}}
```

- Do a full restart (runs script twice with start and stop):

```
service {{service_name}} --full-restart
```

- Show the current status of a service:

```
service {{service_name}} status
```

sesearch

Search SELinux policy rules.

Part of the **`setools`** package.

See also: **`seinfo`**, **`semodule`**.

More information: https://manned.org/sesearch.

- Search for all allow rules:

```
sesearch --allow
```

- Search for rules related to a specific type:

```
sesearch --allow {{[-t|--target]}} {{type_name}}
```

- Search for rules related to a specific source type:

```
sesearch --allow {{[-s|--source]}} {{source_type}}
```

- Search for rules that allow a specific class and permission:

```
sesearch --allow {{[-c|--class]}} {{class_name}} {{[-p|--perm]}} {{permission}}
```

- Search for rules with a specific target type and class:

```
sesearch --allow {{[-t|--target]}} {{type_name}} {{[-c|--class]}} {{class_name}}
```

- Display more detailed information about matched rules:

```
sesearch --allow {{[-t|--target]}} {{type_name}} {{[-v|--verbose]}}
```

sestatus

Print the current SELinux status.

More information: https://manned.org/sestatus.

- Print the current status:

```
sestatus
```

- Print the current states of all policy booleans:

```
sestatus -b
```

- Print the current file and process contexts:

```
sestatus -v
```

setarch

Change the reported architecture for a program's execution, primarily used to modify how programs behave based on system architecture.

Useful for compatibility testing or running legacy applications.

More information: https://manned.org/setarch.8.

- Run a command as if the machine architecture is `i686` (useful for running 32-bit apps on a 64-bit kernel):

```
setarch i686 {{command}}
```

- Run a shell with the `x86_64` architecture:

```
setarch x86_64 {{bash}}
```

- Disable randomization of the virtual address space:

```
setarch {{linux32}} {{[-R|--addr-no-randomize]}} {{command}}
```

- List supported architectures:

```
setarch --list
```

- Display help:

```
setarch {{[-h|--help]}}
```

setcap

Set capabilities of specified file.

See also: **`getcap`**.

More information: https://manned.org/setcap.

- Set capability `cap_net_raw` (to use RAW and PACKET sockets) for a given file:

```
setcap '{{cap_net_raw}}' {{path/to/file}}
```

- Set multiple capabilities on a file (`ep` behind the capability means "effective permitted"):

```
setcap
'{{cap_dac_read_search,cap_sys_tty_config+ep}}' {{path/to/
file}}
```

- Remove all capabilities from a file:

```
setcap -r {{path/to/file}}
```

- Verify that the specified capabilities are currently associated with the specified file:

```
setcap -v '{{cap_net_raw}}' {{path/to/file}}
```

- The optional `-n root_uid` argument can be used to set the file capability for use only in a user namespace with this root user ID owner:

```
setcap -n {{root_uid}} '{{cap_net_admin}}' {{path/to/file}}
```

setenforce

Toggle SELinux between enforcing and permissive modes.

To enable or disable SELinux, edit **`/etc/selinux/config`** instead.

See also: **`getenforce`**, **`semanage-permissive`**.

More information: https://manned.org/setenforce.

- Put SELinux in enforcing mode:

```
setenforce {{1|Enforcing}}
```

- Put SELiunx in permissive mode:

```
setenforce {{0|Permissive}}
```

setfacl

Set file access control lists (ACL).

More information: https://manned.org/setfacl.

- Modify ACL of a file for user with read and write access:

```
setfacl {{[-m|--modify]}} u:{{username}}:rw {{path/to/file_or_directory}}
```

- Modify default ACL of a file for all users:

```
setfacl {{[-d|--default]}} {{[-m|--modify]}} u::rw {{path/to/file_or_directory}}
```

- Remove ACL of a file for a user:

```
setfacl {{[-x|--remove]}} u:{{username}} {{path/to/file_or_directory}}
```

- Remove all ACL entries of a file:

```
setfacl {{[-b|--remove-all]}} {{path/to/file_or_directory}}
```

setfattr

Set extended file attributes.

More information: https://manned.org/setfattr.

- Set name of attribute for file:

```
setfattr {{[-n|--name]}} user.{{attribute_name}} {{path/to/file}}
```

- Set a user-defined value of an extended attribute on a file:

```
setfattr {{[-n|--name]}} user.{{attribute_name}} {{[-v|--value]}} "{{value}}" {{path/to/file}}
```

- Remove a specific attribute of a file:

```
setfattr {{[-x|--remove]}} user.{{attribute_name}} {{path/to/file}}
```

setfiles

Set SELinux file security contexts based on policy rules.

Similar to **`restorecon`** but reads contexts from a file_contexts file.

See also: **`restorecon`**, **`semanage-fcontext`**, **`fixfiles`**.

More information: https://manned.org/setfiles.

- Set file contexts according to the default policy file:

```
sudo setfiles /etc/selinux/targeted/contexts/files/
file_contexts {{path/to/directory}}
```

- Set file contexts recursively and show changes:

```
sudo setfiles /etc/selinux/targeted/contexts/files/
file_contexts {{path/to/directory}} {{[-v|--verbose]}}
```

- Preview what would be changed without actually modifying contexts:

```
sudo setfiles /etc/selinux/targeted/contexts/files/
file_contexts {{path/to/directory}} {{[-n|--nochange]}}
```

- Set file contexts and verify them:

```
sudo setfiles /etc/selinux/targeted/contexts/files/
file_contexts {{path/to/directory}} {{[-v|--verbose]}} {{[-
F|--force]}}
```

- Use a specific root path for context matching:

```
sudo setfiles /etc/selinux/targeted/contexts/files/
file_contexts {{path/to/new_directory}} {{[-r|--rootpath]}}
{{path/to/old_directory}}
```

setfont

Set the terminal font for TTYs.

More information: https://manned.org/setfont.

- Change the terminal font:

```
setfont {{font}}.gz
```

- Change the terminal font by specifying a path:

```
setfont /usr/share/kbd/{{mapping}}/{{font.ext}}
```

- Double font size:

```
setfont {{[-d|--double]}}
```

- Reset to the default font:

```
setfont
```

setsebool

Set SELinux boolean value.

See also: **`semanage-boolean`**, **`getsebool`**.

More information: https://manned.org/setsebool.

- Show the current setting of [a]ll booleans:

```
getsebool -a
```

- Set or unset a boolean temporarily (non-persistent across reboot):

```
sudo setsebool {{httpd_can_network_connect}} {{1|true|on|0|false|off}}
```

- Set or unset a boolean [P]ersistently:

```
sudo setsebool -P {{container_use_devices}} {{1|true|on|0|false|off}}
```

- Set or unset multiple booleans [P]ersistently at once:

```
sudo setsebool -P {{ftpd_use_fusefs=1 mount_anyfile=0 ...}}
```

- Set or unset a boolean persistently (alternative method using `semanage-boolean`):

```
sudo semanage boolean {{[-m|--modify]}} {{-1|--on|-0|--off}} {{haproxy_connect_any}}
```

setserial

Read and modify serial port information.

More information: https://manned.org/setserial.

- Print all information about a specific serial device:

```
setserial -a {{/dev/cuaN}}
```

- Print the configuration summary of a specific serial device (useful for printing during bootup process):

```
setserial -b {{device}}
```

- Set a specific configuration parameter to a device:

```
sudo setserial {{device}} {{parameter}}
```

- Print the configuration of a list of devices:

```
setserial -g {{device1 device2 ...}}
```

setsid

Run a program in a new session if the calling process is not a process group leader.

The created session is by default not controlled by the current terminal.

More information: https://manned.org/setsid.

- Run a program in a new session:

```
setsid {{program}}
```

- Run a program in a new session discarding the resulting output and error:

```
setsid {{program}} > /dev/null 2>&1
```

- Run a program creating a new process:

```
setsid {{[-f|--fork]}} {{program}}
```

- Return the exit code of a program as the exit code of setsid when the program exits:

```
setsid {{[-w|--wait]}} {{program}}
```

- Run a program in a new session setting the current terminal as the controlling terminal:

```
setsid {{[-c|--ctty]}} {{program}}
```

setxkbmap

Set the keyboard using the X Keyboard Extension.

More information: https://manned.org/setxkbmap.

- Set the keyboard in French AZERTY:

```
setxkbmap {{fr}}
```

- Set multiple keyboard layouts, their variants and switching option:

```
setxkbmap -layout {{us,de}} -variant {{,qwerty}} -option
'{{grp:alt_caps_toggle}}'
```

- List all layouts:

```
localectl list-x11-keymap-layouts
```

- List variants for the layout:

```
localectl list-x11-keymap-variants {{de}}
```

- List available switching options:

```
localectl list-x11-keymap-options | grep grp:
```

- Display help:

```
setxkbmap -help
```

sfdisk

Display or manipulate a disk partition table.

More information: https://manned.org/sfdisk.

- Back up the partition layout to a file:

```
sudo sfdisk {{[-d|--dump]}} {{/dev/sdX}} > {{path/to/file.dump}}
```

- Restore a partition layout:

```
sudo sfdisk < {{path/to/file.dump}} {{/dev/sdX}}
```

- Set the type of a partition:

```
sudo sfdisk --part-type {{/dev/sdX}} {{partition_number}} {{swap}}
```

- Delete a partition:

```
sudo sfdisk --delete {{/dev/sdX}} {{partition_number}}
```

- Display help:

```
sfdisk {{[-h|--help]}}
```

sfill

Securely overwrite the free space and inodes of the partition where the specified directory resides.

More information: https://manned.org/sfill.

- Overwrite free space and inodes of a disk with 38 writes (slow but secure):

```
sfill /{{path/to/mounted_disk_directory}}
```

- Overwrite free space and inodes of a disk with 6 writes (fast but [l]ess secure) and show status ([v]erbose):

```
sfill -l -v /{{path/to/mounted_disk_directory}}
```

- Overwrite free space and inodes of a disk with 1 write (very fast but [l]ess secure [v]erbose) and show status:

```
sfill -ll -v /{{path/to/mounted_disk_directory}}
```

- Overwrite only free space of a d[I]sk:

```
sfill -I /{{path/to/mounted_disk_directory}}
```

- Overwrite only free [i]nodes of a disk:

```
sfill -i /{{path/to/mounted_disk_directory}}
```

sh5util

Merge HDF5 files produced by the **`sacct_gather_profile`** plugin.

More information: https://slurm.schedmd.com/sh5util.html.

- Merge HDF5 files produced on each allocated node for the specified job or step:

```
sh5util {{[-j|--jobs]}} {{job_id|job_id.step_id}}
```

- Extract one or more data series from a merged job file:

```
sh5util {{[-j|--jobs]}} {{job_id|job_id.step_id}} {{[-E|--
extract]}} {{[-i|--input]}} {{path/to/file.h5}} {{[-s|--
series]}} {{Energy|Filesystem|Network|Task}}
```

- Extract one data item from all nodes in a merged job file:

```
sh5util {{[-j|--jobs]}} {{job_id|job_id.step_id}} {{[-I|--
item-extract]}} {{[-s|--series]}} {{Energy|Filesystem|
Network|Task}} {{[-d|--data]}} {{data_item}}
```

shar

Create a shell archive.

More information: https://www.gnu.org/software/sharutils/manual/sharutils.html.

- Create a shell script that when executed extracts the given files from itself:

```
shar {{[-V|--vanilla-operation]}} {{path/to/file1 path/to/
file2 ...}} > {{path/to/archive.sh}}
```

- Compress the files in the archive:

```
shar {{[-C|--compactor]}} {{xz}} {{path/to/file1 path/to/
file2 ...}} > {{path/to/archive.sh}}
```

- Treat all files as binary (i.e. `uuencode` everything):

```
shar {{[-B|--uuencode]}} {{path/to/file1 path/to/file2 ...}}
> {{path/to/archive.sh}}
```

- Treat all files as text (i.e. `uuencode` nothing):

```
shar {{[-T|--text-files]}} {{path/to/file1 path/to/
file2 ...}} > {{path/to/archive.sh}}
```

- Include a name and cut mark in the header comment of the archive:

```
shar {{[-n|--archive-name]}} "{{My files}}" {{[-c|--cut-
mark]}} {{path/to/file1 path/to/file2 ...}} > {{path/to/
archive.sh}}
```

sherlock

Find usernames across social networks.

More information: https://github.com/sherlock-project/sherlock.

- Search for a specific username on social networks saving the results to a file:

```
sherlock {{username}} --output {{path/to/file}}
```

- Search for specific usernames on social networks saving the results into a directory:

```
sherlock {{username1 username2 ...}} --folderoutput {{path/to/directory}}
```

- Search for a specific username on social networks using the Tor network:

```
sherlock --tor {{username}}
```

- Make requests over Tor with a new Tor circuit after each request:

```
sherlock --unique-tor {{username}}
```

- Search for a specific username on social networks using a proxy:

```
sherlock {{username}} --proxy {{proxy_url}}
```

- Search for a specific username on social networks and open results in the default web browser:

```
sherlock {{username}} --browse
```

- Display help:

```
sherlock --help
```

shiny-mirrors

Generate a **pacman** mirror list for Manjaro Linux.

Every run of shiny-mirrors requires you to synchronize your database and update your system using **sudo pacman -Syyu**.

More information: https://gitlab.com/Arisa_Snowbell/shiny-mirrors/-/blob/domina/shiny-mirrors/man/shiny-mirrors.md.

- Get the status of the current mirrors:

  ```
  shiny-mirrors status
  ```

- Generate a mirror list using the default behavior:

  ```
  sudo shiny-mirrors refresh
  ```

- Display the current configuration file:

  ```
  shiny-mirrors config show
  ```

- Switch to a different branch interactively:

  ```
  sudo shiny-mirrors config --branch
  ```

shnsplit

Splits audio files according to a **`.cue`** file.

More information: http://shnutils.freeshell.org/shntool/.

- Split a `.wav` + `.cue` file into multiple files:

```
shnsplit -f {{path/to/file.cue}} {{path/to/file.wav}}
```

- Show supported formats:

```
shnsplit -a
```

- Split a `.flac` file into multiple files:

```
shnsplit -f {{path/to/file.cue}} -o flac {{path/to/
file.flac}}
```

- Split a `.wav` file into files of the form "track-number - album - title":

```
shnsplit -f {{path/to/file.cue}} {{path/to/file.wav}} -t "%n
- %a - %t"
```

shntool split

This command is an alias of **`shnsplit`**.

- View documentation for the original command:

```
tldr shnsplit
```

showkey

Display the keycode of pressed keys on the keyboard, helpful for debugging keyboard-related issues and key remapping.

More information: https://manned.org/showkey.

- View keycodes in decimal:

```
sudo showkey
```

- Display scancodes in hexadecimal:

```
sudo showkey {{[-s|--scancodes]}}
```

- Display keycodes in decimal (default):

```
sudo showkey {{[-k|--keycodes]}}
```

- Display keycodes in ASCII, decimal, and hexadecimal:

```
sudo showkey {{[-a|--ascii]}}
```

- Exit the program:

```
<Ctrl d>
```

shutdown

Shutdown and reboot the system.

More information: https://manned.org/shutdown.8.

- Power off ([h]alt) immediately:

```
shutdown -h now
```

- Reboot immediately:

```
shutdown {{[-r|--reboot]}} now
```

- Reboot in 5 minutes:

```
shutdown {{[-r|--reboot]}} +{{5}} &
```

- Shutdown at 1:00 pm (Uses 24h clock):

```
shutdown -h 13:00
```

- [c]ancel a pending shutdown/reboot operation:

```
shutdown -c
```

sic

Simple IRC client.

Part of the suckless tools.

More information: https://manned.org/sic.

- Connect to the default host (irc.ofct.net) with the nickname set in the $USER environment variable:

```
sic
```

- Connect to a given host, using a given nickname:

```
sic -h {{host}} -n {{nickname}}
```

- Connect to a given host, using a given nickname and password:

```
sic -h {{host}} -n {{nickname}} -k {{password}}
```

- Join a channel:

```
:j #{{channel}}<Enter>
```

- Send a message to a channel or user:

```
:m #{{channel|user}}<Enter>
```

- Set default channel or user:

```
:s #{{channel|user}}<Enter>
```

silentcast

Silent screencast creator. Saves in `.mkv` and animated GIF formats.

More information: https://github.com/colinkeenan/silentcast.

- Launch silentcast:

```
silentcast
```

- Launch silentcast on a specific display:

```
silentcast --display={{display}}
```

sinfo

View information about Slurm nodes and partitions.

See also: **`squeue`**, **`sbatch`**, which are also part of the Slurm workload manager.

More information: https://slurm.schedmd.com/sinfo.html.

- Show a quick summary overview of the cluster:

```
sinfo {{[-s|--summarize]}}
```

- View the detailed status of all partitions across the entire cluster:

```
sinfo
```

- View the detailed status of a specific partition:

```
sinfo {{[-p|--partition]}} {{partition_name}}
```

- View information about idle nodes:

```
sinfo {{[-t|--states]}} {{idle}}
```

- Summarise dead nodes:

```
sinfo {{[-d|--dead]}}
```

- List dead nodes and the reasons why:

```
sinfo {{[-R|--list-reasons]}}
```

size

Display the sizes of sections inside binary files.

More information: https://sourceware.org/binutils/docs/binutils/size.html.

- Display the size of sections in a given object or executable file:

```
size {{path/to/file}}
```

- Display the size of sections in a given object or executable file in [o]ctal:

```
size {{[-o|--radix=8]}} {{path/to/file}}
```

- Display the size of sections in a given object or executable file in [d]ecimal:

```
size {{[-d|--radix=10]}} {{path/to/file}}
```

- Display the size of sections in a given object or executable file in he[x]adecimal:

```
size {{[-x|--radix=16]}} {{path/to/file}}
```

slapt-get

An **apt** like system for Slackware package management.

Package sources need to be configured in the slapt-getrc file.

More information: https://software.jaos.org.

- Update the list of available packages and versions:

```
slapt-get --update
```

- Install a package, or update it to the latest available version:

```
slapt-get --install {{package}}
```

- Remove a package:

```
slapt-get --remove {{package}}
```

- Upgrade all installed packages to their latest available versions:

```
slapt-get --upgrade
```

- Locate packages by the package name, disk set, or version:

```
slapt-get --search {{query}}
```

- Show information about a package:

```
slapt-get --show {{package}}
```

slapt-src

A utility to automate building of slackbuilds.

SlackBuild sources need to be configured in the slapt-srcrc file.

More information: https://github.com/jaos/slapt-src.

- Update the list of available slackbuilds and versions:

  `slapt-src {{[-u|--update]}}`

- List all available slackbuilds:

  `slapt-src {{[-l|--list]}}`

- Fetch, build and install the specified slackbuild(s):

  `slapt-src {{[-i|--install]}} {{slackbuild_name}}`

- Locate slackbuilds by their name or description:

  `slapt-src {{[-s|--search]}} {{search_term}}`

- Display information about a slackbuild:

  `slapt-src {{[-w|--show]}} {{slackbuild_name}}`

sleep

Delay for a specified amount of time.

More information: https://www.gnu.org/software/coreutils/manual/html_node/sleep-invocation.html.

- Delay in seconds:

```
sleep {{seconds}}
```

- Delay in [m]inutes. (Other units [d]ay, [h]our, [s]econd, [inf]inity can also be used):

```
sleep {{minutes}}m
```

- Delay for 1 [d]ay 3 [h]ours:

```
sleep 1d 3h
```

- Execute a specific command after 20 [m]inutes delay:

```
sleep 20m && {{command}}
```

slop

Get a selection of the screen.

More information: https://github.com/naelstrof/slop.

- Wait for the user to make a selection and output its geometry to `stdout`:

```
slop
```

- Double click, rather than click and drag, to draw a selection:

```
slop {{[-D|--nodrag]}}
```

- Highlight the selection rather than outlining it:

```
slop {{[-l|--highlight]}}
```

- Specify the output format:

```
slop {{[-f|--format]}} {{format_string}}
```

- Specify the selection rectangle's color:

```
slop {{[-c|--color]}} {{red}},{{green}},{{blue}},{{alpha}}
```

slurmctld

Monitor all other Slurm daemons and resources, accept work (jobs), and allocate resources to those jobs.

More information: https://slurm.schedmd.com/slurmctld.html.

- Clear all previous `slurmctld` states from its last checkpoint:

```
slurmctld -c
```

- Set the daemon's nice value to the specified value, typically a negative number:

```
slurmctld -n {{value}}
```

- Write log messages to the specified file:

```
slurmctld -L {{path/to/output_file}}
```

- Display help:

```
slurmctld -h
```

- Display version:

```
slurmctld -V
```

slurmd

Monitors all tasks running on the compute node, accepts tasks, launches tasks, and kills running tasks upon request.

More information: https://slurm.schedmd.com/slurmd.html.

- Report node rebooted when daemon restarted (Used for testing purposes):

```
slurmd -b
```

- Run the daemon with the given nodename:

```
slurmd -N {{nodename}}
```

- Write log messages to the specified file:

```
slurmd -L {{path/to/output_file}}
```

- Read configuration from the specified file:

```
slurmd -f {{path/to/file}}
```

- Display help:

```
slurmd -h
```

slurmdbd

A secure enterprise-wide interface to a database for Slurm.

More information: https://slurm.schedmd.com/slurmdbd.html.

- Set the daemon's nice value to the specified value, typically a negative number:

```
slurmdbd -n {{value}}
```

- Change the working directory of `slurmdbd` to the LogFile path or to `/var/tmp`:

```
slurmdbd -s
```

- Display help:

```
slurmdbd -h
```

- Display version:

```
slurmdbd -V
```

slurmrestd

Interface to Slurm via REST API. It can be used in two modes: Inetd Mode & Listen Mode.

More information: https://slurm.schedmd.com/slurmrestd.html.

- Change the [g]roup ID (and drop supplemental groups) before processing client requests:

```
slurmrestd -g {{group_id}} {{[host]:port|unix:/path/to/socket}}
```

- Comma-delimited list of [a]uthentication plugins to load:

```
slurmrestd -a {{authentication_plugins}} {{[host]:port|unix:/path/to/socket}}
```

- Read Slurm configuration from the specified [f]ile:

```
slurmrestd -f {{path/to/file}}
```

- Change [u]ser ID before processing client request:

```
slurmrestd -u {{user_id}}
```

- Display [h]elp:

```
slurmrestd -h
```

- Display [V]ersion:

```
slurmrestd -V
```

slurmstepd

Slurm daemon for managing and monitoring individual job steps within a multi-step job.

It should not be invoked manually.

More information: https://slurm.schedmd.com/slurmstepd.html.

- Start the daemon:

```
slurmstepd
```

slurp

Select a region in a Wayland compositor.

More information: https://github.com/emersion/slurp/blob/master/slurp.1.scd.

- Select a region and print it to `stdout`:

```
slurp
```

- Select a region and print it to `stdout`, while displaying the dimensions of the selection:

```
slurp -d
```

- Select a single point instead of a region:

```
slurp -p
```

- Select an output and print its name:

```
slurp -o -f '%o'
```

- Select a specific region and take a borderless screenshot of it, using `grim`:

```
grim -g "$(slurp -w 0)"
```

- Select a specific region and take a borderless video of it, using `wf-recorder`:

```
wf-recorder {{[-g|--geometry]}} "$(slurp -w 0)"
```

sm

Display a short message fullscreen.

More information: https://github.com/nomeata/screen-message.

- Display a message in full-screen:

```
sm "{{Hello World!}}"
```

- Display a message with inverted colors:

```
sm {{[-i|--invert]}} "{{Hello World!}}"
```

- Display a message with a custom foreground color:

```
sm {{[-f|--foreground]}} {{blue}} "{{Hello World!}}"
```

- Display a message with a custom background color:

```
sm {{[-b|--background]}} {{#008888}} "{{Hello World!}}"
```

- Display a message rotated 3 times (in steps of 90 degrees, counterclockwise):

```
sm {{[-r|--rotate]}} {{3}} "{{Hello World!}}"
```

- Display a message using the output from another command:

```
{{echo "Hello World!"}} | sm -
```

smbcacls

View and manipulate Windows ACLs on SMB shares.

Part of the Samba suite.

More information: https://www.samba.org/samba/docs/current/man-html/smbcacls.1.html.

- Display the ACLs for a file or directory on a remote SMB share:

```
smbcacls //{{server}}/{{share}} {{path/to/file_or_directory}} --user {{domain\\username}}%{{password}}
```

- Set a new ACL for a file on a remote SMB share (replace `"ACL:..."` with a valid Windows ACL specification):

```
smbcacls //{{server}}/{{share}} {{path/to/file}} --user {{domain\\username}}%{{password}} "ACL:{{DACL}}"
```

- Remove all existing ACL entries and set a new ACL:

```
smbcacls //{{server}}/{{share}} {{path/to/file}} --user {{domain\\username}}%{{password}} "RESET" "ACL:{{DACL}}"
```

- Specify an alternative workgroup (or domain) and have the program prompt for a password interactively:

```
smbcacls //{{server}}/{{share}} {{path/to/file}} --user {{username}} --workgroup {{workgroup}}
```

smbclient

FTP-like client to access SMB/CIFS resources on servers.

More information: https://manned.org/smbclient.

- Connect to a share (user will be prompted for password; `exit` to quit the session):

```
smbclient {{//server/share}}
```

- Connect with a different username:

```
smbclient {{//server/share}} --user {{username}}
```

- Connect with a different workgroup:

```
smbclient {{//server/share}} --workgroup {{domain}} --user
{{username}}
```

- Connect with a username and password:

```
smbclient {{//server/share}} --user {{username%password}}
```

- Download a file from the server:

```
smbclient {{//server/share}} --directory {{path/to/
directory}} --command "get {{file.txt}}"
```

- Upload a file to the server:

```
smbclient {{//server/share}} --directory {{path/to/
directory}} --command "put {{file.txt}}"
```

- List the shares from a server anonymously:

```
smbclient --list={{server}} --no-pass
```

smbget

wget-like utility for downloading files from SMB servers.

More information: https://www.samba.org/samba/docs/current/man-html/smbget.1.html.

- Download a file from a server:

```
smbget {{smb://server/share/file}}
```

- Download a share or directory recursively:

```
smbget --recursive {{smb://server/share}}
```

- Connect with a username and password:

```
smbget {{smb://server/share/file}} {{[-U|--user]}} {{username%password}}
```

- Require encrypted transfers:

```
smbget {{smb://server/share/file}} {{[-e|--encrypt]}}
```

smbnetfs

Mount SMB shares interactively.

More information: https://sourceforge.net/projects/smbnetfs/.

- Make shares available at `mountpoint`:

```
smbnetfs {{mountpoint}}
```

smbpasswd

Add/remove a Samba user or change its password.

Samba users must have an existing local Unix account.

More information: https://manned.org/smbpasswd.8.

- Change the current user's SMB password:

  ```
  smbpasswd
  ```

- Add a specified user to Samba and set password (user should already exist in system):

  ```
  sudo smbpasswd -a {{username}}
  ```

- Modify an existing Samba user's password:

  ```
  sudo smbpasswd {{username}}
  ```

- Delete a Samba user (use `pdbedit` instead if the Unix account has been deleted):

  ```
  sudo smbpasswd -x {{username}}
  ```

smem

Print memory usage for programs.

More information: https://manned.org/smem.

- Print memory usage for current processes:

```
smem
```

- Print memory usage for current processes for a every user on a system:

```
smem --users
```

- Print memory usage for current processes for a specified user:

```
smem --userfilter {{username}}
```

- Print system memory information:

```
smem --system
```

snake4

Snake game in the terminal.

More information: https://manned.org/snake4.6.

- Start a snake game:

```
snake4
```

- Choose level:

```
{{<1>|<2>|<3>|<4>|<5>}}
```

- Navigate the snake:

```
{{<ArrowUp>|<ArrowDown>|<ArrowLeft>|<ArrowRight>}}
```

- Pause game:

```
<Space>
```

- Quit game:

```
<q>
```

- Show the high scores:

```
snake4 --highscores
```

snake4scores

Show the high scores from the snake4 game.

More information: https://manpages.debian.org/snake4/snake4.6.en.html.

- Show the highscores:

```
snake4scores
```

snap

Sentinel Application Platform (SNAP) for processing satellite data from the European Space Agency (ESA).

More information: https://step.esa.int/main/download/snap-download/.

- Display all updates:

```
snap --nosplash --nogui --modules --list --refresh
```

- Display help:

```
snap --help
```

snap

snap can refer to multiple commands with the same name.

- View documentation for the packaging solution:

  ```
  tldr snap.pkg
  ```

- View documentation for the European Space Agency Earth observation data processing tool:

  ```
  tldr snap.esa
  ```

snap

Manage the "snap" self-contained software packages.

Similar to what **`apt`** is for **`.deb`**.

More information: https://manned.org/snap.

- Search for a package:

```
snap find {{query}}
```

- Install a package:

```
snap install {{package}}
```

- Update a package:

```
snap refresh {{package}}
```

- Update a package to another channel (track, risk, or branch):

```
snap refresh {{package}} --channel={{channel}}
```

- Update all packages:

```
snap refresh
```

- Display basic information about installed snap software:

```
snap list
```

- Uninstall a package:

```
snap remove {{package}}
```

- Check for recent snap changes in the system:

```
snap changes
```

snapper

Filesystem snapshot management tool.

More information: http://snapper.io/manpages/snapper.html.

- List snapshot configs:

```
snapper list-configs
```

- Create snapper config:

```
snapper {{[-c|--config]}} {{config}} create-config {{path/to/directory}}
```

- Create a snapshot with a description:

```
snapper {{[-c|--config]}} {{config}} create {{[-d|--description]}} "{{snapshot_description}}"
```

- List snapshots for a config:

```
snapper {{[-c|--config]}} {{config}} list
```

- Delete a snapshot:

```
snapper {{[-c|--config]}} {{config}} delete {{snapshot_number}}
```

- Delete a range of snapshots:

```
snapper {{[-c|--config]}} {{config}} delete {{snapshot1}}-{{snapshot2}}
```

sockstat

List open Internet or UNIX domain sockets.

See also: **`netstat`**.

More information: https://manned.org/sockstat.

- Show information for IPv4 and IPv6 sockets for both listening and connected sockets:

```
sockstat
```

- Show information for IPv[4]/IPv[6] sockets [l]istening on specific [p]orts using a specific p[R]otocol:

```
sockstat -{{4|6}} -l -R {{tcp|udp|raw|unix}} -p {{port1,port2...}}
```

- Also show [c]onnected sockets and [u]nix sockets:

```
sockstat -cu
```

- Only show sockets of the specified `pid` or process:

```
sockstat -P {{pid|process}}
```

- Only show sockets of the specified `uid` or user:

```
sockstat -U {{uid|user}}
```

- Only show sockets of the specified `gid` or group:

```
sockstat -G {{gid|group}}
```

spa-acp-tool

Debug ALSA card profile probing without running PipeWire.

More information: https://docs.pipewire.org/page_man_spa-acp-tool_1.html.

- List all ALSA objects:

```
spa-acp-tool {{[l|list]}}
```

- Probe a specific ALSA card by ID:

```
spa-acp-tool {{[c|card]}} {{card_id}}
```

- List available ALSA profiles for a card:

```
spa-acp-tool {{[lpr|list-profiles]}} {{card_id}}
```

- Set the active ALSA profile by ID:

```
spa-acp-tool {{[spr|set-profile]}} {{profile_id}}
```

- List available devices for a card:

```
spa-acp-tool {{[ld|list-devices]}} {{card_id}}
```

- Get volume from a device:

```
spa-acp-tool {{[gv|get-volume]}} {{device_id}}
```

- Set volume on a device:

```
spa-acp-tool {{[v|set-volume]}} {{device_id}}
{{volume_level}}
```

- Toggle mute state on a device:

```
spa-acp-tool {{[m|toggle-mute]}} {{device_id}}
```

spa-inspect

Display SPA plugin information, list and try to instantiate available factories.

More information: https://docs.pipewire.org/page_man_spa-inspect_1.html.

- Display information about a SPA plugin:

```
spa-inspect {{path/to/plugin.so}}
```

speaker-test

Speaker test tone generator for ALSA.

See also: **`aplay`**, **`arecord`**, **`amixer`**.

More information: https://manned.org/speaker-test.

- Test the default speakers with pink noise:

```
speaker-test
```

- Test the default speakers with a sine wave:

```
speaker-test {{[-t|--test]}} sine {{[-f|--frequency]}}
{{frequency}}
```

- Test the default speakers with a predefined WAV file:

```
speaker-test {{[-t|--test]}} wav
```

- Test the default speakers with a WAV file:

```
speaker-test {{[-t|--test]}} wav {{[-w|--wavfile]}} {{path/
to/file}}
```

spectre-meltdown-checker

Spectre and Meltdown mitigation detection tool.

More information: https://manned.org/spectre-meltdown-checker.

- Check the currently running kernel for Spectre or Meltdown:

```
sudo spectre-meltdown-checker
```

- Check the currently running kernel and show an explanation of the actions to take to mitigate a vulnerability:

```
sudo spectre-meltdown-checker --explain
```

- Check for specific variants (defaults to all):

```
sudo spectre-meltdown-checker --variant {{1|2|3|3a|4|l1tf|
msbds|mfbds|mlpds|mdsum|taa|mcespc|srbds}}
```

- Display output using a specific output format:

```
sudo spectre-meltdown-checker --batch {{text|json|nrpe|
prometheus|short}}
```

- Don't use the `/sys` interface even if present:

```
sudo spectre-meltdown-checker --no-sysfs
```

- Check a non-running kernel:

```
sudo spectre-meltdown-checker --kernel {{path/to/
kernel_file}}
```

speedometer

Python script that shows a network traffic graph in the terminal.

More information: https://excess.org/speedometer.

- Show graph for a specific interface:

```
speedometer -r {{eth0}} -t {{eth0}}
```

speedread

A simple terminal-based open source Spritz-alike.

Shows input text as a per-word RSVP (rapid serial visual presentation) aligned on optimal reading points, which allows reading text at a much more rapid pace than usual as the eye can stay fixed on a single place.

More information: https://github.com/pasky/speedread.

- Read a text file at a specific speed:

```
cat {{path/to/file.txt}} | speedread -wpm {{250}}
```

- Resume from a specific line:

```
cat {{path/to/file.txt}} | speedread -resume {{5}}
```

- Show multiple words at a time:

```
cat {{path/to/file.txt}} | speedread -multiword
```

- Slow down by 10% during the reading session:

```
<[>
```

- Speed up by 10% during the reading session:

```
<]>
```

- Pause, and show the last few lines as context:

```
<Space>
```

spi

A meta package manager that handles both packages and slackbuilds.

More information: https://github.com/gapan/spi/blob/master/man/spi.t2t.

- Update the list of available packages and slackbuilds:

```
spi {{[-u|--update]}}
```

- Install a package or slackbuild:

```
spi {{[-i|--install]}} {{package/slackbuild_name}}
```

- Upgrade all installed packages to the latest versions available:

```
spi {{[-U|--upgrade]}}
```

- Locate packages or slackbuilds by package name or description:

```
spi {{search_terms}}
```

- Display information about a package or slackbuild:

```
spi --show {{package/slackbuild_name}}
```

- Purge the local package and slackbuild caches:

```
spi --clean
```

sport

Search and install SlackBuilds.

More information: http://slackermedia.info/handbook/doku.php?id=slackbuilds.

- Pull the list of SlackBuilds to run `sport` for the first time:

```
sudo mkdir {{[-p|--parents]}} /usr/ports && sudo rsync {{[-av|--archive --verbose]}} rsync://slackbuilds.org /slackbuilds/$(awk '{print $2}' /etc/slackware-version)/ /usr/ports/
```

- Pull in any updates to the system's tree via `rsync`:

```
sudo sport rsync
```

- Search for a package by name:

```
sport search "{{keyword}}"
```

- Check if a package is installed:

```
sport check {{package}}
```

- Display README and `.info` files of a package:

```
sport cat {{package}}
```

- Install a package once the dependencies are resolved:

```
sudo sport install {{package}}
```

- Install a list of packages from a file (format: packages separated by spaces):

```
sudo sport install $(< {{path/to/list}})
```

sprio

View the factors determining a job's scheduling priority.

More information: https://slurm.schedmd.com/sprio.html.

- View the factors determining the scheduling priority of all jobs:

```
sprio
```

- View the factors determining the specified job's scheduling priority:

```
sprio {{[-j|--jobs]}} {{job_id_1,job_id_2,...}}
```

- Output additional information:

```
sprio {{[-l|--long]}}
```

- View information for the jobs of specified users:

```
sprio {{[-u|--user]}} {{user_name_1,user_name_2,...}}
```

- Print the weights for each factor determining job scheduling priority:

```
sprio {{[-w|--weights]}}
```

sqfscat

Concatenate files from a squashfs filesystem and print them to **`stdout`**.

More information: https://manned.org/sqfscat.

- Display the contents of one or more files from a squashfs filesystem:

```
sqfscat {{filesystem.squashfs}} {{file1 file2 ...}}
```

sqfstar

Create a squashfs filesystem from a tar archive.

More information: https://manned.org/sqfstar.

- Create a squashfs filesystem (compressed using `gzip` by default) from an uncompressed tar archive:

```
sqfstar < {{archive.tar}} {{filesystem.squashfs}}
```

- Create a squashfs filesystem from a tar archive compressed with `gzip`, and [comp]ress the filesystem using a specific algorithm:

```
zcat {{archive.tar.gz}} | sqfstar -comp {{gzip|lzo|lz4|xz|zstd|lzma}} {{filesystem.squashfs}}
```

- Create a squashfs filesystem from a tar archive compressed with `xz`, excluding some of the files:

```
xzcat {{archive.tar.xz}} | sqfstar {{filesystem.squashfs}} {{file1 file2 ...}}
```

- Create a squashfs filesystem from a tar archive compressed with `zstd`, excluding files ending with `.gz`:

```
zstdcat {{archive.tar.zst}} | sqfstar {{filesystem.squashfs}} "{{*.gz}}"
```

- Create a squashfs filesystem from a tar archive compressed with `lz4`, excluding files matching a `regex`:

```
lz4cat {{archive.tar.lz4}} | sqfstar {{filesystem.squashfs}} -regex "{{regex}}"
```

squeue

View the jobs queued in the SLURM scheduler.

More information: https://manned.org/squeue.

- View the queue:

```
squeue
```

- View jobs queued by a specific user:

```
squeue {{[-u|--user]}} {{username}}
```

- View the queue and refresh every 5 seconds:

```
squeue {{[-i|--iterate]}} {{5}}
```

- View the queue with expected start times:

```
squeue --start
```

sreport

Generate reports on jobs, users, and clusters from accounting data.

More information: https://slurm.schedmd.com/sreport.html.

- Show pipe delimited cluster utilization data:

```
sreport {{[-p|--parsable]}} cluster utilization
```

- Show number of jobs run:

```
sreport job sizes printjobcount
```

- Show users with the highest CPU time use:

```
sreport user topuser
```

srun

Create an interactive slurm job or connect to an existing job.

More information: https://slurm.schedmd.com/srun.html.

- Submit a basic interactive job:

```
srun --pty /bin/bash
```

- Submit an interactive job with different attributes:

```
srun --ntasks-per-node {{num_cores}} --mem-per-cpu
{{memory_MB}} --pty /bin/bash
```

- Connect to a worker node with a job running:

```
srun --jobid {{job_id}} --pty /bin/bash
```

ss

Utility to investigate sockets.

More information: https://manned.org/ss.8.

- Show all TCP/UDP/RAW/UNIX sockets:

```
ss {{[-a|--all]}} {{-t|-u|-w|-x}}
```

- Filter TCP sockets by states, only/exclude:

```
ss {{state|exclude}} {{bucket|big|connected|
synchronized|...}}
```

- Show all TCP sockets connected to the local HTTPS port (443):

```
ss {{[-t|--tcp]}} src :{{443}}
```

- Show all TCP sockets listening on the local 8080 port:

```
ss {{[-lt|--listening --tcp]}} src :{{8080}}
```

- Show all TCP sockets along with processes connected to a remote SSH port:

```
ss {{[-pt|--processes --tcp]}} dst :{{ssh}}
```

- Show all UDP sockets connected on specific source and destination ports:

```
ss {{[-u|--udp]}} 'sport == :{{source_port}} and dport == :
{{destination_port}}'
```

- Show all TCP IPv4 sockets locally connected on the subnet 192.168.0.0/16:

```
ss {{[-4t|--ipv4 --tcp]}} src {{192.168/16}}
```

- Kill IPv4 or IPv6 Socket Connection with a specific destination IP and port:

```
ss {{[-K|--kill]}} dst {{ip_address}} dport = {{port}}
```

sshare

List the shares of associations to a cluster.

More information: https://slurm.schedmd.com/sshare.html.

- List Slurm share information:

```
sshare
```

- Control the output format:

```
sshare --{{parsable|parsable2|json|yaml}}
```

- Control the fields to display:

```
sshare {{[-o|--format]}} {{format_string}}
```

- Display information for the specified users only:

```
sshare {{[-u|--users]}} {{user_id_1,user_id_2,...}}
```

sslstrip

Perform Moxie Marlinspike's Secure Sockets Layer (SSL) stripping attacks.

Perform an ARP spoofing attack in conjunction.

More information: https://www.kali.org/tools/sslstrip/.

- Log only HTTPS POST traffic on port 10000 by default:

```
sslstrip
```

- Log only HTTPS POST traffic on port 8080:

```
sslstrip --listen={{8080}}
```

- Log all SSL traffic to and from the server on port 8080:

```
sslstrip --ssl --listen={{8080}}
```

- Log all SSL and HTTP traffic to and from the server on port 8080:

```
sslstrip --listen={{8080}} --all
```

- Specify the file path to store the logs:

```
sslstrip --listen={{8080}} --write={{path/to/file}}
```

- Display help:

```
sslstrip --help
```

sstat

View information about running jobs.

More information: https://slurm.schedmd.com/sstat.html.

- Display status information of a comma-separated list of jobs:

```
sstat {{[-j|--jobs]}} {{job_id}}
```

- Display job ID, average CPU and average virtual memory size of a comma-separated list of jobs, with pipes as column delimiters:

```
sstat {{[-p|--parsable]}} {{[-j|--jobs]}} {{job_id}} {{[-o|--format]}} {{JobID,AveCPU,AveVMSize}}
```

- Display list of fields available:

```
sstat {{[-e|--helpformat]}}
```

st

A simple terminal emulator for the X Window System.

More information: https://manned.org/st.

- Open a terminal:

```
st
```

- Open a terminal with a specific title:

```
st -T {{title}}
```

- Open a terminal, execute a given command, and write the output to a file:

```
st -o {{path/to/file}} -e {{command argument1 argument2}}
```

- Increase/decrease the font size:

```
<Ctrl Shift {{PageUp|PageDown}}>
```

- Copy/paste from the clipboard:

```
<Ctrl Shift {{c|v}}>
```

st

st can refer to multiple commands.

- View documentation for the terminal emulator:

```
tldr st.1
```

- View documentation for the statistics tool:

```
tldr st.2
```

startx

A front-end to **`xinit`** that provides a nice user interface for running a single session of the X Window System.

More information: https://x.org/releases/X11R7.5/doc/man/man1/startx.1.html.

- Start an X session:

```
startx
```

- Start an X session with a predefined depth value:

```
startx -- -depth {{value}}
```

- Start an X session with a predefined dpi value:

```
startx -- -dpi {{value}}
```

- Override the settings in the `.xinitrc` file and start a new X session:

```
startx /{{path/to/window_manager_or_desktop_environment}}
```

steamos-add-to-steam

Add a program to Steam library.

More information: https://gitlab.com/users/evlaV/projects.

- Add a program to Steam library:

```
steamos-add-to-steam {{path/to/file}}
```

steamos-boot-install

This command is an alias of **`steamos-finalize-install`**.

- View documentation for the original command:

  `tldr steamos-finalize-install`

steamos-chroot

Switch root directory in a SteamOS environment.

More information: https://gitlab.com/users/evlaV/projects.

- Switch to the other A/B partition:

```
steamos-chroot {{[-p|--partset]}} other
```

- Switch to a partition on another drive:

```
steamos-chroot {{[-d|--disk]}} {{/dev/sdX}} {{[-p|--
partset]}} {{A|B}}
```

- Display help:

```
steamos-chroot {{[-h|--help]}}
```

steamos-devmode

Enable SteamOS developer mode.

Note: Developer mode disables read-only protections and initializes the package manager.

More information: https://gitlab.com/users/evlaV/projects.

- Enable developer mode:

```
sudo steamos-devmode enable
```

- Enable developer mode without confirmation prompt:

```
sudo steamos-devmode enable --no-prompt
```

- Check if developer mode is currently enabled:

```
sudo steamos-devmode status
```

steamos-dump-info

View SteamOS system information.

More information: https://gitlab.com/users/evlaV/projects.

- View SteamOS system information:

```
sudo steamos-dump-info
```

steamos-factory-reset-config

Generate configuration files required by **`steamos-factory-reset`**.

This command is typically called internally and not meant to be run manually.

More information: https://gitlab.com/users/evlaV/projects.

- Generate factory reset configuration data:

  `steamos-factory-reset-config`

steamos-factory-reset

Reset SteamOS to factory settings.

More information: https://gitlab.com/users/evlaV/projects.

- Begin a factory reset (requires confirmation):

```
steamos-factory-reset
```

- Display help:

```
steamos-factory-reset --help
```

steamos-finalize-install

Complete a SteamOS installation by setting up bootloaders and applying system updates.

More information: https://gitlab.com/users/evlaV/projects.

- Finalize the installation:

```
sudo steamos-finalize-install
```

- Finalize without updating bootloaders or kernel:

```
sudo steamos-finalize-install --no-bootloaders --no-kernel
```

- Skip all migration steps:

```
sudo steamos-finalize-install --no-migrate
```

- Set a specific root hash during finalization:

```
sudo steamos-finalize-install --roothash {{hash}}
```

- Force system migration steps regardless of environment:

```
sudo steamos-finalize-install --force
```

steamos-readonly

Set the readonly status of the filesystem.

More information: https://gitlab.com/users/evlaV/projects.

- Set the filesystem to be mutable:

```
sudo steamos-readonly disable
```

- Set the filesystem to be read only:

```
sudo steamos-readonly enable
```

steamos-select-branch

Select a SteamOS system release branch.

More information: https://gitlab.com/users/evlaV/projects.

- View the current branch:

  ```
  steamos-select-branch -c
  ```

- List all available branches:

  ```
  steamos-select-branch -l
  ```

- Switch to the beta branch:

  ```
  steamos-select-branch beta
  ```

- Switch back to the stable release:

  ```
  steamos-select-branch stable
  ```

- Switch to the bleeding edge branch:

  ```
  steamos-select-branch main
  ```

steamos-session-select

Manipulate which session is currently in use.

More information: https://gitlab.com/users/evlaV/projects.

- Change to desktop mode:

```
steamos-session-select plasma
```

- Change to gamemode (sets the system to boot into gamemode if `-persistent` options were selected previously):

```
steamos-session-select
```

- Change to Wayland desktop mode:

```
steamos-session-select plasma-wayland
```

- Change to Wayland desktop mode and have the device boot to desktop:

```
steamos-session-select plasma-wayland-persistent
```

- Change to X11 desktop mode and have the device boot to desktop:

```
steamos-session-select plasma-x11-persistent
```

steamos-update

Update SteamOS.

More information: https://gitlab.com/users/evlaV/projects.

- Update the operating system:

```
steamos-update
```

- Check if there is an update available:

```
steamos-update check
```

steghide

Steganography tool for JPEG, BMP, WAV and AU file formats.

More information: https://manned.org/steghide.

- Embed data in a PNG, prompting for a passphrase:

```
steghide embed {{[-cf|--coverfile]}} {{path/to/image.png}}
{{[-ef|--embedfile]}} {{path/to/data.txt}}
```

- Extract data from a WAV audio file:

```
steghide extract {{[-sf|--stegofile]}} {{path/to/sound.wav}}
```

- Display file information, trying to detect an embedded file:

```
steghide info {{path/to/file.jpg}}
```

- Embed data in a JPEG image, using maximum compression:

```
steghide embed {{[-cf|--coverfile]}} {{path/to/image.jpg}}
{{[-ef|--embedfile]}} {{path/to/data.txt}} {{[-z|--
compress]}} {{9}}
```

- Get the list of supported encryption algorithms and modes:

```
steghide encinfo
```

- Embed encrypted data in a JPEG image, e.g. with Blowfish in CBC mode:

```
steghide embed {{[-cf|--coverfile]}} {{path/to/image.jpg}}
{{[-ef|--embedfile]}} {{path/to/data.txt}} {{[-e|--
encryption]}} {{blowfish|...}} {{cbc|...}}
```

stegsnow

Steganography tool for concealing and extracting messages in text files encoded as tabs and spaces.

More information: https://darkside.com.au/snow/manual.html.

- Extract [m]essage from file:

```
stegsnow {{path/to/file.txt}}
```

- Extract [C]ompressed and [p]assword protected [m]essage from file:

```
stegsnow -C -p {{password}} {{path/to/file.txt}}
```

- Determine approximate [S]torage capacity with line [l]ength less than 72 for file:

```
stegsnow -S -l 72 {{path/to/file.txt}}
```

- Conceal [m]essage in text from file and save to result:

```
stegsnow -m '{{message}}' {{path/to/file.txt}} {{path/to/result.txt}}
```

- Conceal message [f]ile content [C]ompressed in text from file and save to result:

```
stegsnow -C -f '{{path/to/message.txt}}' {{path/to/file.txt}} {{path/to/result.txt}}
```

- Conceal [m]essage [C]ompressed and [p]assword protected in text from file and save to result:

```
stegsnow -C -p {{password}} -m '{{message}}' {{path/to/file.txt}} {{path/to/result.txt}}
```

strace

Troubleshooting tool for tracing system calls.

More information: https://manned.org/strace.

- Start tracing a specific process by its PID:

```
sudo strace {{[-p|--attach]}} {{pid}}
```

- Trace a process and filter output by system call [e]xpression:

```
sudo strace {{[-p|--attach]}} {{pid}} -e
{{system_call,system_call2,...}}
```

- Count time, calls, and errors for each system call and report a summary on program exit:

```
sudo strace {{[-p|--attach]}} {{pid}} {{[-c|--summary-only]}}
```

- Show the time spent in every system call and specify the maximum string size to print:

```
sudo strace {{[-p|--attach]}} {{pid}} {{[-T|--syscall-
times]}} {{[-s|--string-limit]}} {{32}}
```

- Start tracing a program by executing it:

```
strace {{program}}
```

- Start tracing file operations of a program:

```
strace -e trace=file {{program}}
```

- Start tracing network operations of a program as well as all its forked and child processes, saving the output to a file:

```
strace {{[-f|--follow-forks]}} -e trace=network {{[-o|--
output]}} {{trace.txt}} {{program}}
```

stratis

Manage local storage pools and volumes using the Stratis storage manager.

Stratis volumes use the XFS filesystem and require the **`stratisd`** service.

More information: https://stratis-storage.github.io.

- Start the Stratis service (must be active before managing pools or volumes):

```
sudo systemctl start stratisd
```

- Create a storage pool from one or more devices:

```
sudo stratis pool create {{pool_name}} {{/dev/sdX}} {{/dev/
sdY}}
```

- Create a filesystem (volume) in a pool:

```
sudo stratis filesystem create {{pool_name}} {{volume_name}}
```

- List all Stratis filesystems:

```
sudo stratis filesystem list
```

- Format and mount a Stratis volume manually:

```
sudo mkfs.xfs /dev/stratis/{{pool_name}}/{{volume_name}} &&
sudo mount /dev/stratis/{{pool_name}}/{{volume_name}} {{/mnt/
target}}
```

- Add a volume to `/etc/fstab` for mounting at boot:

```
echo /dev/stratis/{{pool_name}}/{{volume_name}} {{/mnt/
target}} xfs defaults,x-systemd.requires=stratisd.service 0 0
| sudo tee {{[-a|--append]}} /etc/fstab
```

- Extend an existing pool by adding a new device:

```
sudo stratis pool add-data {{pool_name}} {{/dev/sdZ}}
```

- Delete a volume:

```
sudo stratis filesystem destroy {{pool_name}} {{volume_name}}
```

stress

Stress test CPU, memory, and IO on a Linux system.

More information: https://manned.org/stress.

- Spawn 4 workers to stress test CPU:

```
stress {{[-c|--cpu]}} {{4}}
```

- Spawn 2 workers to stress test IO and timeout after 5 seconds:

```
stress {{[-i|--io]}} {{2}} {{[-t|--timeout]}} {{5}}
```

- Spawn 2 workers to stress test memory (each worker allocates 256M bytes):

```
stress {{[-m|--vm]}} {{2}} --vm-bytes {{256M}}
```

- Spawn 2 workers spinning on write()/unlink() (each worker writes 1G bytes):

```
stress {{[-d|--hdd]}} {{2}} --hdd-bytes {{1GB}}
```

strigger

View or modify Slurm trigger information.

Triggers are actions that are automatically run when an event occurs on a Slurm cluster.

More information: https://slurm.schedmd.com/strigger.html.

- Register a new trigger. Execute the specified program when the specified event occurs:

```
strigger --set --{{primary_database_failure|
primary_slurmdbd_failure|primary_slurmctld_acct_buffer_full|
primary_slurmctld_failure|...}} {{[-p|--program]}} {{path/to/
executable}}
```

- Execute the specified program when the specified job terminated:

```
strigger --set {{[-j|--jobid]}} {{job_id}} {{[-f|--fini]}}
{{[-p|--program]}} "{{path/to/executable}} {{argument1
argument2 ...}}"
```

- View active triggers:

```
strigger --get
```

- View active triggers regarding the specified job:

```
strigger --get {{[-j|--jobid]}} {{job_id}}
```

- Clear the specified trigger:

```
strigger --clear {{trigger_id}}
```

strip

Discard symbols from executables or object files.

More information: https://manned.org/strip.

- Replace the input file with its stripped version:

```
strip {{path/to/file}}
```

- Strip symbols from a file, saving the output to a specific file:

```
strip {{path/to/input_file}} -o {{path/to/output_file}}
```

- Strip debug symbols only:

```
strip {{[-d|--strip-debug]}} {{path/to/file.o}}
```

su

Switch shell to another user.

More information: https://manned.org/su.

- Switch to superuser (requires the root password):

```
su
```

- Switch to a given user (requires the user's password):

```
su {{username}}
```

- Switch to a given user and simulate a full login shell:

```
su - {{username}}
```

- Execute a command as another user:

```
su - {{username}} {{[-c|--command]}} "{{command}}"
```

sulogin

Log in as root during single-user mode.

More information: https://manned.org/man/sulogin.

- Start `sulogin` on the default console:

```
sudo sulogin
```

- Start `sulogin` on a specific TTY device:

```
sudo sulogin {{/dev/ttyX}}
```

- Set a maximum timeout(in seconds) for entering the root password before continuing normal boot:

```
sudo sulogin {{[-t|--timeout]}} {{timeout}}
```

- Start root's shell as a login shell:

```
sudo sulogin {{[-p|--login-shell]}}
```

- Force a root shell without asking for a password when default methods of obtaining the password fail:

```
sudo sulogin {{[-e|--force]}}
```

sview

Start a GUI to view and modify the state of Slurm.

More information: https://slurm.schedmd.com/sview.html.

- Start a GUI to view and modify the state of Slurm:

```
sview
```

swaks

Swiss Army Knife SMTP, the all-purpose SMTP transaction tester.

More information: https://github.com/jetmore/swaks/blob/develop/doc/base.pod.

- Deliver a standard test email to `user@example.com` on port 25 of `test-server.example.net`:

```
swaks {{[-t|--to]}} {{user@example.com}} {{[-s|--server]}} {{test-server.example.net}}
```

- Deliver a standard test email, requiring CRAM-MD5 authentication as user `me@example.com`. An "X-Test" header will be added to the email body:

```
swaks {{[-t|--to]}} {{user@example.com}} {{[-f|--from]}} {{me@example.com}} {{[-a|--auth]}} {{CRAM-MD5}} {{[-au|--auth-user]}} {{me@example.com}} --header-X-Test "{{test_email}}"
```

- Test a virus scanner using EICAR in an attachment. Don't show the message DATA part:

```
swaks {{[-t|--to]}} {{user@example.com}} --attach - {{[-s|--server]}} {{test-server.example.com}} {{[-n|--suppress-data]}} {{path/to/eicar.txt}}
```

- Test a spam scanner using GTUBE in the body of an email, routed via the MX records for `example.com`:

```
swaks {{[-t|--to]}} {{user@example.com}} --body {{path/to/gtube_file}}
```

- Deliver a standard test email to `user@example.com` using the LMTP protocol via a UNIX domain socket file:

```
swaks {{[-t|--to]}} {{user@example.com}} --socket {{/var/lda.sock}} --protocol {{LMTP}}
```

swaplabel

Print or change the label or UUID of a swap area.

Note: **`path/to/file`** can either point to a regular file or a swap partition.

More information: https://manned.org/swaplabel.

- Display the current label and UUID of a swap area:

  `swaplabel {{path/to/file}}`

- Set the label of a swap area:

  `swaplabel {{[-L|--label]}} {{new_label}} {{path/to/file}}`

- Set the UUID of a swap area (you can generate a UUID using `uuidgen`):

  `swaplabel {{[-U|--uuid]}} {{new_uuid}} {{path/to/file}}`

swapoff

Disable devices and files for swapping.

Note: **`path/to/file`** can either point to a regular file or a swap partition.

More information: https://manned.org/swapoff.8.

- Disable a given swap area:

```
sudo swapoff {{path/to/file}}
```

- Disable all swap areas in `/proc/swaps`:

```
sudo swapoff {{[-a|--all]}}
```

- Disable a swap partition by its label:

```
sudo swapoff -L {{label}}
```

swapon

Enable devices and files for swapping.

Note: **`path/to/file`** can either point to a regular file or a swap partition.

More information: https://manned.org/swapon.8.

- Show swap information:

  `swapon`

- Enable a given swap area:

  `sudo swapon {{path/to/file}}`

- Enable all swap areas specified in `/etc/fstab` except those with the `noauto` option:

  `sudo swapon {{[-a|--all]}}`

- Enable a swap partition by its label:

  `sudo swapon -L {{label}}`

switch_root

Use a different filesystem as the root of the mount tree.

Note: **`switch_root`** will fail to function if the new root is not the root of a mount. Use bind-mounting as a workaround.

See also: **`chroot`**, **`mount`**.

More information: https://manned.org/switch_root.8.

- Move `/proc`, `/dev`, `/sys` and `/run` to the specified filesystem, use this filesystem as the new root and start the specified init process:

```
switch_root {{new_root}} {{/sbin/init}}
```

- Display help:

```
switch_root {{[-h|--help]}}
```

swupd

Package management utility for Clear Linux.

More information: https://www.clearlinux.org/clear-linux-documentation/guides/clear/swupd.html.

- Update to the latest version:

```
sudo swupd update
```

- Show current version, and check whether a newer one exists:

```
swupd check-update
```

- List installed bundles:

```
swupd bundle-list
```

- Locate the bundle where a wanted package exists:

```
swupd search -b {{package}}
```

- Install a new bundle:

```
sudo swupd bundle-add {{bundle}}
```

- Remove a bundle:

```
sudo swupd bundle-remove {{bundle}}
```

- Correct broken or missing files:

```
sudo swupd verify
```

swww

Efficient animated wallpaper daemon for Wayland.

See also: **`swww-daemon`**.

More information: https://github.com/LGFae/swww.

- Set wallpaper:

```
swww img {{path/to/image}}
```

- Set wallpaper to specified outputs:

```
swww img {{[-o|--outputs]}} {{output1,output2,...}} {{path/to/image}}
```

- Restore last wallpaper:

```
swww restore
```

- Kill daemon:

```
swww kill
```

- Display output information:

```
swww query
```

sxiv

Simple X Image Viewer.

More information: https://manned.org/sxiv.

- Open an image:

```
sxiv {{path/to/image}}
```

- Open an image in fullscreen mode:

```
sxiv -f {{path/to/file}}
```

- Open a newline-separated list of images, reading filenames from `stdin`:

```
echo {{path/to/file}} | sxiv -i
```

- Open one or more images as a slideshow:

```
sxiv -S {{seconds}} {{path/to/image1 path/to/image2 ...}}
```

- Open one or more images in thumbnail mode:

```
sxiv -t {{path/to/image1 path/to/image2 ...}}
```

synopkg

Package management utility for Synology DiskStation Manager.

More information: https://www.synology.com/dsm.

- List the names of installed packages:

```
synopkg list --name
```

- List packages which depend on a specific package:

```
synopkg list --depend-on {{package}}
```

- Start/Stop a package:

```
sudo synopkg {{start|stop}} {{package}}
```

- Print the status of a package:

```
synopkg status {{package}}
```

- Uninstall a package:

```
sudo synopkg uninstall {{package}}
```

- Check if updates are available for a package:

```
synopkg checkupdate {{package}}
```

- Upgrade all packages to the latest version:

```
sudo synopkg upgradeall
```

- Install a package from a synopkg file:

```
sudo synopkg install {{path/to/package.spk}}
```

synoupgrade

Upgrade Synology DiskStation Manager (DSM) - the Synology NAS operating system.

More information: https://www.synology.com/dsm.

- Check if upgrades are available:

```
sudo synoupgrade --check
```

- Check for patches without upgrading the DSM version:

```
sudo synoupgrade --check-smallupdate
```

- Download the latest upgrade available (use `--download-smallupdate` for patches):

```
sudo synoupgrade --download
```

- Start the upgrade process:

```
sudo synoupgrade --start
```

- Upgrade to the latest version automatically:

```
sudo synoupgrade --auto
```

- Apply patches without upgrading the DSM version automatically:

```
sudo synoupgrade --auto-smallupdate
```

- Upgrade the DSM using a patch file (should be an absolute path):

```
sudo synoupgrade --patch /{{path/to/file.pat}}
```

- Display help:

```
synoupgrade
```

sysctl

List and change kernel runtime variables.

More information: https://manned.org/sysctl.8.

- Show all available variables and their values:

```
sysctl {{[-a|--all]}}
```

- Set a changeable kernel state variable:

```
sysctl {{[-w|--write]}} {{section.tunable}}={{value}}
```

- Get currently open file handlers:

```
sysctl fs.file-nr
```

- Get limit for simultaneous open files:

```
sysctl fs.file-max
```

- Apply changes from `/etc/sysctl.conf`:

```
sysctl {{[-p|--load]}}
```

sysdig

System troubleshooting, analysis and exploration.

Capture, filter and store systemcalls.

More information: https://github.com/draios/sysdig/wiki.

- Capture all the events from the live system and print them to screen:

```
sysdig
```

- Capture all the events from the live system and save them to disk:

```
sysdig {{[-w|--write]}} {{path/to/file}}.scap
```

- Read events from a file and print them to screen:

```
sysdig {{[-r|--read]}} {{path/to/file}}.scap
```

- Filter and Print all the open system calls invoked by cat:

```
sysdig proc.name=cat and evt.type=open
```

- Register any found plugin and use dummy as input source passing to it open params:

```
sysdig -I dummy:'{{parameter}}'
```

- List the available chisels:

```
sysdig {{[-cl|--list-chisels]}}
```

- Use the spy_ip chisel to look at the data exchanged with ip address:

```
sysdig {{[-c|--chisel]}} spy_ip {{ip_address}}
```

systemctl add-requires

Add **`Requires`** dependencies to a target for one or more units.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#add-wants%20TARGET%0A%20%20%20%20%20%20%20%20%20%20UNIT%E2%80%A6.

- Add a `Requires` dependency from a target to a unit:

```
systemctl add-requires {{target}} {{unit}}
```

- Add multiple `Requires` dependencies at once:

```
systemctl add-requires {{target}} {{unit1 unit2 ...}}
```

- Add a user-level `Requires` dependency:

```
systemctl add-requires {{target}} {{unit}} --user
```

systemctl add-wants

Add **`Wants`** dependencies to a target for one or more units.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#add-wants%20TARGET%0A%20%20%20%20%20%20%20%20%20%20UNIT%E2%80%A6.

- Add a `Wants` dependency from a target to a unit:

```
systemctl add-wants {{target}} {{unit}}
```

- Add multiple `Wants` dependencies at once:

```
systemctl add-wants {{target}} {{unit1 unit2 ...}}
```

- Add a user-level `Wants` dependency:

```
systemctl add-wants {{target}} {{unit}} --user
```

systemctl bind

Ephemerally bind-mount a file or directory from the host into a unit's mount namespace.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#%0A%20%20%20%20%20%20%20%20%20%20%20%20bind%0A%20%20%20%2

- Bind-mount a host path into the same location inside the unit:

```
systemctl bind {{unit}} /{{path/to/host_directory}}
```

- Bind-mount a host path into a different location inside the unit:

```
systemctl bind {{unit}} /{{path/to/host_directory}} /{{path/to/unit_directory}}
```

- Bind-mount a path as read-only inside the unit:

```
systemctl bind {{unit}} /{{path/to/host_directory}} --read-only
```

- Create the destination path inside the unit before binding:

```
systemctl bind {{unit}} /{{path/to/host_directory}} /{{path/to/unit_directory}} --mkdir
```

systemctl cancel

Cancel one or more pending jobs in the system manager or user manager.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#cancel%20JOB%E2%80%A6.

- Cancel a job by its numeric ID:

```
systemctl cancel {{job_id}}
```

- Cancel multiple jobs:

```
systemctl cancel {{job_id1 job_id2 ...}}
```

- Cancel all pending jobs:

```
systemctl cancel
```

- Cancel a job in the user service manager:

```
systemctl cancel {{job_id}} --user
```

systemctl cat

Show the full contents of unit files as systemd sees them.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#cat%20PATTERN%E2%80%A6.

- Show the contents and absolute path of a unit file:

```
systemctl cat {{unit}}
```

- Show the contents of multiple unit files:

```
systemctl cat {{unit1 unit2 ...}}
```

- Show the contents of a unit file for a template:

```
systemctl cat {{template@}}
```

- Show the contents of a user unit file:

```
systemctl cat {{unit}} --user
```

systemctl daemon-reexec

Reexecute systemd while preserving current unit states.

See also: **`systemctl daemon-reload`** for applying changes in unit files.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#daemon-reexec.

- Reexecute systemd:

```
systemctl daemon-reexec
```

systemctl daemon-reload

Reload systemd manager configuration.

Use this after creating, modifying, or deleting unit files.

See also: **`systemctl reload`** for reloading service configuration.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#daemon-reload.

- Reload systemd to apply changes in unit files:

```
systemctl daemon-reload
```

systemctl default

Enter system default mode.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#default.

- Enter default mode as a blocking operation:

```
systemctl default
```

- Enter default mode as asynchronous operation:

```
systemctl default --no-block
```

systemctl disable

Disable systemd services.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#disable%20UNIT%E2%80%A6.

- Stop a service from running on boot:

```
systemctl disable {{unit}}
```

- Stop a service from running on boot and stop its current execution:

```
systemctl disable {{unit}} --now
```

- Stop a user service from running on login:

```
systemctl disable {{unit}} --user
```

systemctl edit

Edit systemd unit files.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#edit%20UNIT%E2%80%A6.

- Overlay a unit file non-destructively:

```
sudo systemctl edit {{unit_file}}
```

- Edit an unit file:

```
sudo systemctl edit {{unit_file}} {{[-l|--full]}}
```

- Create a new unit file:

```
sudo systemctl edit {{unit_file}} {{[-lf|--full --force]}}
```

- Overlay a user unit file:

```
systemctl edit {{unit_file}} --user
```

systemctl emergency

Enter emergency mode **`emergency.target`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#emergency.

- Enter emergency mode:

```
systemctl emergency
```

- Enter emergency mode without blocking (return immediately):

```
systemctl emergency --no-block
```

systemctl enable

Enable systemd services.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#enable%20UNIT%E2%80%A6.

- Enable a service to run on boot:

```
systemctl enable {{unit}}
```

- Enable a service to run on boot and start it now:

```
systemctl enable {{unit}} --now
```

- Enable a user unit to run on login:

```
systemctl enable {{unit}} --user
```

systemctl exit

Ask the service manager to quit.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#exit%20EXIT_CODE.

- Exit the user service manager:

```
systemctl exit --user
```

- Exit the user service manager with a specific exit code:

```
systemctl exit {{code}} --user
```

- Ask the container's service manager to exit (equivalent of `systemctl poweroff` if not in a container):

```
systemctl exit
```

systemctl freeze

Freeze one or more units.

Frozen units can be resumed with **`systemctl thaw`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#freeze%20PATTERN%E2%80%A6.

- Freeze a specific unit:

```
systemctl freeze {{unit}}
```

- Freeze multiple units:

```
systemctl freeze {{unit1 unit2 ...}}
```

- Freeze all running units:

```
systemctl freeze '*'
```

systemctl get-default

Display the default target (runlevel) that the system will boot into.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#get-default.

- Check the default target on your system:

```
systemctl get-default
```

systemctl-halt

Shut down and halt the system (stop the OS kernel but keep hardware powered on).

See also: **`halt`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#halt.

- Halt the system:

```
systemctl halt
```

- Halt the system immediately without asking services to stop gracefully:

```
systemctl halt --force
```

- Halt the system immediately without sending notifications to logged-in users:

```
systemctl halt --force --no-wall
```

- Halt the system immediately without terminating any processes or unmounting filesystems (dangerous, may cause data loss):

```
systemctl halt --force --force
```

- Schedule a halt at a specific time (e.g., 23:00):

```
systemctl halt --when 23:00
```

- Schedule a halt after a certain duration (e.g., 2 hours):

```
systemctl halt --when +2h
```

- Cancel a scheduled halt:

```
systemctl halt --when cancel
```

systemctl help

Show the manual pages for one or more units, or for the unit a process belongs to (by PID).

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#help%20PATTERN%E2%80%A6%7CPID%E2%80%A6.

- Show the manual page for a specific unit:

```
systemctl help {{unit}}
```

- Show the manual pages for multiple units:

```
systemctl help {{unit1 unit2 ...}}
```

- Show the manual page for a user unit:

```
systemctl help {{unit}} --user
```

- Show the manual page without a pager (all at once):

```
systemctl help {{unit}} --no-pager
```

- Show the manual page for the unit of a process by PID:

```
systemctl help {{pid}}
```

systemctl hibernate

Hibernate the system by saving the current state to disk and powering off.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#hibernate.

- Hibernate the system immediately:

```
systemctl hibernate
```

- Force hibernation even if inhibitors are present:

```
systemctl hibernate --force
```

- Hibernate the system without sending a message to logged-in users:

```
systemctl hibernate --no-wall
```

systemctl hybrid-sleep

Put the system into hybrid sleep, which combines suspend-to-RAM and hibernate.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#hybrid-sleep.

- Put the system into hybrid sleep immediately:

```
systemctl hybrid-sleep
```

- Force hybrid sleep even if inhibitors are present:

```
systemctl hybrid-sleep --force
```

- Put the system into hybrid sleep without sending a wall message to logged-in users:

```
systemctl hybrid-sleep --no-wall
```

systemctl is-active

Check if one or more systemd units are active.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#is-active%20PATTERN%E2%80%A6.

- Check whether a unit is active:

```
systemctl is-active {{unit}}
```

- Check whether multiple units are active:

```
systemctl is-active {{unit1 unit2 ...}}
```

- Check whether a unit is active without printing the state to `stdout`:

```
systemctl is-active {{unit}} {{[-q|--quiet]}}
```

- Check whether a user unit is active:

```
systemctl is-active {{unit}} --user
```

systemctl is-enabled

Check whether unit files are enabled.

See also: **`systemctl enable`**, **`systemctl disable`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#is-enabled%20UNIT%E2%80%A6.

- Show the enablement state:

```
systemctl is-enabled {{unit1 unit2 ...}}
```

- Suppress output and return only the exit code:

```
systemctl is-enabled {{unit}} --quiet
```

- Show installation targets and symlink paths:

```
systemctl is-enabled {{unit}} --full
```

systemctl is-failed

Check if one or more systemd units have failed.

See also: **`systemctl is-active`**, **`systemctl status`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#is-failed%20PATTERN%E2%80%A6.

- Check if there are any failed units:

```
systemctl is-failed
```

- Check if a unit or multiple units have failed:

```
systemctl is-failed {{unit1 unit2 ...}}
```

- Suppress output and return only the exit code:

```
systemctl is-failed {{unit}} {{[-q|--quiet]}}
```

- Check if a user unit has failed:

```
systemctl is-failed {{unit}} --user
```

systemctl is-system-running

Check the current state of the system.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#is-system-running.

- Check whether the system is operational and print the current state:

```
systemctl is-system-running
```

- Check and print the current state quietly (no output, only exit status):

```
systemctl is-system-running --quiet
```

- Wait until the boot process is completed before printing the current state:

```
systemctl is-system-running --wait
```

systemctl kill

Send a signal to one or more processes of a unit.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#kill%20PATTERN%E2%80%A6.

- Send the `SIGTERM` signal to a unit to terminate it:

```
systemctl kill {{unit}}
```

- Send a specific signal to a unit:

```
systemctl kill {{[-s|--signal]}} {{signal_number|signal_name}} {{unit}}
```

- Send a `SIGHUP` signal to only the main process of a unit:

```
systemctl kill {{[-s|--signal]}} SIGHUP --kill-whom main {{unit}}
```

- List all available signals:

```
systemctl kill {{[-s|--signal]}} help
```

systemctl list-automounts

List automount units currently in memory, showing mount paths and unit names.

See also: **`systemctl list-units`**, **`systemctl list-unit-files`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#list-automounts%20PATTERN%E2%80%A6.

- List automount units currently in memory:

```
systemctl list-automounts
```

- List all automount units, including inactive ones:

```
systemctl list-automounts {{[-a|--all]}}
```

- Filter automount units by state:

```
systemctl list-automounts --state {{active|inactive|
failed|...}}
```

- Filter automount units by name pattern:

```
systemctl list-automounts {{pattern1 pattern2 ...}}
```

systemctl list-dependencies

Show a unit's dependency tree in systemd.

See also: **`systemctl list-units`** to list loaded units.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#%0A%20%20%20%20%20%20%20%20%20%20%20%20list-dependencies%0A%20%20%20%20%20%20%20%20%20%20%20%20UNIT...%0A%20%20%20%20%20%20%20%20%20%20.

- Show the dependency tree of `default.target`:

```
systemctl list-dependencies
```

- Show the dependency tree of a specific unit:

```
systemctl list-dependencies {{unit}}
```

- Include all dependency types (not only `Requires=` and `Wants=`):

```
systemctl list-dependencies {{unit}} {{[-a|--all]}}
```

- Limit the tree to a specific unit type:

```
systemctl list-dependencies {{unit}} {{[-t|--type]}} {{service|socket|target|mount|...}}
```

- Reverse the direction to show units that depend on the specified unit:

```
systemctl list-dependencies {{unit}} --reverse
```

- Print output without headers or footers (for scripts):

```
systemctl list-dependencies {{unit}} --no-legend
```

systemctl list-jobs

List of active systemd jobs that are currently queued or running on the system.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#list-jobs%20PATTERN%E2%80%A6.

- List all active jobs:

```
systemctl list-jobs
```

- Filter jobs for a specific unit:

```
systemctl list-jobs {{unit}}
```

systemctl list-sockets

List active socket units currently in memory, ordered by listening address.

See also: **`systemctl list-units`**, **`systemctl list-unit-files`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#list-sockets%20PATTERN%E2%80%A6.

- List active socket units currently in memory:

```
systemctl list-sockets
```

- List active socket units with their socket types:

```
systemctl list-sockets --show-types
```

- List all socket units, including inactive and failed ones:

```
systemctl list-sockets {{[-a|--all]}}
```

- List socket units filtered by state:

```
systemctl list-sockets --state {{active|inactive|failed|...}}
```

- List socket units matching a name pattern:

```
systemctl list-sockets {{pattern1 pattern2 ...}}
```

systemctl list-timers

List all active systemd timers.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#list-timers%20PATTERN%E2%80%A6.

- List all active timers:

```
systemctl list-timers
```

- List all timers, including inactive ones:

```
systemctl list-timers {{[-a|--all]}}
```

- List timers matching a pattern:

```
systemctl list-timers {{pattern}}
```

- List timers matching a specific state:

```
systemctl list-timers --state {{active|inactive|failed|...}}
```

systemctl list-unit-files

List installed unit files and their enablement states.

See also: **`systemctl list-units`** for listing units currently loaded in memory.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#list-unit-files%20PATTERN%E2%80%A6.

- List installed unit files and their states:

```
systemctl list-unit-files
```

- Filter by state:

```
systemctl list-unit-files --state {{enabled|disabled|static|...}}
```

- Filter by unit type:

```
systemctl list-unit-files {{[-t|--type]}} {{service|socket|timer|...}}
```

- Filter by a name pattern:

```
systemctl list-unit-files '{{sshd*}}'
```

- Print output directly to `stdout`:

```
systemctl list-unit-files --no-pager
```

- Print output without headers or footers:

```
systemctl list-unit-files --no-legend
```

systemctl list-units

List units that systemd currently has in memory.

See also: **`systemctl list-unit-files`** for listing installed unit files.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#list-units%20PATTERN%E2%80%A6.

- List units which are active, have pending jobs, or have failed:

```
systemctl list-units
```

- List all units, including inactive ones:

```
systemctl list-units {{[-a|--all]}}
```

- Filter by unit type:

```
systemctl list-units {{[-t|--type]}} {{service|socket|timer|...}}
```

- Filter by state:

```
systemctl list-units --state {{running|listening|dead|...}}
```

- Filter by a name pattern:

```
systemctl list-units 'systemd*'
```

- Print output directly to `stdout`:

```
systemctl list-units --no-pager
```

- Print output without headers or footers (for scripts):

```
systemctl list-units --no-legend
```

systemctl mask

Link units to **`/dev/null`** so that they can be even started.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#mask%20UNIT%E2%80%A6.

- Mask a service:

```
systemctl mask {{service_name}}
```

- Ensure that the service is shut down while masking:

```
systemctl mask {{service_name}} --now
```

- Mask a user service:

```
systemctl mask {{service_name}} --user
```

systemctl-poweroff

Power off the system.

See also: **`poweroff`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#poweroff.

- Power off the system:

```
systemctl poweroff
```

- Power off the system immediately without asking services to stop gracefully:

```
systemctl poweroff --force
```

- Power off the system immediately without sending notifications to logged-in users:

```
systemctl poweroff --force --no-wall
```

systemctl preset-all

Reset the enablement state of all installed units to the defaults specified in preset policy files.

See also: **`systemctl preset`**, **`systemctl list-unit-files`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#preset-all.

- Reset the enablement state of all installed units:

```
sudo systemctl preset-all
```

- Enable only if marked as enabled in the preset policy:

```
sudo systemctl preset-all --preset-mode enable-only
```

- Disable only if marked as disabled in the preset policy:

```
sudo systemctl preset-all --preset-mode disable-only
```

- Suppress output and return only the exit code:

```
sudo systemctl preset-all {{[-q|--quiet]}}
```

systemctl preset

Reset the enablement state of unit files to the defaults specified in preset policy files.

See also: **`systemctl preset-all`**, **`systemctl list-unit-files`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#preset%20UNIT%E2%80%A6.

- Reset the enablement state to preset defaults:

```
systemctl preset {{unit1 unit2 ...}}
```

- Enable only if marked as enabled in the preset policy:

```
systemctl preset {{unit}} --preset-mode enable-only
```

- Disable only if marked as disabled in the preset policy:

```
systemctl preset {{unit}} --preset-mode disable-only
```

- Suppress output and return only the exit code:

```
systemctl preset {{unit}} {{[-q|--quiet]}}
```

systemctl reboot

Reboot the system.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#reboot.

- Reboot the system:

```
systemctl reboot
```

- Reboot into the BIOS/UEFI menu:

```
systemctl reboot --firmware-setup
```

systemctl reload

Reload a service's configuration without restarting it.

This reloads the service itself (like Apache or Nginx configs), not the systemd unit file.

To reload unit files, use **`systemctl daemon-reload`**.

- Reload a service:

  ```
  systemctl reload {{nginx}}
  ```

- Reload multiple services:

  ```
  systemctl reload {{networking apache2 ...}}
  ```

- Reload a service for the current user:

  ```
  systemctl reload {{pipewire}} --user
  ```

systemctl rescue

Enter rescue mode.

See also: **`systemctl emergency`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#rescue.

- Enter rescue mode:

```
systemctl rescue
```

- Enter rescue mode asynchronously:

```
systemctl rescue --no-block
```

systemctl reset-failed

Reset the "failed" state of one or more units.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#reset-failed%20%5BPATTERN%E2%80%A6%5D.

- Reset the failed state of all units:

```
systemctl reset-failed
```

- Reset the failed state of a specific unit:

```
systemctl reset-failed {{unit}}
```

- Reset multiple units at once:

```
systemctl reset-failed {{unit_1 unit_2 ...}}
```

systemctl restart

Stop and then start one or more systemd units.

Can be used in place of **`systemctl start`** on a stopped unit, but **`start`** is safer so that a running unit isn't accidentally restarted.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#restart%20PATTERN%E2%80%A6.

- Restart a unit:

```
systemctl restart {{unit}}
```

- Restart more than one unit:

```
systemctl restart {{unit1 unit2 ...}}
```

- Restart a user unit:

```
systemctl restart {{unit}} --user
```

systemctl revert

Revert unit files to their vendor versions.

Undoes the effects of **`edit`**, **`enable`**, **`disable`**, **`set-property`**, and **`mask`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#revert%20UNIT%E2%80%A6.

- Revert unit files to their default settings:

```
systemctl revert {{unit1 unit2 ...}}
```

- Revert a user unit file:

```
systemctl revert {{unit}} --user
```

systemctl set-default

Symlink the **`default.target`** alias to the given target unit.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#set-default%20TARGET.

- Set `systemd`'s default boot mode:

```
systemctl set-default {{target_name.target}}
```

- Set `systemd` to boot to GUI mode by default:

```
systemctl set-default graphical.target
```

- Set `systemd` to boot to CLI mode by default:

```
systemctl set-default multi-user.target
```

systemctl set-environment

Set one or more service manager environment variables.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#set-environment%20VARIABLE=VALUE%E2%80%A6.

- Set a single environment variable:

```
systemctl set-environment {{var value}}
```

- Set multiple environment variables at once:

```
systemctl set-environment {{var1 value1 var2 value2 ...}}
```

- Set an environment variable for the user service manager:

```
systemctl set-environment {{var value}} --user
```

systemctl set-property

Set the specified unit properties at runtime.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#set-property%20UNIT%20PROPERTY=VALUE%E2%80%A6.

- Set a property for a running service:

```
systemctl set-property {{unit}} {{property}}={{value}}
```

- Set multiple properties at once:

```
systemctl set-property {{unit}} {{property_1=value_1 property_2=value_2 ...}}
```

- Set a property only for the current runtime session (not persistent):

```
systemctl set-property {{unit}} {{property}}={{value}} --runtime
```

- Reset a property to its default value:

```
systemctl set-property {{unit}} {{property}}=
```

- Reset multiple properties to its default values:

```
systemctl set-property {{unit}} {{property_1= property_2= ...}}
```

systemctl show-environment

Display systemd environment blocks.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#show-environment.

- Display the environment block for the system:

```
systemctl show-environment
```

- Display the environment block for the current user:

```
systemctl show-environment --user
```

systemctl show

Show properties of units or systemd itself.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#show%20PATTERN%E2%80%A6%7CJOB%E2%80%A6.

- Show properties of the system service manager:

```
systemctl show
```

- Show properties of the user service manager:

```
systemctl show --user
```

- Show properties of a specific unit:

```
systemctl show {{unit}}
```

- Show properties of a specific user unit:

```
systemctl show {{unit}} --user
```

- Include empty properties in the list:

```
systemctl show {{[-a|--all]}}
```

- Only show the specified properties:

```
systemctl show {{unit}} {{[-p|--property]}}
{{Wants,Conflicts,...}}
```

systemctl soft-reboot

Shut down and reboot userspace, leaving the kernel running.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#soft-reboot.

- Perform a soft reboot immediately:

```
systemctl soft-reboot
```

- Force a soft reboot:

```
systemctl soft-reboot --force
```

- Schedule a soft reboot for a specific time:

```
systemctl soft-reboot --when "{{timestamp}}"
```

- Cancel a scheduled soft reboot:

```
systemctl soft-reboot --when cancel
```

systemctl start

Start systemd units.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#start%20PATTERN%E2%80%A6.

- Start a unit:

```
systemctl start {{unit}}
```

- Start a user unit:

```
systemctl start {{unit}} --user
```

systemctl status

Display the status of systemd units.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#status%20PATTERN%E2%80%A6%7CPID%E2%80%A6%5D.

- Show the status of a systemd unit:

```
systemctl status {{unit}}.{{service|timer|socket|target|...}}
```

- Show the status of failed units:

```
systemctl status --failed
```

- List all running services:

```
systemctl status
```

- List all units in the system:

```
systemctl status --all
```

- List all units of a specific type:

```
systemctl status --type {{service|timer|socket|target|...}}
```

- List all units with a specific state:

```
systemctl status --state {{active|inactive|failed}}
```

- Show the status of a user unit:

```
systemctl status {{unit}} --user
```

systemctl stop

Stop systemd units.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#stop%20PATTERN%E2%80%A6.

- Stop a unit:

```
systemctl stop {{unit}}
```

- Stop a service and suppress warnings:

```
systemctl stop {{unit}} --no-warn
```

- Stop a user unit:

```
systemctl stop {{unit}} --user
```

systemctl suspend-then-hibernate

Suspend the system, then automatically hibernate after a period of inactivity.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#suspend-then-hibernate.

- Suspend the system and hibernate after the configured delay:

```
systemctl suspend-then-hibernate
```

- Force suspend-then-hibernate (bypass inhibitor locks):

```
systemctl suspend-then-hibernate --force
```

systemctl suspend

Suspend the system.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#suspend.

- Suspend the system immediately:

```
systemctl suspend
```

- Schedule a suspend after a 5 minute delay:

```
sleep 300 && systemctl suspend
```

- Suspend the system and then hibernate after a delay:

```
systemctl hybrid-sleep
```

systemctl switch-root

Switch to a new root filesystem and execute a new system manager.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#switch-root%20ROOT%20INIT.

- Switch to a new root filesystem and execute its default init system:

```
systemctl switch-root {{path/to/new_root}}
```

- Switch to a new root filesystem and run a specific init binary:

```
systemctl switch-root {{path/to/new_root}} {{/sbin/init}}
```

- Switch to a new root filesystem with verbose output:

```
systemctl switch-root {{path/to/new_root}} --verbose
```

systemctl thaw

Thaw (resume) one or more frozen units.

Units can be frozen with **`systemctl freeze`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#thaw%20PATTERN%E2%80%A6.

- Thaw a specific unit:

```
systemctl thaw {{unit}}
```

- Thaw multiple units:

```
systemctl thaw {{unit1 unit2 ...}}
```

- Thaw all currently frozen units:

```
systemctl thaw '*'
```

systemctl try-reload-or-restart

Reload one or more units if they support it; otherwise restart them.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#try-reload-or-restart%20PATTERN%E2%80%A6.

- Reload or restart a specific unit:

```
systemctl try-reload-or-restart {{unit}}
```

- Reload or restart multiple units:

```
systemctl try-reload-or-restart {{unit1 unit2 ...}}
```

- Reload or restart all units matching a pattern:

```
systemctl try-reload-or-restart '{{pattern}}'
```

systemctl try-restart

Restart one or more units only if they are currently running.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#try-restart%20PATTERN%E2%80%A6.

- Restart a specific unit if it is running:

```
systemctl try-restart {{unit}}
```

- Restart multiple units if they are running:

```
systemctl try-restart {{unit1 unit2 ...}}
```

- Restart all units matching a pattern if they are running:

```
systemctl try-restart '{{pattern}}'
```

systemctl unmask

Unmask units to make them startable again.

This undoes the effect of **`systemctl mask`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#unmask%20UNIT%E2%80%A6.

- Unmask a service:

```
systemctl unmask {{service_name}}
```

- Unmask and start a service immediately:

```
systemctl unmask {{service_name}} --now
```

- Unmask a user service:

```
systemctl unmask {{service_name}} --user
```

systemctl unset-environment

Unset one or more service manager environment variables.

This undoes the effect of **`systemctl set-environment`**.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#unset-environment%20VARIABLE%E2%80%A6.

- Unset a single environment variable:

```
systemctl unset-environment {{var}}
```

- Unset multiple environment variables at once:

```
systemctl unset-environment {{var1 var2 ...}}
```

- Unset an environment variable in the user service manager:

```
systemctl unset-environment {{var}} --user
```

systemctl whoami

Show units that processes belong to.

If no PID is specified, shows the unit the **`systemctl`** command itself is invoked in.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html#whoami%20%5BPID%E2%80%A6%5D.

- Show the unit of the current shell (where `systemctl` is running):

```
systemctl whoami
```

- Show the unit of the current shell in user service manager (services managed for your login session):

```
systemctl whoami --user
```

- Show the unit a specific process belongs to:

```
systemctl whoami {{pid}}
```

- Show the units for multiple processes:

```
systemctl whoami {{pid1 pid2 ...}}
```

systemctl

Control the systemd system and service manager.

Some subcommands such as **`disable`**, **`status`**, **`reboot`** etc. have their own usage documentation.

More information: https://www.freedesktop.org/software/systemd/man/systemctl.html.

- Show all running services:

```
systemctl status
```

- List failed units:

```
systemctl --failed
```

- Start/Stop/Restart/Reload/Show the status a service:

```
systemctl {{start|stop|restart|reload|status}} {{unit}}
```

- Enable/Disable a unit to be started on bootup:

```
systemctl {{enable|disable}} {{unit}}
```

- Reload systemd, scan for new or changed units:

```
systemctl daemon-reload
```

- Check if a unit is active/enabled/failed:

```
systemctl {{is-active|is-enabled|is-failed}} {{unit}}
```

- List all service/socket/automount units filtering by running/failed state:

```
systemctl list-units {{[-t|--type]}} {{service|socket|
automount}} --state {{failed|running}}
```

- Show the contents & absolute path of a unit file or edit it:

```
systemctl {{cat|edit}} {{unit}}
```

systemd-ac-power

Report whether the computer is connected to an external power source.

More information: https://www.freedesktop.org/software/systemd/man/systemd-ac-power.html.

- Silently check and return a 0 status code when running on AC power, and a non-zero code otherwise:

```
systemd-ac-power
```

- Additionally print `yes` or `no` to `stdout`:

```
systemd-ac-power {{[-v|--verbose]}}
```

systemd-analyze

Analyze and debug system manager.

Show timing details about the boot process of units (services, mount points, devices, sockets).

More information: https://www.freedesktop.org/software/systemd/man/systemd-analyze.html.

- Print the last system startup time:

```
systemd-analyze
```

- List all running units, ordered by the time they took to initialize:

```
systemd-analyze blame
```

- Print a tree of the time-critical chain of units:

```
systemd-analyze critical-chain
```

- Create an SVG file showing when each system service started, highlighting the time that they spent on initialization:

```
systemd-analyze plot > {{path/to/file.svg}}
```

- Plot a dependency graph and convert it to an SVG file:

```
systemd-analyze dot | dot -T {{svg}} > {{path/to/file.svg}}
```

- Show security scores of running units:

```
systemd-analyze security
```

- View all udev rules together:

```
systemd-analyze cat-config udev/rules.d
```

systemd-ask-password

Query the user for a system password.

More information: https://www.freedesktop.org/software/systemd/man/systemd-ask-password.html.

- Query a system password with a specific prompt:

```
systemd-ask-password "{{prompt}}"
```

- Specify an identifier for the password query:

```
systemd-ask-password --id {{identifier}} "{{prompt}}"
```

- Use a kernel keyring key name as a cache for the password:

```
systemd-ask-password --keyname {{key_name}} "{{prompt}}"
```

- Set a custom timeout for the password query:

```
systemd-ask-password --timeout {{seconds}} "{{prompt}}"
```

- Force the use of an agent system and never ask on current TTY:

```
systemd-ask-password --no-tty "{{prompt}}"
```

- Store a password in the kernel keyring without displaying it:

```
systemd-ask-password --no-output --keyname {{key_name}} "{{prompt}}"
```

- Pass the asked password to another program:

```
systemd-ask-password | {{command}}
```

- Display help:

```
systemd-ask-password {{[-h|--help]}}
```

systemd-cat

Connect a pipeline or program's output streams with the systemd journal.

More information: https://www.freedesktop.org/software/systemd/man/systemd-cat.html.

- Write the output of the specified command to the journal (both output streams are captured):

```
systemd-cat {{command}}
```

- Write the output of a pipeline to the journal (`stderr` stays connected to the terminal):

```
{{command}} | systemd-cat
```

- Use the specified identifier (default: `cat` when reading from a pipeline, executable name otherwise):

```
{{command}} | systemd-cat {{[-t|--identifier]}} {{id}}
```

- Use the specified default priority level for all logged messages:

```
systemd-cat {{[-p|--priority]}} {{emerg|alert|crit|err|
warning|notice|info|debug}} {{command}}
```

- Use the specified default priority level for the logged messages from the command's `stderr`:

```
systemd-cat --stderr-priority {{emerg|alert|crit|err|warning|
notice|info|debug}} {{command}}
```

systemd-cgls

Show the contents of the selected Linux control group hierarchy in a tree.

More information: https://www.freedesktop.org/software/systemd/man/systemd-cgls.html.

- Display the whole control group hierarchy on your system:

```
systemd-cgls
```

- Display a control group tree of a specific resource controller:

```
systemd-cgls {{cpu|memory|io}}
```

- Display the control group hierarchy of one or more systemd units:

```
systemd-cgls {{[-u|--unit]}} {{unit1 unit2 ...}}
```

systemd-cgtop

Show the top control groups of the local Linux control group hierarchy, ordered by their CPU, memory, or disk I/O load.

See also: **`top`**.

More information: https://www.freedesktop.org/software/systemd/man/systemd-cgtop.html.

- Start an interactive view:

```
systemd-cgtop
```

- Change the sort order:

```
systemd-cgtop --order {{cpu|memory|path|tasks|io}}
```

- Show the CPU usage by time instead of percentage:

```
systemd-cgtop --cpu=percentage
```

- Change the update interval in seconds (or one of these time units: `ms`, `us`, `min`):

```
systemd-cgtop {{[-d|--delay]}} {{interval}}
```

- Only count userspace processes (without kernel threads):

```
systemd-cgtop -P
```

systemd-confext

This command is an alias of **`systemd-sysext`**.

It follows the same principle as **`systemd-sysext`**, but instead of working on **`/usr`** and **`/opt`**, **`confext`** will extend only **`/etc`**.

- View documentation for the original command:

```
tldr systemd-sysext
```

systemd-creds

List, show, encrypt and decrypt service credentials.

More information: https://www.freedesktop.org/software/systemd/man/systemd-creds.html.

- Encrypt a file and set a specific name:

```
systemd-creds encrypt --name {{name}} {{path/to/input_file}} {{path/to/output}}
```

- Decrypt the file again:

```
systemd-creds decrypt {{path/to/input_file}} {{path/to/output_file}}
```

- Encrypt text from `stdin`:

```
echo -n {{text}} | systemd-creds encrypt --name {{name}} - {{path/to/output}}
```

- Encrypt the text and append it to the service file (the credentials will be available in `$CREDENTIALS_DIRECTORY`):

```
echo -n {{text}} | systemd-creds encrypt --name {{name}} --pretty - - >> {{service}}
```

- Create a credential that is only valid until the given timestamp:

```
systemd-creds encrypt --not-after "{{timestamp}}" {{path/to/input_file}} {{path/to/output_file}}
```

systemd-cryptenroll

Interactively enroll or remove methods used to unlock LUKS2-encrypted devices. Uses a password to unlock the device unless otherwise specified.

In order to allow a partition to be unlocked during system boot, update the **`/etc/crypttab`** file or the initramfs.

More information: https://www.freedesktop.org/software/systemd/man/systemd-cryptenroll.html.

- Enroll a new password (similar to `cryptsetup luksAddKey`):

```
systemd-cryptenroll --password {{path/to/luks2_block_device}}
```

- Enroll a new recovery key (i.e. a randomly generated passphrase that can be used as a fallback):

```
systemd-cryptenroll --recovery-key {{path/to/
luks2_block_device}}
```

- List available tokens, or enroll a new PKCS#11 token:

```
systemd-cryptenroll --pkcs11-token-uri {{list|auto|
pkcs11_token_uri}} {{path/to/luks2_block_device}}
```

- List available FIDO2 devices, or enroll a new FIDO2 device (`auto` can be used as the device name when there is only one token plugged in):

```
systemd-cryptenroll --fido2-device {{list|auto|path/to/
fido2_hidraw_device}} {{path/to/luks2_block_device}}
```

- Enroll a new FIDO2 device with user verification (biometrics):

```
systemd-cryptenroll --fido2-device {{auto|path/to/
fido2_hidraw_device}} --fido2-with-user-verification yes
{{path/to/luks2_block_device}}
```

- Unlock using a FIDO2 device, and enroll a new FIDO2 device:

```
systemd-cryptenroll --unlock-fido2-device {{path/to/
fido2_hidraw_unlock_device}} --fido2-device {{path/to/
fido2_hidraw_enroll_device}} {{path/to/luks2_block_device}}
```

- Enroll a TPM2 security chip (only secure-boot-policy PCR) and require an additional alphanumeric PIN:

```
systemd-cryptenroll --tpm2-device {{auto|path/to/
tpm2_block_device}} --tpm2-with-pin yes {{path/to/
luks2_block_device}}
```

- Remove all empty passwords/all passwords/all FIDO2 devices/all PKCS#11 tokens/all TPM2 security chips/all recovery keys/all methods:

  ```
  systemd-cryptenroll --wipe-slot {{empty|password|fido2|
  pkcs#11|tpm2|recovery|all}} {{path/to/luks2_block_device}}
  ```

systemd-cryptsetup

Create or remove decrypted mappings of encrypted volumes. Equivalent of **`cryptsetup open`** and **`cryptsetup close`**.

Arguments to this command are written exactly like a line in **`/etc/crypttab`**. It's used by systemd to unlock devices on boot.

See also: **`cryptsetup`**.

More information: https://www.freedesktop.org/software/systemd/man/systemd-cryptsetup.html.

- Open a LUKS volume and create a decrypted mapping at `/dev/mapper/mapping_name`:

```
systemd-cryptsetup attach {{mapping_name}} {{/dev/sdXY}}
```

- Open a LUKS volume with additional options and create a decrypted mapping at `/dev/mapper/mapping_name`:

```
systemd-cryptsetup attach {{mapping_name}} {{/dev/sdXY}} none {{crypttab_options}}
```

- Open a LUKS volume with a keyfile and create a decrypted mapping at `/dev/mapper/mapping_name`:

```
systemd-cryptsetup attach {{mapping_name}} {{/dev/sdXY}} {{path/to/keyfile}} {{crypttab_options}}
```

- Remove an existing mapping:

```
systemd-cryptsetup detach {{mapping_name}}
```

systemd-delta

Find overridden systemd-related configuration files.

More information: https://www.freedesktop.org/software/systemd/man/systemd-delta.html.

- Show all overridden configuration files:

```
systemd-delta
```

- Show only files of specific types (comma-separated list):

```
systemd-delta {{[-t|--type]}} {{masked|equivalent|redirected|overridden|extended|unchanged}}
```

- Show only files whose path starts with the specified prefix (Note: A prefix is a directory containing subdirectories with systemd configuration files):

```
systemd-delta {{/etc|/run|/usr/lib|...}}
```

- Further restrict the search path by adding a suffix (the prefix is optional):

```
systemd-delta {{prefix}}/{{tmpfiles.d|sysctl.d|systemd/system|...}}
```

systemd-detect-virt

Detect execution in a virtualized environment.

More information: https://www.freedesktop.org/software/systemd/man/systemd-detect-virt.html.

- List detectable virtualization technologies:

```
systemd-detect-virt --list
```

- Detect virtualization, print the result and return a zero status code when running in a VM or a container, and a non-zero code otherwise:

```
systemd-detect-virt
```

- Silently check without printing anything:

```
systemd-detect-virt {{[-q|--quiet]}}
```

- Only detect container virtualization:

```
systemd-detect-virt {{[-c|--container]}}
```

- Only detect hardware virtualization:

```
systemd-detect-virt {{[-v|--vm]}}
```

systemd-dissect

Introspect and interact with file system OS disk images, specifically Discoverable Disk Images (DDIs).

More information: https://www.freedesktop.org/software/systemd/man/systemd-dissect.html.

- Show general image information about the OS image:

```
systemd-dissect {{path/to/image.raw}}
```

- Mount an OS image:

```
systemd-dissect {{[-m|--mount]}} {{path/to/image.raw}} {{/mnt/image}}
```

- Unmount an OS image:

```
systemd-dissect {{[-u|--umount]}} {{/mnt/image}}
```

- List files in an image:

```
systemd-dissect {{[-l|--list]}} {{path/to/image.raw}}
```

- Attach an OS image to an automatically allocated loopback block device and print its path:

```
systemd-dissect --attach {{path/to/image.raw}}
```

- Detach an OS image from a loopback block device:

```
systemd-dissect --detach {{path/to/device}}
```

systemd-escape

Escape strings for usage in systemd unit names.

More information: https://www.freedesktop.org/software/systemd/man/systemd-escape.html.

- Escape the given text:

```
systemd-escape {{text}}
```

- Reverse the escaping process:

```
systemd-escape {{[-u|--unescape]}} {{text}}
```

- Treat the given text as a path:

```
systemd-escape {{[-p|--path]}} {{text}}
```

- Append the given suffix to the escaped text:

```
systemd-escape --suffix {{suffix}} {{text}}
```

- Use a template and inject the escaped text:

```
systemd-escape --template {{template}} {{text}}
```

systemd-firstboot

Initialize basic system settings on or before the first boot-up of a system.

More information: https://www.freedesktop.org/software/systemd/man/systemd-firstboot.html.

- Operate on the specified directory instead of the root directory of the host system:

```
sudo systemd-firstboot --root {{path/to/root_directory}}
```

- Set the system keyboard layout:

```
sudo systemd-firstboot --keymap {{keymap}}
```

- Set the system hostname:

```
sudo systemd-firstboot --hostname {{hostname}}
```

- Set the root user's password:

```
sudo systemd-firstboot --root-password {{password}}
```

- Prompt the user interactively for a specific basic setting:

```
sudo systemd-firstboot --prompt {{setting}}
```

- Force writing configuration even if the relevant files already exist:

```
sudo systemd-firstboot --force
```

- Remove all existing files that are configured by `systemd-firstboot`:

```
sudo systemd-firstboot --reset
```

- Remove the password of the system's root user:

```
sudo systemd-firstboot --delete-root-password
```

systemd-hwdb

Hardware database management tool.

More information: https://www.freedesktop.org/software/systemd/man/systemd-hwdb.html.

- Update the binary hardware database in `/etc/udev`:

```
systemd-hwdb update
```

- Query the hardware database and print the result for a specific modalias:

```
systemd-hwdb query {{modalias}}
```

- Update the binary hardware database, returning a non-zero exit value on any parsing error:

```
systemd-hwdb {{[-s|--strict]}} update
```

- Update the binary hardware database in `/usr/lib/udev`:

```
systemd-hwdb --usr update
```

- Update the binary hardware database in the specified root path:

```
systemd-hwdb {{[-r|--root]}} {{path/to/root}} update
```

systemd-id128

Generate and print sd-128 identifiers.

More information: https://www.freedesktop.org/software/systemd/man/systemd-id128.html.

- Generate a new random identifier:

```
systemd-id128 new
```

- Print the identifier of the current machine:

```
systemd-id128 machine-id
```

- Print the identifier of the current boot:

```
systemd-id128 boot-id
```

- Print the identifier of the current service invocation (this is available in systemd services):

```
systemd-id128 invocation-id
```

- Generate a new random identifier and print it as a UUID (five groups of digits separated by hyphens):

```
systemd-id128 new {{[-u|--uuid]}}
```

systemd-inhibit

Prohibit the system from entering certain power states.

Inhibitor locks may be used to block or delay system sleep and shutdown requests as well as automatic idle handling.

More information: https://www.freedesktop.org/software/systemd/man/systemd-inhibit.html.

- List all active inhibition locks and the reasons for their creation:

```
systemd-inhibit --list
```

- Block system shutdown for a specified number of seconds with the `sleep` command:

```
systemd-inhibit --what shutdown sleep {{5}}
```

- Keep the system from sleeping or idling until the download is complete:

```
systemd-inhibit --what sleep:idle wget {{https://example.com/file}}
```

- Ignore lid close switch until the script exits:

```
systemd-inhibit --what sleep:handle-lid-switch {{path/to/script}}
```

- Ignore power button press while command is running:

```
systemd-inhibit --what handle-power-key {{command}}
```

- Describe who and why created the inhibitor (default: the command and its arguments for `--who` and `Unknown reason` for `--why`):

```
systemd-inhibit --who {{$USER}} --why {{reason}} --what {{operation}} {{command}}
```

systemd-machine-id-setup

Initialize the machine ID stored in **`/etc/machine-id`** at install time with a provisioned or randomly generated ID.

Note: Always use **`sudo`** to execute these commands as they require elevated privileges.

More information: https://www.freedesktop.org/software/systemd/man/systemd-machine-id-setup.html.

- Print the generated or committed machine ID:

  ```
  systemd-machine-id-setup --print
  ```

- Specify an image policy:

  ```
  systemd-machine-id-setup --image-policy {{your_policy}}
  ```

- Display the output as JSON:

  ```
  sudo systemd-machine-id-setup --json pretty
  ```

- Operate on a disk image instead of a directory tree:

  ```
  systemd-machine-id-setup --image /{{path/to/image}}
  ```

systemd-mount

Establish and destroy transient mount or auto-mount points.

More information: https://www.freedesktop.org/software/systemd/man/systemd-mount.html.

- Mount a file system (image or block device) at `/run/media/system/LABEL` where LABEL is the filesystem label or the device name if there is no label:

```
systemd-mount {{path/to/file_or_device}}
```

- Mount a file system (image or block device) at a specific location:

```
systemd-mount {{path/to/file_or_device}} {{path/to/mount_point}}
```

- List all local, known block devices with file systems that may be mounted:

```
systemd-mount --list
```

- Create an automount point that mounts the actual file system at the time of first access:

```
systemd-mount --automount yes {{path/to/file_or_device}}
```

- Unmount one or more devices:

```
systemd-mount {{[-u|--umount]}} {{path/to/mount_point_or_device1 path/to/mount_point_or_device2 ...}}
```

- Mount a file system (image or block device) with a specific file system type:

```
systemd-mount {{[-t|--type]}} {{file_system_type}} {{path/to/file_or_device}} {{path/to/mount_point}}
```

- Mount a file system (image or block device) with additional mount options:

```
systemd-mount {{[-o|--options]}} {{mount_options}} {{path/to/file_or_device}} {{path/to/mount_point}}
```

systemd-notify

Notify the service manager about start-up completion and other daemon status changes.

This command is useless outside systemd service scripts.

More information: https://www.freedesktop.org/software/systemd/man/systemd-notify.html.

- Notify systemd that the service has completed its initialization and is fully started. It should be invoked when the service is ready to accept incoming requests:

```
systemd-notify --booted
```

- Signal to systemd that the service is ready to handle incoming connections or perform its tasks:

```
systemd-notify --ready
```

- Provide a custom status message to systemd (this information is shown by `systemctl status`):

```
systemd-notify --status "{{Add custom status message here...}}"
```

systemd-nspawn

Spawn a command or OS in a lightweight container.

More information: https://www.freedesktop.org/software/systemd/man/systemd-nspawn.html.

- Run a command in a container:

```
systemd-nspawn {{[-D|--directory]}} {{path/to/container_root}}
```

- Run a full Linux-based OS in a container:

```
systemd-nspawn {{[-b|--boot]}} {{[-D|--directory]}} {{path/to/container_root}}
```

- Run the specified command as PID 2 in the container (as opposed to PID 1) using a stub init process:

```
systemd-nspawn {{[-D|--directory]}} {{path/to/container_root}} {{[-a|--as-pid2]}}
```

- Specify the machine name and hostname:

```
systemd-nspawn {{[-M|--machine]}} {{container_name}} --hostname {{container_host}} {{[-D|--directory]}} {{path/to/container_root}}
```

systemd-path

List and query system and user paths.

More information: https://www.freedesktop.org/software/systemd/man/systemd-path.html.

- Display a list of known paths and their current values:

```
systemd-path
```

- Query the specified path and display its value:

```
systemd-path "{{path_name}}"
```

- Suffix printed paths with `suffix_string`:

```
systemd-path --suffix {{suffix_string}}
```

- Print a short version string and then exit:

```
systemd-path --version
```

systemd-repart

Automatically grow and add partitions.

Grows and adds partitions based on the configuration files described in repart.d.

Does not automatically resize file system on partition. See systemd-growfs to extend file system.

More information: https://www.freedesktop.org/software/systemd/man/systemd-repart.html.

- Grow the root partition (/) to all available disk space:

```
systemd-repart
```

- View changes without applying:

```
systemd-repart --dry-run yes
```

- Grow root partition size to 10 gigabytes:

```
systemd-repart --size 10G --root /
```

systemd-resolve

Resolve domain names, IPV4 and IPv6 addresses, DNS resource records, and services.

Note: This tool has been renamed to **`resolvectl`** in new versions of **`systemd`**.

More information: https://manned.org/systemd-resolve.

- View documentation for `resolvectl`:

```
tldr resolvectl
```

systemd-run

Run programs in transient scope units, service units, or path-, socket-, or timer-triggered service units.

More information: https://www.freedesktop.org/software/systemd/man/systemd-run.html.

- Start a transient service:

```
sudo systemd-run {{command}} {{argument1 argument2 ...}}
```

- Start a transient service under the service manager of the current user (no privileges):

```
systemd-run --user {{command}} {{argument1 argument2 ...}}
```

- Start a transient service with a custom unit name and description:

```
sudo systemd-run {{[-u|--unit]}} {{name}} --description
{{string}} {{command}} {{argument1 argument2 ...}}
```

- Start a transient service that does not get cleaned up after it terminates with a custom environment variable:

```
sudo systemd-run {{[-r|--remain-after-exit]}} --set-
env={{name}}={{value}} {{command}} {{argument1
argument2 ...}}
```

- Start a transient timer that periodically runs its transient service (see `man systemd.time` for calendar event format):

```
sudo systemd-run --on-calendar={{calendar_event}} {{command}}
{{argument1 argument2 ...}}
```

- Share the terminal with the program (allowing interactive input/output) and make sure the execution details remain after the program exits:

```
systemd-run {{[-r|--remain-after-exit]}} --pty {{command}}
```

- Set properties (e.g. CPUQuota, MemoryMax) of the process and wait until it exits:

```
systemd-run {{[-p|--property]}} MemoryMax={{memory_in_bytes}}
{{[-p|--property]}} CPUQuota={{percentage_of_CPU_time}}% --
wait {{command}}
```

- Use the program in a shell pipeline:

```
{{command1}} | systemd-run {{[-P|--pipe]}} {{command2}} |
{{command3}}
```

systemd-socket-activate

Socket activation for systemd services.

More information: https://www.freedesktop.org/software/systemd/man/systemd-socket-activate.html.

- Activate a service when a specific socket is connected:

```
systemd-socket-activate {{path/to/socket.service}}
```

- Activate multiple sockets for a service:

```
systemd-socket-activate {{path/to/socket1.service}} {{path/to/socket2.service}}
```

- Pass environment variables to the service being activated:

```
{{SYSTEMD_SOCKET_ACTIVATION=1}} systemd-socket-activate {{path/to/socket.service}}
```

- Activate a service along with a notification socket:

```
systemd-socket-activate {{path/to/socket.socket}} {{path/to/service.service}}
```

- Activate a service with a specified port:

```
systemd-socket-activate {{path/to/socket.service}} {{[-l|--listen]}} {{8080}}
```

systemd-stdio-bridge

Implement a proxy between **`stdin`**/**`stdout`** and a D-Bus.

Note: It expects to receive an open connection via **`stdin`**/**`stdout`** when started, and will create a new connection to the specified bus.

More information: https://www.freedesktop.org/software/systemd/man/systemd-stdio-bridge.html.

- Forward `stdin`/`stdout` to the local system bus:

```
systemd-stdio-bridge
```

- Forward `stdin`/`stdout` to a specific user's D-Bus:

```
systemd-stdio-bridge --{{user}}
```

- Forward `stdin`/`stdout` to the local system bus within a specific container:

```
systemd-stdio-bridge {{[-M|--machine]}} {{mycontainer}}
```

- Forward `stdin`/`stdout` to a custom D-Bus address:

```
systemd-stdio-bridge {{[-p|--bus-path]}} unix:path=/{{path/to/dbus_socket}}
```

systemd-sysext

Activate or deactivate system extension images.

More information: https://www.freedesktop.org/software/systemd/man/systemd-sysext.html.

- List installed extension images:

```
systemd-sysext list
```

- Merge system extension images into `/usr/` and `/opt/`:

```
systemd-sysext merge
```

- Check the current merge status:

```
systemd-sysext status
```

- Unmerge all currently installed system extension images from `/usr/` and `/opt/`:

```
systemd-sysext unmerge
```

- Refresh system extension images (a combination of `unmerge` and `merge`):

```
systemd-sysext refresh
```

systemd-sysusers

Create system users and groups.

If the config file is not specified, files in the **`sysusers.d`** directories are used.

More information: https://www.freedesktop.org/software/systemd/man/systemd-sysusers.html.

- Create users and groups from a specific configuration file:

```
systemd-sysusers {{path/to/file}}
```

- Process configuration files and print what would be done without actually doing anything:

```
systemd-sysusers --dry-run {{path/to/file}}
```

- Print the contents of all configuration files (before each file, its name is printed as a comment):

```
systemd-sysusers --cat-config
```

- Create users based on files listed in the previous command:

```
systemd-sysusers
```

systemd-tmpfiles

Create, delete and clean up volatile and temporary files and directories.

This command is automatically invoked on boot by systemd services, and running it manually is usually not needed.

More information: https://www.freedesktop.org/software/systemd/man/systemd-tmpfiles.html.

- Create files and directories as specified in the configuration:

```
systemd-tmpfiles --create
```

- Clean up files and directories with age parameters configured:

```
systemd-tmpfiles --clean
```

- Remove files and directories as specified in the configuration:

```
systemd-tmpfiles --remove
```

- Apply operations for user-specific configurations:

```
systemd-tmpfiles --create --user
```

- Execute lines marked for early boot:

```
systemd-tmpfiles --create --boot
```

systemd-tty-ask-password-agent

List or process pending systemd password requests.

More information: https://www.freedesktop.org/software/systemd/man/systemd-tty-ask-password-agent.html.

- List all currently pending system password requests:

```
systemd-tty-ask-password-agent --list
```

- Continuously process password requests:

```
systemd-tty-ask-password-agent --watch
```

- Process all currently pending system password requests by querying the user on the calling TTY:

```
systemd-tty-ask-password-agent --query
```

- Forward password requests to wall instead of querying the user on the calling TTY:

```
systemd-tty-ask-password-agent --wall
```

systemd-umount

This command is an alias of **`systemd-mount --umount`**.

- View documentation for the original command:

```
tldr systemd-mount
```

systemsettings

KDE central settings program.

More information: https://manned.org/systemsettings.

- Open the settings GUI:

  ```
  systemsettings
  ```

- List all possible KCM modules for `systemsettings`:

  ```
  systemsettings --list
  ```

- Display help:

  ```
  systemsettings {{[-h|--help]}}
  ```

systool

View system device information by bus, and classes.

This command is part of the **`sysfs`** package.

More information: https://manned.org/systool.

- List all attributes of devices of a bus (eg. `pci`, `usb`). View all buses using `ls /sys/bus`:

```
systool -b {{bus}} -v
```

- List all attributes of a class of devices (eg. `drm`, `block`). View all classes using `ls /sys/class`:

```
systool -c {{class}} -v
```

- Show only device drivers of a bus (eg. `pci`, `usb`):

```
systool -b {{bus}} -D
```

tailf

This command has been superseded by **`tail -f`**.

More information: https://manned.org/tailf.1.

- View documentation for the recommended replacement:

```
tldr tail
```

talk

A visual communication program which copies lines from your terminal to that of another user.

More information: https://www.gnu.org/software/inetutils/manual/inetutils.html#talk-invocation.

- Start a talk session with a user on the same machine:

```
talk {{username}}
```

- Start a talk session with a user on the same machine, who is logged in on tty3:

```
talk {{username}} {{tty3}}
```

- Start a talk session with a user on a remote machine:

```
talk {{username}}@{{hostname}}
```

- Clear text on both terminal screens:

```
<Ctrl d>
```

- Exit the talk session:

```
<Ctrl c>
```

taskset

Get or set a process' CPU affinity or start a new process with a defined CPU affinity.

More information: https://manned.org/taskset.

- Get a running process' CPU affinity by PID:

```
taskset {{[-p|--pid]}} {{[-c|--cpu-list]}} {{pid}}
```

- Set a running process' CPU affinity by PID:

```
taskset {{[-p|--pid]}} {{[-c|--cpu-list]}} {{cpu_id}} {{pid}}
```

- Start a new process with affinity for a single CPU:

```
taskset {{[-c|--cpu-list]}} {{cpu_id}} {{command}}
```

- Start a new process with affinity for multiple non-sequential CPUs:

```
taskset {{[-c|--cpu-list]}} {{cpu_id_1}},{{cpu_id_2}},
{{cpu_id_3}}
```

- Start a new process with affinity for CPUs 1 through 4:

```
taskset {{[-c|--cpu-list]}} {{cpu_id_1}}-{{cpu_id_4}}
```

tc

Show/manipulate traffic control settings.

More information: https://manned.org/tc.

- Add constant network delay to outbound packages:

```
sudo tc {{[q|qdisc]}} {{[a|add]}} dev {{eth0}} root netem delay {{delay_in_milliseconds}}ms
```

- Add normal distributed network delay to outbound packages:

```
sudo tc {{[q|qdisc]}} {{[a|add]}} dev {{eth0}} root netem delay {{mean_delay_ms}}ms {{delay_std_ms}}ms
```

- Add package corruption/loss/duplication to a portion of packages:

```
sudo tc {{[q|qdisc]}} {{[a|add]}} dev {{eth0}} root netem {{corruption|loss|duplication}} {{effect_percentage}}%
```

- Limit bandwidth, burst rate and max latency:

```
sudo tc {{[q|qdisc]}} {{[a|add]}} dev eth0 root tbf rate {{max_bandwidth_mb}}mbit burst {{max_burst_rate_kb}}kbit latency {{max_latency_before_drop_ms}}ms
```

- Show active traffic control policies:

```
tc {{[q|qdisc]}} {{[s|show]}} dev {{eth0}}
```

- Delete all traffic control rules:

```
sudo tc {{[q|qdisc]}} {{[d|delete]}} dev {{eth0}}
```

- Change traffic control rule:

```
sudo tc {{[q|qdisc]}} {{[c|change]}} dev {{eth0}} root netem {{policy}} {{policy_parameters}}
```

tcpflow

Capture TCP traffic for debugging and analysis.

More information: https://manned.org/tcpflow.

- Show all data on the given interface and port:

```
tcpflow -c -i {{eth0}} port {{80}}
```

tcpick

Packet sniffing and network traffic analysis tool.

It can capture and display TCP connections and data. It can also monitor network traffic on a interface, host, or port.

More information: https://manned.org/tcpick.8.

- Capture traffic on a specific interface, port and host::

```
sudo tcpick {{[-i|--interface]}} {{interface}} {{[-C|--colors]}} -h {{host}} -p {{port}}
```

- Capture traffic on port 80 (HTTP) of a specific host:

```
sudo tcpick {{[-i|--interface]}} {{eth0}} {{[-C|--colors]}} -h {{192.168.1.100}} -p {{80}}
```

- Display help:

```
tcpick --help
```

tcpkill

Kill specified in-progress TCP connections.

More information: https://manned.org/tcpkill.

- Kill in-progress connections at a specified interface, host and port:

```
tcpkill -i {{eth1}} host {{192.95.4.27}} and port {{2266}}
```

tcptraceroute

A traceroute implementation using TCP packets.

More information: https://manned.org/tcptraceroute.

- Trace the route to a host:

```
tcptraceroute {{host}}
```

- Specify the destination port and packet length in bytes:

```
tcptraceroute {{host}} {{destination_port}} {{packet_length}}
```

- Specify the local source port and source address:

```
tcptraceroute {{host}} -p {{source_port}} -s
{{source_address}}
```

- Set the first and maximum TTL:

```
tcptraceroute {{host}} -f {{first_ttl}} -m {{max_ttl}}
```

- Specify the wait time and number of queries per hop:

```
tcptraceroute {{host}} -w {{wait_time}} -q
{{number_of_queries}}
```

- Specify the interface:

```
tcptraceroute {{host}} -i {{interface}}
```

telinit

Change SysV runlevel.

Since the concept SysV runlevels is obsolete the runlevel requests will be transparently translated into systemd unit activation requests.

More information: https://manned.org/telinit.

- Power off the machine:

```
telinit 0
```

- Reboot the machine:

```
telinit 6
```

- Change SysV run level:

```
telinit {{2|3|4|5}}
```

- Change to rescue mode:

```
telinit 1
```

- Reload daemon configuration:

```
telinit q
```

- Do not send a wall message before reboot/power-off (6/0):

```
telinit --no-wall {{value}}
```

terminator

Arrange multiple GNOME terminals in one window.

More information: https://gnome-terminator.readthedocs.io/en/latest/advancedusage.html.

- Start `terminator` window:

  ```
  terminator
  ```

- Start with a fullscreen window:

  ```
  terminator {{[-f|--fullscreen]}}
  ```

- Split terminals horizontally:

  ```
  <Ctrl Shift o>
  ```

- Split terminals vertically:

  ```
  <Ctrl Shift e>
  ```

- Open new tab:

  ```
  <Ctrl Shift t>
  ```

termshark

Terminal UI for **`tshark`**, but looks more like **`wireshark`**.

More information: https://github.com/gcla/termshark/blob/master/docs/UserGuide.md.

- Monitor the default network interface:

  ```
  sudo termshark
  ```

- Specify the interface to monitor:

  ```
  sudo termshark {{interface}}
  ```

termusic

A terminal music player written in Rust that uses vim-like key bindings.

See also: **`cmus`**, **`ncmpcpp`**, **`audacious`**.

More information: https://github.com/tramhao/termusic.

- Open termusic to a specific directory. (It can be set permanently in `~/.config/termusic/config.toml`):

```
termusic {{path/to/directory}}
```

- Disable showing the album cover for a specific file:

```
termusic -c {{path/to/music_file}}
```

- Display help:

```
termusic --help
```

Terraria

Create and start a headless Terraria server.

More information: https://terraria.wiki.gg/wiki/Server.

- Start an interactive server setup:

```
{{path/to/TerrariaServer}}
```

- Start a Terraria server:

```
{{path/to/TerrariaServer}} -world {{path/to/world.wld}}
```

tftp

Trivial File Transfer Protocol client.

More information: https://manned.org/tftp.1.

- Connect to a TFTP server specifying its IP address and port:

```
tftp {{server_ip}} {{port}}
```

- Connect to a TFTP server and execute a TFTP [c]ommand:

```
tftp {{server_ip}} -c {{command}}
```

- Connect to a TFTP server using IPv6 and force originating port to be in [R]ange:

```
tftp {{server_ip}} -6 -R {{port}}:{{port}}
```

- Set the transfer mode to binary or ASCIi through the tftp client:

```
mode {{binary|ascii}}
```

- Download file from a server through the tftp client:

```
get {{file}}
```

- Upload file to a server through the tftp client:

```
put {{file}}
```

- Exit the tftp client:

```
quit
```

thunar

Graphical file manager for XFCE desktop environments.

See also: **`caja`**, **`dolphin`**, **`nautilus`**, **`mc`**.

More information: https://docs.xfce.org/xfce/thunar/start.

- Open a new window showing the current directory:

```
thunar
```

- Open the bulk rename utility:

```
thunar --bulk-rename
```

- Close all open thunar windows:

```
thunar --quit
```

tic

Compile terminfo and install for ncurses.

More information: https://manned.org/tic.

- Compile and install terminfo for a terminal:

```
tic -xe {{terminal}} {{path/to/terminal.info}}
```

- Check terminfo file for errors:

```
tic -c {{path/to/terminal.info}}
```

- Print database locations:

```
tic -D
```

timedatectl

Control the system time and date.

More information: https://www.freedesktop.org/software/systemd/man/timedatectl.html.

- Check the current system clock time:

```
timedatectl
```

- Set the local time of the system clock directly:

```
timedatectl set-time "{{yyyy-MM-dd hh:mm:ss}}"
```

- List available timezones:

```
timedatectl list-timezones
```

- Set the system timezone:

```
timedatectl set-timezone {{timezone}}
```

- Enable Network Time Protocol (NTP) synchronization:

```
timedatectl set-ntp on
```

- Change the hardware clock time standard to localtime:

```
timedatectl set-local-rtc 1
```

timeshift

System restore utility.

More information: https://github.com/linuxmint/timeshift.

- List snapshots:

```
sudo timeshift --list
```

- Create a new snapshot (if scheduled):

```
sudo timeshift --check
```

- Create a new snapshot (even if not scheduled):

```
sudo timeshift --create
```

- Restore a snapshot (selecting which snapshot to restore interactively):

```
sudo timeshift --restore
```

- Restore a specific snapshot:

```
sudo timeshift --restore --snapshot '{{snapshot}}'
```

- Delete a specific snapshot:

```
sudo timeshift --delete --snapshot '{{snapshot}}'
```

tlp-stat

Generate TLP status reports.

See also: **`tlp`**.

More information: https://linrunner.de/tlp/usage/tlp-stat.

- Generate status report with configuration and all active settings:

```
sudo tlp-stat
```

- Show information about various devices:

```
sudo tlp-stat --{{battery|disk|processor|graphics|pcie|rfkill|usb}}
```

- Show verbose information about devices that support verbosity:

```
sudo tlp-stat {{[-v|--verbose]}} --{{battery|processor|pcie|usb}}
```

- Show configuration:

```
sudo tlp-stat {{[-c|--config]}}
```

- Monitor [p]ower supply udev [ev]ents:

```
sudo tlp-stat {{[-P|--pev]}}
```

- Show [p]ower [sup]ply diagonistics:

```
sudo tlp-stat --psup
```

- Show [t]emperatures and fan speed:

```
sudo tlp-stat {{[-t|--temp]}}
```

- Show general system information:

```
sudo tlp-stat {{[-s|--system]}}
```

tlp

Advanced power management for Linux.

See also: **`tlp-stat`**.

More information: https://linrunner.de/tlp/usage/tlp.html.

- Apply settings (according to the actual power source):

```
sudo tlp start
```

- Apply battery settings (ignoring the actual power source):

```
sudo tlp bat
```

- Apply AC settings (ignoring the actual power source):

```
sudo tlp ac
```

tmt run

Execute **tmt** test steps. By default, all steps are run.

More information: https://tmt.readthedocs.io/en/stable/stories/cli.html#run.

- Run all test steps for each plan:

```
tmt run
```

- Run only the discover step to show what tests would be run:

```
tmt run discover {{[-v|--verbose]}}
```

- Run all steps and adjust the provision step options:

```
tmt run {{[-a|--all]}} provision {{[-h|--how]}} {{container}}
{{[-i|--image]}} {{fedora:rawhide}}
```

- Run only selected plans and tests:

```
tmt run plan {{[-n|--name]}} {{/plan/name}} test {{[-n|--
name]}} {{/test/name}}
```

- Show results from the last run in a web browser:

```
tmt run {{[-l|--last]}} report {{[-h|--how]}} {{html}} {{[-
o|--open]}}
```

- Run tests with the provided context:

```
tmt run {{[-c|--context]}} {{key=value}} {{[-c|--context]}}
{{distro=fedora}}
```

- Run tests interactively (debug test code in the middle of a test):

```
tmt run {{[-a|--all]}} execute {{[-h|--how]}} {{tmt}} --
interactive
```

- Use dry mode to see what actions would happen and use the highest verbosity:

```
tmt run {{[-n|--dry]}} {{[-vvv|--verbose --verbose --
verbose]}}
```

tmt try

Quickly experiment with tests and environments.

More information: https://tmt.readthedocs.io/en/stable/stories/cli.html#try.

- Quickly experiment with the default provision method (no tests in the CWD):

```
tmt try
```

- Run a test in the current working directory:

```
cd {{path/to/test_directory}} && tmt try
```

- Use a specific operating system:

```
tmt try {{rhel-9}}
```

- Select both custom image and provision method:

```
tmt try {{fedora@container}}
```

- Select tests with custom filter:

```
tmt try {{[-t|--test]}} {{feature}}
```

- Provision guest and wait for instructions:

```
tmt try {{[-a|--ask]}}
```

- Directly log into the guest without asking:

```
tmt try {{[-l|--login]}}
```

- Display help:

```
tmt try --help
```

tmt

Test Management Tool for creating, running, and debugging tests.

Some subcommands such as **`run`**, **`try`**, etc. have their own usage documentation.

More information: https://tmt.readthedocs.io/en/stable/examples.html.

- List available tests, plans, and stories:

```
tmt
```

- Initialize tmt files/project structure:

```
tmt init
```

- Create a new test with a template and a link:

```
tmt test create {{[-t|--template]}} {{beakerlib}} --link
{{verifies:issue#1234}}
```

- List available tests, plans, or stories:

```
tmt {{test|plan|story}} ls {{pattern}}
```

- Show detailed test metadata in the given context:

```
tmt {{[-c|--context]}} {{arch=aarch64}} test show
```

- Validate tmt files against the specification:

```
tmt lint
```

- Use filter:

```
tmt tests ls {{[-f|--filter]}} {{tag:foo}} {{[-f|--filter]}}
{{tier:0}}
```

- Display help:

```
tmt --help
```

tod

A tiny Todoist client in Rust.

It takes simple input and dumps it in your inbox or another project. Taking advantage of natural language processing to assign due dates, tags, etc.

More information: https://github.com/alanvardy/tod.

- Import your projects (this is necessary to enable project prompts):

```
tod project import
```

- Quickly create a task with due date:

```
tod --quickadd {{Buy more milk today}}
```

- Create a new task (you will be prompted for content and project):

```
tod task create
```

- Create a task in a project:

```
tod task create --content "{{Write more rust}}" --project
{{code}}
```

- Get the next task for a project:

```
tod task next
```

- Get your work schedule:

```
tod task list --scheduled --project {{work}}
```

- Get all tasks for work:

```
tod task list --project {{work}}
```

togglesebool

Flip the current (non-persistent) values of SELinux booleans.

Note: This tool has been deprecated and often removed in favor of **`setsebool`**.

More information: https://manned.org/togglesebool.

- Flip the current (non-persistent) values of the specified booleans:

```
sudo togglesebool {{virt_use_samba virt_use_usb ...}}
```

toilet

Display ASCII-art fonts.

More information: http://caca.zoy.org/wiki/toilet.

- Generate ASCII art for a given text:

```
toilet {{input_text}}
```

- Generate ASCII art using a custom font file:

```
toilet {{input_text}} {{[-f|--font]}} {{font_filename}}
```

- Generate ASCII art using a filter:

```
toilet {{input_text}} {{[-F|--filter]}} {{filter_name}}
```

- Show available toilet filters:

```
toilet {{[-F|--filter]}} list
```

tomb

Manage encrypted storage directories that can be safely transported and hidden in a filesystem.

More information: https://dyne.org/docs/tomb/manpage/.

- Create a new tomb with an initial size of 100 MB:

```
tomb dig -s {{100}} {{encrypted_directory.tomb}}
```

- Create a new key file that can be used to lock a tomb; user will be prompted for a password for the key:

```
tomb forge {{encrypted_directory.tomb.key}}
```

- Forcefully create a new key, even if the tomb isn't allowing key forging (due to swap):

```
tomb forge {{encrypted_directory.tomb.key}} -f
```

- Initialize and lock an empty tomb using a key made with `forge`:

```
tomb lock {{encrypted_directory.tomb}} -k
{{encrypted_directory.tomb.key}}
```

- Mount a tomb (by default in `/media`) using its key, making it usable as a regular filesystem directory:

```
tomb open {{encrypted_directory.tomb}} -k
{{encrypted_directory.tomb.key}}
```

- Close a tomb (fails if the tomb is being used by a process):

```
tomb close {{encrypted_directory.tomb}}
```

- Forcefully close all open tombs, killing any applications using them:

```
tomb slam all
```

- List all open tombs:

```
tomb list
```

toolbox create

Create a new **`toolbox`** container.

More information: https://manned.org/toolbox-create.1.

- Create a `toolbox` container for a specific distribution:

```
toolbox create {{[-d|--distro]}} {{distribution}}
```

- Create a `toolbox` container for a specific release of the current distribution:

```
toolbox create {{[-r|--release]}} {{release}}
```

- Create a `toolbox` container with a custom image:

```
toolbox create {{[-i|--image]}} {{name}}
```

- Create a `toolbox` container from a custom Fedora image:

```
toolbox create {{[-i|--image]}} {{quay.io/fedora/fedora:tag}}
```

- Create a `toolbox` container using the default image for a specific Fedora release:

```
toolbox create {{[-d|--distro]}} {{fedora}} {{[-r|--release]}} f{{version}}
```

toolbox enter

Enter a **`toolbox`** container for interactive use.

See also: **`toolbox run`**.

More information: https://manned.org/toolbox-enter.1.

- Enter a `toolbox` container using the default image of a specific distribution:

  ```
  toolbox enter {{[-d|--distro]}} {{distribution}}
  ```

- Enter a `toolbox` container using the default image of a specific release of the current distribution:

  ```
  toolbox enter {{[-r|--release]}} {{release}}
  ```

- Enter a toolbox container using the default image for a specific Fedora release:

  ```
  toolbox enter {{[-d|--distro]}} {{fedora}} {{[-r|--release]}} f{{version}}
  ```

toolbox help

Display help information about **`toolbox`**.

More information: https://manned.org/toolbox-help.1.

- Display the `toolbox` manual:

  ```
  toolbox help
  ```

- Display the `toolbox` manual for a specific subcommand:

  ```
  toolbox help {{subcommand}}
  ```

toolbox init-container

Initialize a running **`toolbox`** container.

This command should not be executed by the user, and cannot be run on the host.

More information: https://manned.org/toolbox-init-container.1.

- Initialize a running toolbox:

```
toolbox init-container --gid {{gid}} --home {{home}} --home-
link --media-link --mnt-link --monitor-host --shell {{shell}}
--uid {{uid}} --user {{user}}
```

toolbox list

List existing **`toolbox`** containers and images.

More information: https://manned.org/toolbox-list.1.

- List all `toolbox` containers and images:

  `toolbox list`

- List only `toolbox` containers:

  `toolbox list {{[-c|--containers]}}`

- List only `toolbox` images:

  `toolbox list {{[-i|--images]}}`

toolbox rm

Remove one or more **`toolbox`** containers.

See also: **`toolbox rmi`**.

More information: https://manned.org/toolbox-rm.1.

- Remove a toolbox container:

```
toolbox rm {{container_name}}
```

- Remove all `toolbox` containers:

```
toolbox rm {{[-a|--all]}}
```

- Force the removal of a currently active `toolbox` container:

```
toolbox rm {{[-f|--force]}} {{container_name}}
```

toolbox rmi

Remove **`toolbox`** images.

See also: **`toolbox rm`**.

More information: https://manned.org/toolbox-rmi.1.

- Remove one or more `toolbox` image:

  ```
  toolbox rmi {{image_name1 image_name2 ...}}
  ```

- Remove all `toolbox` images:

  ```
  toolbox rmi {{[-a|--all]}}
  ```

- Force the removal of a `toolbox` image which is currently being used by a container (the container will be removed as well):

  ```
  toolbox rmi {{[-f|--force]}} {{image_name}}
  ```

toolbox run

Run a command in an existing **`toolbox`** container.

See also: **`toolbox enter`**.

More information: https://manned.org/toolbox-run.

- Run a command inside a specific `toolbox` container:

```
toolbox run {{[-c|--container]}} {{container_name}} {{command}}
```

- Run a command inside a `toolbox` container for a specific release of a distribution:

```
toolbox run {{[-d|--distro]}} {{distribution}} {{[-r|--release]}} {{release}} {{command}}
```

- Run `emacs` inside a `toolbox` container using the default image for a specific Fedora release:

```
toolbox run {{[-d|--distro]}} {{fedora}} {{[-r|--release]}} f{{version}} {{emacs}}
```

toolbox

Manage containerized command-line environments on Linux.

Some subcommands such as **`create`** have their own usage documentation.

More information: https://manned.org/toolbox.1.

- Run a `toolbox` subcommand:

```
toolbox {{subcommand}}
```

- Display help for a specific subcommand (such as `create`, `enter`, `rm`, `rmi`, etc.):

```
toolbox help {{subcommand}}
```

- Display help:

```
toolbox {{[-h|--help]}}
```

- Display version:

```
toolbox --version
```

top

Display dynamic real-time information about running processes.

See also: **`htop`**, **`atop`**, **`glances`**, **`btop`**, **`btm`**.

More information: https://manned.org/top.

- Start `top`:

```
top
```

- Do not show any idle or zombie processes:

```
top {{[-i|--idle-toggle]}}
```

- Show only processes owned by given user:

```
top {{[-u|--filter-only-euser]}} {{username}}
```

- Sort processes by a field:

```
top {{[-o|--sort-override]}} {{field_name}}
```

- Show the individual threads of a given process:

```
top {{[-Hp|--threads-show --pid]}} {{process_id}}
```

- Show only the processes with the given PID(s), passed as a comma-separated list. (Normally you wouldn't know PIDs off hand. This example picks the PIDs from the process name):

```
top {{[-p|--pid]}} $(pgrep {{[-d|--delimiter]}}
',' {{process_name}})
```

- Display help about interactive commands:

```
<?>
```

tor

Enable anonymous communication through the Tor network.

More information: https://manned.org/tor.

- Connect to the Tor network:

```
tor
```

- View Tor configuration:

```
tor --config
```

- Check Tor status:

```
tor --status
```

- Run as client only:

```
tor --client
```

- Run as relay:

```
tor --relay
```

- Run as bridge:

```
tor --bridge
```

- Run as a hidden service:

```
tor --hidden-service
```

torify

Route network traffic through the Tor network.

Note: This command has been deprecated, and is now a backwards-compatible wrapper of **`torsocks`**.

More information: https://manned.org/torify.

- Route traffic via Tor:

```
torify {{command}}
```

- Toggle Tor in shell:

```
torify {{on|off}}
```

- Spawn a Tor-enabled shell:

```
torify --shell
```

- Check for a Tor-enabled shell:

```
torify show
```

- Specify Tor configuration file:

```
torify -c {{config-file}} {{command}}
```

- Use a specific Tor SOCKS proxy:

```
torify -P {{proxy}} {{command}}
```

- Redirect output to a file:

```
torify {{command}} > {{path/to/output}}
```

torsocks

Route the traffic of any application through the Tor network.

Note: **`torsocks`** will assume that it should connect to the Tor SOCKS proxy running at 127.0.0.1:9050 being the defaults of the Tor daemon.

More information: https://manned.org/torsocks.

- Run a command using Tor:

```
torsocks {{command}}
```

- Enable or disable Tor in this shell:

```
. torsocks {{on|off}}
```

- Spawn a new Tor enabled shell:

```
torsocks --shell
```

- Check if current shell is Tor enabled (`LD_PRELOAD` value will be empty if disabled):

```
torsocks show
```

- Isolate traffic through a different Tor circuit, improving anonymity:

```
torsocks {{[-i|--isolate]}} {{curl https://
check.torproject.org/api/ip}}
```

- Connect to a Tor proxy running on a specific address and port:

```
torsocks {{[-a|--address]}} {{ip}} {{[-P|--port]}} {{port}}
{{command}}
```

trace-cmd list

Show available tracers, events, or options for tracing.

See also: **`trace-cmd record`**, **`trace-cmd report`**.

More information: https://manned.org/trace-cmd-list.

- List available tracers:

```
sudo trace-cmd list -t
```

- List available plugins:

```
sudo trace-cmd list -p
```

- List available events:

```
sudo trace-cmd list -e
```

- List available functions:

```
sudo trace-cmd list -f
```

trace-cmd record

Capture kernel trace events.

See also: **`trace-cmd list`**, **`trace-cmd report`**.

More information: https://manned.org/trace-cmd-record.

- Record a trace with a specific plugin:

```
sudo trace-cmd record -p {{plugin}}
```

- Record a trace of a specific executable:

```
sudo trace-cmd record -F {{executable}}
```

- Record a trace of a specific function:

```
sudo trace-cmd record -g {{function}}
```

- Exclude a specific function from the trace:

```
sudo trace-cmd record -n {{function}}
```

- Limit the function call graph depth:

```
sudo trace-cmd record --max-graph-depth {{depth}}
```

- Record a trace from a specific process ID:

```
sudo trace-cmd record -P {{pid}}
```

trace-cmd report

Display recorded trace events.

See also: **`trace-cmd list`**, **`trace-cmd record`**.

More information: https://manned.org/trace-cmd-report.

- Display the recorded trace:

```
sudo trace-cmd report
```

- Display the recorded trace for a specific CPU:

```
sudo trace-cmd report --cpu {{cpu_number}}
```

trace-cmd

Utility to interact with the Ftrace Linux kernel internal tracer.

See also: **`trace-cmd list`**, **`trace-cmd record`**, **`trace-cmd report`**.

More information: https://manned.org/trace-cmd.

- Display the status of tracing system:

```
sudo trace-cmd stat
```

- List available tracers:

```
sudo trace-cmd list -t
```

- Start tracing with a specific plugin:

```
sudo trace-cmd start -p {{function|function_graph|
preemptirqsoff|irqsoff|preemptoff|wakeup|...}}
```

- View the trace output:

```
sudo trace-cmd show
```

- Stop the tracing but retain the buffers:

```
sudo trace-cmd stop
```

- Clear the trace buffers:

```
sudo trace-cmd clear
```

- Record a trace:

```
sudo trace-cmd record
```

- Display the recorded trace:

```
sudo trace-cmd report
```

tracepath

Trace the path to a network host discovering MTU along this path.

More information: https://manned.org/tracepath.

- A preferred way to trace the path to a host:

```
tracepath -p {{33434}} {{host}}
```

- Specify the initial destination port, useful with non-standard firewall settings:

```
tracepath -p {{destination_port}} {{host}}
```

- Print both hostnames and numerical IP addresses:

```
tracepath -b {{host}}
```

- Specify a maximum TTL (number of hops):

```
tracepath -m {{max_hops}} {{host}}
```

- Specify the initial packet length (defaults to 65535 for IPv4 and 128000 for IPv6):

```
tracepath -l {{packet_length}} {{host}}
```

- Use only IPv6 addresses:

```
tracepath -6 {{host}}
```

trap

Execute a command upon an event.

More information: https://www.gnu.org/software/bash/manual/bash.html#index-trap.

- List the available event names (e.g. `SIGWINCH`):

```
trap -l
```

- List the commands and the names of the expected events:

```
trap
```

- Execute a command when a signal is received:

```
trap 'echo "Caught signal {{SIGHUP}}"' {{SIGHUP}}
```

- Remove commands:

```
trap - {{SIGHUP}} {{SIGINT}}
```

- Ignore a signal:

```
trap '' {{SIGINT}}
```

trash-empty

This command has been moved to **`trash`**.

- View documentation for `trash-empty`:

  ```
  tldr trash
  ```

trash-list

This command has been moved to **`trash`**.

- View documentation for `trash-list`:

  ```
  tldr trash
  ```

trash-put

This command is an alias of **`trash`**.

- View documentation for the original command:

```
tldr trash
```

trash-restore

This command has been moved to **`trash`**.

- View documentation for `trash-restore`:

  ```
  tldr trash
  ```

trash-rm

This command has been moved to **`trash`**.

- View documentation for `trash-rm`:

  ```
  tldr trash
  ```

trash

Manage the trashcan/recycling bin.

More information: https://github.com/andreafrancia/trash-cli.

- Send a file to the trash:

```
trash {{path/to/file}}
```

- List all files in the trash:

```
trash-list
```

- Interactively restore a file from the trash:

```
trash-restore
```

- Empty the trash:

```
trash-empty
```

- Permanently delete all files in the trash which are older than 10 days:

```
trash-empty 10
```

- Remove all files in the trash, which match a specific blob pattern:

```
trash-rm "{{*.o}}"
```

- Remove all files with a specific original location:

```
trash-rm /{{path/to/file_or_directory}}
```

trashy

An alternative to **`rm`** and **`trash-cli`** written in Rust.

More information: https://github.com/oberblastmeister/trashy.

- Move a specific file to the trash:

```
trash {{path/to/file}}
```

- Move specific files to the trash:

```
trash {{path/to/file1 path/to/file2 ...}}
```

- List items in the trash:

```
trash list
```

- Restore a specific file from the trash:

```
trash restore {{file}}
```

- Remove a specific file from the trash:

```
trash empty {{file}}
```

- Restore all files from the trash:

```
trash restore --all
```

- Remove all files from the trash:

```
trash empty --all
```

trayer

A lightweight GTK-2 based systray.

More information: https://github.com/sargon/trayer-srg.

- Run `trayer`:

```
trayer
```

- Position `trayer` to a specific edge:

```
trayer --edge {{left|right|top|bottom}}
```

- Provide a specific height and width of the panel (in pixels):

```
trayer --width {{10}} --height {{32}}
```

- Provide the width of the panel in pixels or percentages:

```
trayer --widthtype {{pixel|percent}} --width {{72}}
```

- Align `trayer` to a specific direction:

```
trayer --align {{left|center|right}}
```

- Provide spacing between icons (in pixels):

```
trayer --iconspacing {{10}}
```

treetime

TreeTime provides routines for ancestral sequence reconstruction and inference of molecular-clock phylogenies.

More information: https://treetime.readthedocs.io/en/latest/tutorials.html.

- Infer ancestral sequences maximizing the joint or marginal likelihood:

  ```
  treetime ancestral
  ```

- Analyze patterns of recurrent mutations aka homoplasies:

  ```
  treetime homoplasy
  ```

- Estimate molecular clock parameters and reroot the tree:

  ```
  treetime clock
  ```

- Map discrete character such as host or country to the tree:

  ```
  treetime mugration
  ```

trizen

Arch Linux utility for building packages from the Arch User Repository (AUR).

More information: https://github.com/trizen/trizen/blob/master/TRIZEN.md.

- Synchronize and update all AUR packages:

```
trizen -Syua
```

- Install a new package:

```
trizen -S {{package}}
```

- Remove a package and its dependencies:

```
trizen -Rs {{package}}
```

- Search the package database for a keyword:

```
trizen -Ss {{keyword}}
```

- Show information about a package:

```
trizen -Si {{package}}
```

- List installed packages and versions:

```
trizen -Qe
```

trust

Operate on the trust policy store.

More information: https://manned.org/trust.

- List trust policy store items:

```
trust list
```

- List information about specific items in the trust policy store:

```
trust list --filter={{blocklist|ca-anchors|certificates|
trust-policy}}
```

- Store a specific trust anchor in the trust policy store:

```
trust anchor {{path/to/certificate.crt}}
```

- Remove a specific anchor from the trust policy store:

```
trust anchor --remove {{path/to/certificate.crt}}
```

- Extract trust policy from the shared trust policy store:

```
trust extract --format=x509-directory --filter=ca-anchors
{{path/to/directory}}
```

- Display help for a subcommand:

```
trust {{subcommand}} --help
```

tshark

Packet analysis tool, CLI version of Wireshark.

More information: https://tshark.dev/.

- Monitor everything on localhost:

```
tshark
```

- Only capture packets matching a specific capture filter:

```
tshark -f '{{udp port 53}}'
```

- Only show packets matching a specific output filter:

```
tshark -Y '{{http.request.method == "GET"}}'
```

- Decode a TCP port using a specific protocol (e.g. HTTP):

```
tshark -d tcp.port=={{8888}},{{http}}
```

- Specify the format of captured output:

```
tshark -T {{json|text|ps|...}}
```

- Select specific fields to output:

```
tshark -T {{fields|ek|json|pdml}} -e {{http.request.method}}
-e {{ip.src}}
```

- Write captured packet to a file:

```
tshark -w {{path/to/file}}
```

- Analyze packets from a file:

```
tshark -r {{path/to/file.pcap}}
```

ttyplot

A realtime plotting utility for the command-line with data input from **`stdin`**.

More information: https://github.com/tenox7/ttyplot.

- Plot the values 1, 2 and 3 (Note: `cat` prevents `ttyplot` from exiting):

```
{ echo {{1 2 3}}; cat; } | ttyplot
```

- Set a specific title and unit:

```
{ echo {{1 2 3}}; cat; } | ttyplot -t {{title}} -u {{unit}}
```

- Use a while loop to continuously plot random values:

```
{ while {{true}}; do echo {{$RANDOM}}; sleep {{1}}; done } |
ttyplot
```

- Parse the output from `ping` and visualize it:

```
ping {{8.8.8.8}} | sed -u '{{s/^.*time=//g; s/ ms//g}}' |
ttyplot -t "{{ping to 8.8.8.8}}" -u {{ms}}
```

tune.exfat

Adjust tunable filesystem parameters on an exFAT filesystem.

More information: https://manned.org/tune.exfat.

- Print the volume label of a filesystem:

```
tune.exfat {{[-l|--print-label]}} {{/dev/sdXY}}
```

- Set the volume label of a filesystem:

```
tune.exfat {{[-L|--set-label]}} {{new_label}} {{/dev/sdXY}}
```

- Print the volume GUID of a filesystem:

```
tune.exfat {{[-u|--print-guid]}} {{/dev/sdXY}}
```

- Set the volume GUID of a filesystem:

```
tune.exfat {{[-U|--set-guid]}} {{new_guid}} {{/dev/sdXY}}
```

- Print the volume serial of a filesystem:

```
tune.exfat {{[-i|--print-serial]}} {{/dev/sdXY}}
```

- Set the volume serial of a filesystem:

```
tune.exfat {{[-I|--set-serial]}} {{new_serial}} {{/dev/sdXY}}
```

tune2fs

Adjust parameters of an ext2, ext3 or ext4 filesystem.

May be used on mounted filesystems.

More information: https://manned.org/tune2fs.

- Set the max number of counts before a filesystem is checked to 2:

```
sudo tune2fs -c 2 {{/dev/sdXN}}
```

- Set the filesystem label to MY_LABEL:

```
sudo tune2fs -L 'MY_LABEL' {{/dev/sdXN}}
```

- Enable discard and user-specified extended attributes for a filesystem:

```
sudo tune2fs -o discard,user_xattr {{/dev/sdXN}}
```

- Enable journaling for a filesystem:

```
sudo tune2fs -o has_journal {{/dev/sdXN}}
```

- Assign a new randomly-generated UUID to a filesystem:

```
sudo tune2fs -U random {{/dev/sdXN}}
```

tuned-adm

Manage and optimize system performance tuning profiles on Linux.

More information: https://tuned-project.org.

- List available profiles:

```
tuned-adm list
```

- Show the currently active profile:

```
tuned-adm active
```

- Set a specific tuning profile:

```
tuned-adm profile {{profile_name}}
```

- Recommend a suitable profile based on the current system:

```
tuned-adm recommend
```

- Disable tuning:

```
tuned-adm off
```

tunelp

Set various parameters for parallel port devices for troubleshooting or for better performance.

Part of **`util-linux`**.

More information: https://manned.org/tunelp.

- Check the status of a parallel port device:

```
tunelp {{[-s|--status]}} {{/dev/lp0}}
```

- Reset a given parallel port:

```
tunelp {{[-r|--reset]}} {{/dev/lp0}}
```

- Use a given IRQ for a device, each one representing an interrupt line:

```
tunelp {{[-i|--irq]}} 5 {{/dev/lp0}}
```

- Try a given number of times to output a character to the printer before sleeping for a given time:

```
tunelp {{[-c|--chars]}} {{times}} {{[-t|--time]}} {{time_in_centiseconds}} {{/dev/lp0}}
```

- Enable or disable aborting on error (disabled by default):

```
tunelp {{[-a|--abort]}} {{on|off}}
```

turbostat

Report processor topology, frequency, temperature, power, and idle statistics.

More information: https://manned.org/turbostat.8.

- Display statistics every 5 seconds:

```
sudo turbostat
```

- Display statistics every specified amount of seconds:

```
sudo turbostat {{[-i|--interval]}} {{n_seconds}}
```

- Do not decode and print the system configuration header information:

```
sudo turbostat --quiet
```

- Display useful information about CPU every 1 second, without header information:

```
sudo turbostat --quiet {{[-i|--interval]}} 1 --cpu 0-
{{CPU_thread_count}} --show
"PkgWatt","Busy%","Core","CoreTmp","Thermal"
```

- Display help:

```
turbostat --help
```

tuxi

Scrape Google search results and SERPs and provide instant and concise answers.

More information: https://github.com/Bugswriter/tuxi#usage.

- Make a search using Google:

```
tuxi {{search_terms}}
```

- Display the search results in [r]aw format (no pretty output, no colors):

```
tuxi -r {{search_terms}}
```

- Display only search results (silences "Did you mean?", greetings and usage):

```
tuxi -q {{search_terms}}
```

- Display help:

```
tuxi -h
```

tzselect

Interactively select timezone.

Note: This program doesn't actually set the timezone.

More information: https://manned.org/tzselect.

- Open the interactive menu for timezone selection and print the selected timezone to `stdout`:

```
tzselect
```

- Ask for nearest timezone to coordinates in ISO 6709 notation:

```
tzselect -c {{coordinates}}
```

ubuntu-bug

This command is an alias of **apport-bug**.

- View documentation for the original command:

```
tldr apport-bug
```

ubuntu-drivers

Install drivers on Ubuntu.

More information: https://documentation.ubuntu.com/server/how-to/graphics/install-nvidia-drivers/index.html.

- List available drivers for the current hardware:

```
sudo ubuntu-drivers list
```

- Install drivers for detected hardware:

```
sudo ubuntu-drivers install
```

- Display help:

```
ubuntu-drivers
```

ubuntu-security-status

Display information about security support for installed Ubuntu packages.

More information: https://git.launchpad.net/ubuntu/+source/update-manager/tree/ubuntu-security-status.

- Display the number of unsupported packages:

```
ubuntu-security-status
```

- List packages that are no longer available for download:

```
ubuntu-security-status --unavailable
```

- List third-party packages:

```
ubuntu-security-status --thirdparty
```

ubus

Interact with an OpenWrt ubusd server.

More information: https://openwrt.org/docs/techref/ubus.

- List available objects:

```
ubus list
```

- Retrieve system information in JSON format:

```
ubus call system board
```

- Listen to events:

```
ubus subscribe {{event_name}}
```

- Display help:

```
ubus
```

uci

Manage OpenWrt configuration files.

More information: https://openwrt.org/docs/techref/uci.

- Fetch a value:

```
uci get {{network.lan.ipaddr}}
```

- List all options and their values:

```
uci show {{network}}
```

- Set a value:

```
uci set {{config}}.{{section}}.{{option}}={{value}}
```

- Add a new section:

```
uci add {{config}} {{section}}
```

- Delete a section or value:

```
uci delete {{config}}.{{section}}.{{option}}
```

- Commit changes:

```
uci commit {{config}}
```

- Discard uncommitted changes:

```
uci revert {{config}}
```

- Display help:

```
uci
```

udevadm

Linux **udev** management tool.

More information: https://www.freedesktop.org/software/systemd/man/udevadm.html.

- Monitor all device events:

```
sudo udevadm monitor
```

- Print `uevents` sent out by the kernel:

```
sudo udevadm monitor {{[-k|--kernel]}}
```

- Print device events after being processed by `udev`:

```
sudo udevadm monitor {{[-u|--udev]}}
```

- List attributes of device `/dev/sda`:

```
sudo udevadm info {{[-a|--attribute-walk]}} {{/dev/sda}}
```

- Reload all `udev` rules:

```
sudo udevadm control {{[-R|--reload]}}
```

- Trigger all `udev` rules to run:

```
sudo udevadm trigger
```

- Test an event run by simulating loading of `/dev/sda`:

```
sudo udevadm test {{/dev/sda}}
```

udevmon

Intercept and monitor input devices for launching tasks.

Filters or modifies events according to configuration file(s) (default: **`/etc/interception/udevmon.d/*.yaml`**).

More information: https://gitlab.com/interception/linux/tools/-/tree/master#udevmon.

- Start udevmon with specified configuration file:

```
udevmon -c {{path/to/config.yaml}}
```

udisksctl

Interact with **`udisksd`** to query and manipulate storage devices.

More information: https://storaged.org/doc/udisks2-api/latest/udisksctl.1.html.

- Show high-level information about disk drives and block devices:

```
udisksctl status
```

- Show detailed information about a device:

```
udisksctl info {{[-b|--block-device]}} {{/dev/sdX}}
```

- Show detailed information about a device partition:

```
udisksctl info {{[-b|--block-device]}} {{/dev/sdXN}}
```

- Mount a device partition and prints the mount point:

```
udisksctl mount {{[-b|--block-device]}} {{/dev/sdXN}}
```

- Unmount a device partition:

```
udisksctl unmount {{[-b|--block-device]}} {{/dev/sdXN}}
```

- Power off a device to safely remove it:

```
udisksctl power-off {{[-b|--block-device]}} {{/dev/sdX}}
```

- Monitor the daemon for events:

```
udisksctl monitor
```

ufw

Uncomplicated Firewall.

Frontend for **`iptables`** aiming to make configuration of a firewall easier.

More information: https://wiki.ubuntu.com/UncomplicatedFirewall.

- Enable `ufw`:

  ```
  sudo ufw enable
  ```

- Disable `ufw`:

  ```
  sudo ufw disable
  ```

- Show `ufw` rules, along with their numbers:

  ```
  sudo ufw status numbered
  ```

- Allow incoming traffic on port 5432 on this host with a comment identifying the service:

  ```
  sudo ufw allow 5432 comment "{{Service}}"
  ```

- Allow only TCP traffic from 192.168.0.4 to any address on this host, on port 22:

  ```
  sudo ufw allow proto tcp from 192.168.0.4 to any port 22
  ```

- Deny traffic on port 80 on this host:

  ```
  sudo ufw deny 80
  ```

- Deny all UDP traffic to ports in range 8412:8500:

  ```
  sudo ufw deny proto udp from any to any port 8412:8500
  ```

- Delete a particular rule. The rule number can be retrieved from the `ufw status numbered` command:

  ```
  sudo ufw delete {{rule_number}}
  ```

uinput

Intercept and write input events to a virtual keyboard device using /dev/uinput.

More information: https://gitlab.com/interception/linux/tools/-/tree/master#uinput.

- Show resulting YAML device description merge and exit (dry-run):

```
uinput -p
```

- Merge YAML device description(s) to resulting virtual device:

```
sudo uinput -c {{path/to/device1.yaml path/to/device2.yaml ...}}
```

- Merge reference device description from device node(s) to resulting virtual device:

```
sudo uinput -d {{/dev/input/eventX /dev/input/eventY ...}}
```

ul

Underline a text.

Each character in a string must be underlined separately.

More information: https://manned.org/ul.

- Display the contents of the file with underlines where applicable:

```
ul {{file.txt}}
```

- Display the contents of the file with underlines made of dashes `-`:

```
ul {{[-i|--indicated]}} {{file.txt}}
```

umount

Unlink a filesystem from its mount point, making it no longer accessible.

A filesystem cannot be unmounted when it is busy.

More information: https://manned.org/umount.8.

- Unmount a filesystem, by passing the path to the source it is mounted from:

```
sudo umount {{path/to/device_file}}
```

- Unmount a filesystem, by passing the path to the target where it is mounted:

```
sudo umount {{path/to/mounted_directory}}
```

- When an unmount fails, try to remount the filesystem read-only:

```
sudo umount {{[-r|--read-only]}} {{path/to/
mounted_directory}}
```

- Recursively unmount each specified directory:

```
sudo umount {{[-R|--recursive]}} {{path/to/
mounted_directory}}
```

- Unmount all mounted filesystems (except the `proc` filesystem):

```
sudo umount {{[-a|--all]}}
```

uname26

This command is an alias of **`setarch uname26`**.

- View documentation for the original command:

  ```
  tldr setarch
  ```

uncompress

Uncompress files compressed using the Unix **`compress`** command.

More information: https://manned.org/uncompress.1.

- Uncompress specific files:

```
uncompress {{path/to/file1.Z path/to/file2.Z ...}}
```

- Uncompress specific files while ignoring non-existent ones:

```
uncompress -f {{path/to/file1.Z path/to/file2.Z ...}}
```

- Write to `stdout` (no files are changed and no `.Z` files are created):

```
uncompress -c {{path/to/file1.Z path/to/file2.Z ...}}
```

- Verbose mode (write to `stderr` about percentage reduction or expansion):

```
uncompress -v {{path/to/file1.Z path/to/file2.Z ...}}
```

unix2dos

Change Unix-style line endings to DOS-style.

Replaces LF with CRLF.

See also: **`unix2mac`**, **`dos2unix`**, **`mac2unix`**.

More information: https://manned.org/unix2dos.

- Change the line endings of a file:

```
unix2dos {{path/to/file}}
```

- Create a copy with DOS-style line endings:

```
unix2dos {{[-n|--newfile]}} {{path/to/file}} {{path/to/new_file}}
```

- Display file information:

```
unix2dos {{[-i|--info]}} {{path/to/file}}
```

- Keep/add/remove Byte Order Mark:

```
unix2dos --{{keep-bom|add-bom|remove-bom}} {{path/to/file}}
```

unix2mac

Change Unix-style line endings to macOS-style.

Replaces LF with CR.

See also: **`unix2dos`**, **`dos2unix`**, **`mac2unix`**.

More information: https://manned.org/unix2mac.

- Change the line endings of a file:

```
unix2mac {{path/to/file}}
```

- Create a copy with macOS-style line endings:

```
unix2mac {{[-n|--newfile]}} {{path/to/file}} {{path/to/new_file}}
```

- Display file information:

```
unix2mac {{[-i|--info]}} {{path/to/file}}
```

- Keep/add/remove Byte Order Mark:

```
unix2mac --{{keep-bom|add-bom|remove-bom}} {{path/to/file}}
```

unmount

The correct command is **`umount`** (u-mount).

More information: https://manned.org/umount.8.

- View documentation for the correct command:

```
tldr umount
```

unopkg

LibreOffice extensions manager.

Download extensions from https://extensions.libreoffice.org.

See also: **`libreoffice`**.

More information: https://manned.org/unopkg.

- Add and deploy given extension:

```
unopkg add {{path/to/extension}}
```

- Remove extension:

```
unopkg remove {{extensions_id}}
```

- Display information about deployed extensions:

```
unopkg list
```

- Raise extensions dialog (GUI):

```
unopkg gui
```

- Reinstall all deployed extensions:

```
unopkg reinstall
```

- Display help:

```
unopkg {{[-h|--help]}}
```

unshadow

Utility provided by the John the Ripper project to obtain the traditional Unix password file if the system uses shadow passwords.

More information: https://www.openwall.com/john/doc/.

- Combine the `/etc/shadow` and `/etc/passwd` of the current system:

```
sudo unshadow /etc/passwd /etc/shadow
```

- Combine two arbitrary shadow and password files:

```
sudo unshadow {{path/to/passwd}} {{path/to/shadow}}
```

unshare

Execute a command in new user-defined namespaces.

More information: https://manned.org/unshare.

- Execute a command without sharing access to connected networks:

```
unshare {{[-n|--net]}} {{command}} {{command_arguments}}
```

- Execute a command as a child process without sharing mounts, processes, or networks:

```
unshare {{[-m|--mount]}} {{[-i|--pid]}} {{[-n|--net]}} {{[-f|--fork]}} {{command}} {{command_arguments}}
```

unsquashfs

Uncompress, extract and list files in squashfs filesystems.

More information: https://manned.org/unsquashfs.

- Extract a squashfs filesystem to `squashfs-root` in the current working directory:

```
unsquashfs {{filesystem.squashfs}}
```

- Extract a squashfs filesystem to the specified directory:

```
unsquashfs {{[-d|-dest]}} {{path/to/directory}} {{filesystem.squashfs}}
```

- Display the names of files as they are extracted:

```
unsquashfs {{[-i|-info]}} {{filesystem.squashfs}}
```

- Display the names of files and their attributes as they are extracted:

```
unsquashfs {{[-li|-linfo]}} {{filesystem.squashfs}}
```

- List files inside the squashfs filesystem (without extracting):

```
unsquashfs {{[-l|-ls]}} {{filesystem.squashfs}}
```

- List files and their attributes inside the squashfs filesystem (without extracting):

```
unsquashfs {{[-ll|-lls]}} {{filesystem.squashfs}}
```

unzipsfx

Create a self-extracting compressed binary file by prepending self-extracting stubs on a Zip file.

More information: https://manned.org/unzipsfx.

- Create a self-extracting binary file of a Zip archive:

```
cat unzipsfx {{path/to/archive.zip}} > {{filename}} && chmod 755 {{filename}}
```

- Extract a self-extracting binary in the current directory:

```
{{./path/to/binary}}
```

- Test a self-extracting binary for errors:

```
{{./path/to/binary}} -t
```

- Print content of a file in the self-extracting binary without extraction:

```
{{./path/to/binary}} -c {{path/to/file}}
```

- Print comments on Zip archive in the self-extracting binary:

```
{{./path/to/binary}} -z
```

update-alternatives

Conveniently maintain symbolic links to determine default commands.

More information: https://manned.org/update-alternatives.

- Add a symbolic link:

```
sudo update-alternatives --install {{path/to/symlink}} {{command_name}} {{path/to/command_binary}} {{priority}}
```

- Configure a symbolic link for `java`:

```
sudo update-alternatives --config {{java}}
```

- Remove a symbolic link:

```
sudo update-alternatives --remove {{java}} {{/opt/java/jdk1.8.0_102/bin/java}}
```

- Display information about a specified command:

```
update-alternatives --display {{java}}
```

- Display all commands and their current selection:

```
update-alternatives --get-selections
```

update-initramfs

Manage initramfs.

More information: https://manned.org/update-initramfs.

- Create a new initramfs (use `all` for all installed kernel versions):

```
sudo update-initramfs -c -k {{kernel_version}}
```

- Update an existing initramfs:

```
sudo update-initramfs -u
```

- Remove an existing initramfs (be careful when using `all` for `kernel_version`):

```
sudo update-initramfs -d -k {{kernel_version}}
```

update-rc.d

Install and remove services which are System-V style init script links.

Init scripts are in the **`/etc/init.d/`**.

More information: https://manned.org/update-rc.d.

- Install a service:

```
update-rc.d {{mysql}} defaults
```

- Enable a service:

```
update-rc.d {{mysql}} enable
```

- Disable a service:

```
update-rc.d {{mysql}} disable
```

- Forcibly remove a service:

```
update-rc.d -f {{mysql}} remove
```

updatectl

System update utility.

More information: https://www.freedesktop.org/software/systemd/man/updatectl.html.

- Check to see if the system has any available updates:

```
updatectl check
```

- Update to the latest version:

```
updatectl update
```

- Show update targets:

```
updatectl list
```

- Show data about a target and its versions:

```
updatectl list {{target}}
```

- Return local data about a target without fetching from the network:

```
updatectl --offline list {{target}}
```

- Apply an update to a target and then reboot the system:

```
updatectl --reboot update {{target}}
```

- Clean up old versions of a specified target:

```
updatectl vacuum {{target}}
```

- Display help:

```
updatectl {{[-h|--help]}}
```

updatedb

Create or update the database used by **`locate`**.

It is usually run daily by cron.

More information: https://manned.org/updatedb.

- Refresh database content:

```
sudo updatedb
```

- Display file names as soon as they are found:

```
sudo updatedb {{[-v|--verbose]}}
```

updpkgsums

Update the checksums of the sources in a **PKGBUILD**.

Unless a preexisting hashing algorithm is used, SHA256 will be used.

More information: https://manned.org/updpkgsums.

- Update the checksums in a PKGBUILD:

```
updpkgsums
```

- Display help:

```
updpkgsums {{[-h|--help]}}
```

- Display version:

```
updpkgsums {{[-v|--version]}}
```

upnpc

Configure port forwarding rules on your router via the UPnP protocol.

More information: https://manned.org/upnpc.

- Forward the external TCP port 80 to port 8080 on a local machine:

```
upnpc -a {{192.168.0.1}} 8080 80 tcp
```

- Delete any port redirection for external TCP port 80:

```
upnpc -d 80 tcp
```

- Get information about UPnP devices on your network:

```
upnpc -s
```

- List existing redirections:

```
upnpc -l
```

upower

System utility to provide power and battery information and statistics.

More information: https://upower.freedesktop.org/docs/upower.1.html.

- Display power and battery information:

```
upower {{[-d|--dump]}}
```

- List all power devices:

```
upower {{[-e|--enumerate]}}
```

- Watch for and print power status changes:

```
upower {{[-m|--monitor]}}
```

- Watch for and print detailed power status changes:

```
upower --monitor-detail
```

- Display version:

```
upower {{[-v|--version]}}
```

uprecords

Display a summary of historical uptime records.

More information: https://manned.org/uprecords.

- Display a summary of the top 10 historical uptime records:

```
uprecords
```

- Display the top 25 records:

```
uprecords -m {{25}}
```

- Display the downtime between reboots instead of the kernel version:

```
uprecords -d
```

- Show the most recent reboots:

```
uprecords -B
```

- Don't truncate information:

```
uprecords -w
```

urpme

Uninstall packages in Mageia.

See also: **`urpmi`**, **`urpmi.update`**, **`urpmi.addmedia`**, **`urpmi.removemedia`**, **`urpmf`**, **`urpmq`**.

More information: https://man.linuxreviews.org/man8/urpme.8.html.

- Uninstall a package:

```
sudo urpme {{package}}
```

- Uninstall orphan packages (Note: Use it with caution as it might unintentionally remove important packages):

```
sudo urpme --auto-orphans
```

- Uninstall a package and its dependencies:

```
sudo urpme --auto-orphans {{package}}
```

urpmf

Find files in packages and query information about them in Mageia.

See also: **urpmi**, **urpme**, **urpmi.addmedia**, **urpmi.removemedia**, **urpmi.update**, **urpmq**.

More information: https://man.linuxreviews.org/man8/urpmf.8.html.

- Search for packages that contain a file:

```
urpmf {{filename}}
```

- Search for packages that contain both a keyword [a]nd another in their summaries:

```
urpmf --summary {{keyword1}} -a {{keyword2}}
```

- Search for packages that contain a keyword [o]r another in their descriptions:

```
urpmf --description {{keyword1}} -o {{keyword2}}
```

- Search for packages that do not contain a keyword in their name ignoring case distinction using "|" as the [F]ield separator (":" by default):

```
urpmf --description ! {{keyword}} -F'|'
```

urpmi.addmedia

Add media in Mageia.

Note: Mageia documentation uses medium and repository as synonymous.

See also: **`urpmi`**, **`urpmi.update`**, **`urpme`**, **`urpmi.removemedia`**, **`urpmf`**, **`urpmq`**.

More information: https://man.linuxreviews.org/man8/urpmi.addmedia.8.html.

- Add a medium:

```
sudo urpmi.addmedia {{medium}} {{ftp://ftp.site.com/path/to/Mageia/RPMS}}
```

- Add a medium from a hard drive (run `genhdlist2` in the directory first):

```
sudo urpmi.addmedia --distrib HD file://{{path/to/repo}}
```

- Add important media from a chosen mirror:

```
sudo urpmi.addmedia --distrib ftp://{{mirror_website}}/mirror/mageia/distrib/{{version}}/{{arch}}
```

- Automatically select mirrors from a mirror list:

```
sudo urpmi.addmedia --distrib --mirrorlist {{mirrorlist}}
```

urpmi

Install packages in Mageia.

See also: **`urpm.update`**, **`urpme`**, **`urpmi.addmedia`**, **`urpmi.removemedia`**, **`urpmf`**, **`urpmq`**.

More information: https://man.linuxreviews.org/man8/urpmi.8.html.

- Install a package from the repository or from a local RPM file:

```
sudo urpmi {{package|path/to/file.rpm}}
```

- Download a package without installing it:

```
urpmi --no-install {{package}}
```

- Update all installed packages (run `urpmi.update -a` to get the available updates):

```
sudo urpmi --auto-select
```

- Update a package of one or more machines on the network according to `/etc/urpmi/parallel.cfg`:

```
sudo urpmi --parallel local {{package}}
```

- Mark all orphaned packages as manually installed:

```
sudo urpmi $(urpmq --auto-orphans -f)
```

urpmi.removemedia

Remove media in Mageia.

Note: Mageia documentation uses medium and repository as synonymous.

See also: **`urpmi`**, **`urpme`**, **`urpmi.addmedia`**, **`urpmi.update`**, **`urpmf`**, **`urpmq`**.

More information: https://man.linuxreviews.org/man8/urpmi.removemedia.8.html.

- Remove a medium:

```
sudo urpmi.removemedia {{medium}}
```

- Remove [a]ll media:

```
sudo urpmi.removemedia -a
```

- Remove media fuzz[y] matching on media names:

```
sudo urpmi.removemedia -y {{keyword}}
```

urpmi.update

Update the list of packages from a package repository in Mageia.

Note: Mageia documentation uses medium and repository as synonymous.

See also: **`urpmi`**, **`urpme`**, **`urpmi.addmedia`**, **`urpmi.removemedia`**, **`urpmf`**, **`urpmq`**.

More information: https://man.linuxreviews.org/man8/urpmi.update.8.html.

- Update all enabled media:

```
urpmi.update -a
```

- Update specific media (including disabled media):

```
urpmi.update {{medium1 medium2 ...}}
```

- Update all media that contain a specific keyword:

```
urpmi.update {{keyword}}
```

- Update all configured media:

```
urpmi.update e
```

urpmq

Query information about packages and media in Mageia.

See also: **`urpmi`**, **`urpmi.update`**, **`urpmi.addmedia`**, **`urpmi.removemedia`**, **`urpmf`**, **`urpme`**.

More information: https://man.linuxreviews.org/man8/urpmq.8.html.

- Display information about an installable package:

```
urpmq -i {{package}}
```

- Display direct dependencies of a package:

```
urpmq --requires {{package}}
```

- Display direct and indirect dependencies of a package:

```
urpmq {{[-d|--requires-recursive]}} {{package}}
```

- List the not installed packages needed for an RPM file with their sources:

```
sudo urpmq {{[-d|--requires-recursive]}} -m --sources {{path/to/file.rpm}}
```

- List all configured media with their URLs, including inactive media:

```
urpmq --list-media --list-url
```

- Search for a package printing [g]roup, version and [r]elease:

```
urpmq -g -r {{[-y|--fuzzy]}} {{keyword}}
```

- Search for a package with using its exact name:

```
urpmq -g -r {{package}}
```

urxvt

Rxvt-unicode.

A customizable terminal emulator.

More information: https://manned.org/urxvt.

- Open a new urxvt window:

```
urxvt
```

- Run in a specific directory:

```
urxvt -cd {{path/to/directory}}
```

- Run a command in a new urxvt window:

```
urxvt -e {{command}}
```

- Run a command and keep the window open:

```
urxvt --hold -e {{command}}
```

- Run a command within the `sh` shell:

```
urxvt -e {{sh}} -c {{command}}
```

usbip

Use USB devices remotely.

More information: https://usbip.sourceforge.net.

- List all local USB devices and their bus ID's:

```
usbip list {{[-l|--local]}}
```

- Start a `usbip` daemon on the server:

```
systemctl start usbipd
```

- Bind a USB device to `usbip` on the server:

```
sudo usbip bind {{[-b|--busid]}} {{bus_id}}
```

- Load the kernel module required by `usbip` on the client:

```
sudo modprobe vhci-hcd
```

- Attach to the `usbip` device on the client (bus ID is the same as on the server):

```
sudo usbip attach {{[-r|--remote]}} {{ip_address}} {{[-b|--busid]}} {{bus_id}}
```

- List attached devices:

```
usbip port
```

- Detach from a device:

```
sudo usbip detach {{[-p|--port]}} {{port}}
```

- Unbind a device:

```
usbip unbind {{[-b|--busid]}} {{bus_id}}
```

useradd

Create a new user.

See also: **`users`**, **`userdel`**, **`usermod`**.

More information: https://manned.org/useradd.

- Create a new user:

```
sudo useradd {{username}}
```

- Create a new user with the specified user ID:

```
sudo useradd {{[-u|--uid]}} {{id}} {{username}}
```

- Create a new user with the specified shell:

```
sudo useradd {{[-s|--shell]}} {{path/to/shell}} {{username}}
```

- Create a new user belonging to additional groups (mind the lack of whitespace):

```
sudo useradd {{[-G|--groups]}} {{group1,group2,...}} {{username}}
```

- Create a new user with the default home directory:

```
sudo useradd {{[-m|--create-home]}} {{username}}
```

- Create a new user with the home directory filled by template directory files:

```
sudo useradd {{[-k|--skel]}} {{path/to/template_directory}} {{[-m|--create-home]}} {{username}}
```

- Create a new system user without the home directory:

```
sudo useradd {{[-r|--system]}} {{username}}
```

userdbctl

Inspect users, groups and group memberships on the system.

More information: https://www.freedesktop.org/software/systemd/man/userdbctl.html.

- List all known user records:

```
userdbctl user
```

- Show details of a specific user:

```
userdbctl user {{username}}
```

- List all known groups:

```
userdbctl group
```

- Show details of a specific group:

```
userdbctl group {{groupname}}
```

- List all services currently providing user/group definitions to the system:

```
userdbctl services
```

userdel

Remove a user account or remove a user from a group.

See also: **`users`**, **`useradd`**, **`usermod`**.

More information: https://manned.org/userdel.

- Remove a user:

```
sudo userdel {{username}}
```

- Remove a user in other root directory:

```
sudo userdel {{[-R|--root]}} {{path/to/other_root}}
{{username}}
```

- Remove a user along with the home directory and mail spool:

```
sudo userdel {{[-r|--remove]}} {{username}}
```

usermod

Modify a user account.

See also: **`users`**, **`useradd`**, **`userdel`**.

More information: https://manned.org/usermod.

- Change a username:

```
sudo usermod {{[-l|--login]}} {{new_username}} {{username}}
```

- Change a user ID:

```
sudo usermod {{[-u|--uid]}} {{id}} {{username}}
```

- Change a user shell:

```
sudo usermod {{[-s|--shell]}} {{path/to/shell}} {{username}}
```

- Add a user to supplementary groups (mind the lack of whitespace):

```
sudo usermod {{[-aG|--append --groups]}}
{{group1,group2,...}} {{username}}
```

- Remove a user from specific groups:

```
sudo usermod {{[-rG|--remove --groups]}}
{{group1,group2,...}} {{username}}
```

- Change a user home directory:

```
sudo usermod {{[-m|--move-home]}} {{[-d|--home]}} {{path/to/
new_home}} {{username}}
```

- Lock an account:

```
sudo usermod {{[-L|--lock]}} {{username}}
```

- Unlock an account:

```
sudo usermod {{[-U|--unlock]}} {{username}}
```

utmpdump

Dump and load btmp, utmp and wtmp accounting files.

More information: https://manned.org/utmpdump.

- Dump the `/var/log/wtmp` file to `stdout` as plain text:

  `utmpdump {{/var/log/wtmp}}`

- Load a previously dumped file into `/var/log/wtmp`:

  `utmpdump {{[-r|--reverse]}} {{dumpfile}} > {{/var/log/wtmp}}`

uuid

Generate and decode Universally Unique Identifiers (UUID).

See also: **`uuidgen`**.

More information: https://manned.org/uuid.

- Generate a UUIDv1 (based on time and system's hardware address, if present):

```
uuid
```

- Generate a UUIDv4 (based on random data):

```
uuid -v {{4}}
```

- Generate multiple UUIDv4 identifiers at once:

```
uuid -v {{4}} -n {{number_of_uuids}}
```

- Generate a UUIDv4 and specify the output format:

```
uuid -v {{4}} -F {{BIN|STR|SIV}}
```

- Generate a UUIDv4 and write the output to a file:

```
uuid -v {{4}} -o {{path/to/file}}
```

- Generate a UUIDv5 (based on the supplied object name) with a specified namespace prefix:

```
uuid -v {{5}} ns:{{DNS|URL|OID|X500}} {{object_name}}
```

- Decode a given UUID:

```
uuid -d {{uuid}}
```

uuidd

Daemon for generating UUIDs.

More information: https://manned.org/uuidd.

- Generate a random UUID:

```
uuidd {{[-r|--random]}}
```

- Generate a bulk number of random UUIDs:

```
uuidd {{[-r|--random]}} {{[-n|--uuids]}} {{number_of_uuids}}
```

- Generate a time-based UUID, based on the current time and MAC address of the system:

```
uuidd {{[-t|--time]}}
```

uuidgen

Generate unique identifiers (UUIDs).

See also: **`uuid`**.

More information: https://manned.org/uuidgen.

- Create a random UUIDv4:

```
uuidgen {{[-r|--random]}}
```

- Create a UUIDv1 based on the current time:

```
uuidgen {{[-t|--time]}}
```

- Create a UUIDv5 of the name with a specified namespace prefix:

```
uuidgen {{[-s|--sha1]}} {{[-n|--namespace]}} {{@dns|@url|
@oid|@x500}} {{[-N|--name]}} {{object_name}}
```

uuidparse

Parse universally unique identifiers.

See also: **`uuidgen`**.

More information: https://manned.org/uuidparse.1.

- Parse the specified UUIDs, use a tabular output format:

```
uuidparse {{uuid1 uuid2 ...}}
```

- Parse UUIDs from `stdin`:

```
{{command}} | uuidparse
```

- Use the JSON output format:

```
uuidparse {{[-J|--json]}} {{uuid1 uuid2 ...}}
```

- Do not print a header line:

```
uuidparse {{[-n|--noheadings]}} {{uuid1 uuid2 ...}}
```

- Use the raw output format:

```
uuidparse {{[-r|--raw]}} {{uuid1 uuid2 ...}}
```

- Specify which of the four output columns to print:

```
uuidparse {{[-o|--output]}} {{UUID,VARIANT,TYPE,TIME}}
```

- Display help:

```
uuidparse {{[-h|--help]}}
```

uvcdynctrl

Manage dynamic controls in uvcvideo.

More information: https://manned.org/uvcdynctrl.

- List all available cameras:

```
uvcdynctrl {{[-l|--list]}}
```

- Use a specific device (defaults to `video0`):

```
uvcdynctrl {{[-d|--device]}} {{device_name}}
```

- List available controls:

```
uvcdynctrl {{[-c|--clist]}}
```

- Set a new control value (for negative values, use `-- -value`):

```
uvcdynctrl {{[-s|--set]}} {{control_name}} {{value}}
```

- Get the current control value:

```
uvcdynctrl {{[-g|--get]}} {{control_name}}
```

- Save the state of the current controls to a file:

```
uvcdynctrl {{[-W|--save]}} {{filename}}
```

- Load the state of the controls from a file:

```
uvcdynctrl {{[-L|--load]}} {{filename}}
```

v4l2-ctl

Control video devices.

More information: https://manned.org/v4l2-ctl.

- List all video devices:

```
v4l2-ctl {{[-A|--list-devices]}}
```

- List supported video formats and resolutions of default video device `/dev/video0`:

```
v4l2-ctl --list-formats-ext
```

- List supported video formats and resolutions of a specific video device:

```
v4l2-ctl --list-formats-ext {{[-d|--device]}} {{path/to/video_device}}
```

- Get all details of a video device:

```
v4l2-ctl --all {{[-d|--device]}} {{path/to/video_device}}
```

- Capture a JPEG photo with a specific resolution from video device:

```
v4l2-ctl {{[-d|--device]}} {{path/to/video_device}} --set-fmt-video=width={{width}},height={{height}},pixelformat=MJPG --stream-mmap --stream-to={{path/to/output.jpg}} --stream-count=1
```

- Capture a raw video stream from video device:

```
v4l2-ctl {{[-d|--device]}} {{path/to/video_device}} --set-fmt-video=width={{width}},height={{height}},pixelformat={{format}} --stream-mmap --stream-to={{path/to/output}} --stream-count={{number_of_frames_to_capture}}
```

- List all video device's controls and their values:

```
v4l2-ctl {{[-l|--list-ctrls]}} {{[-d|--device]}} {{path/to/video_device}}
```

- Set the value of a video device control:

```
v4l2-ctl {{[-d|--device]}} {{path/to/video_device}} {{[-c|--set-ctrl]}} {{control_name}}={{value}}
```

Valheim

Create and start a headless Valheim server.

Note: Configure the server by editing a copy of the **`start_server.sh`** script.

More information: https://www.valheimgame.com/support/a-guide-to-dedicated-servers/.

- Start the server:

  ```
  {{path/to/start_server.sh}}
  ```

- Stop the server gracefully:

  ```
  <Ctrl c>
  ```

- Start the server using the provided Docker script:

  ```
  {{path/to/docker_start_server.sh}} {{path/to/
  start_server.sh}}
  ```

vcgencmd

Print system information for a Raspberry Pi.

More information: https://www.raspberrypi.com/documentation/computers/os.html#vcgencmd.

- List all available commands:

```
vcgencmd commands
```

- Print the current CPU temperature:

```
vcgencmd measure_temp
```

- Print the current voltage:

```
vcgencmd measure_volts
```

- Print the throttled state of the system as a bit pattern:

```
vcgencmd get_throttled
```

- Print the bootloader configuration (only available on Raspberry Pi 4 models):

```
vcgencmd bootloader_config
```

- Display help:

```
vcgencmd --help
```

ventoy

A tool to create bootable USB drives using ISO files.

More information: https://www.ventoy.net/en/doc_start.html#doc_linux_cli.

- Install Ventoy to a specific drive with the defaults:

```
sudo ventoy -i {{/dev/sdX}}
```

- Install Ventoy with GPT partition style instead of MBR:

```
sudo ventoy -i -g {{/dev/sdX}}
```

- Install Ventoy with GPT partition style and a custom partition label and secure boot disabled:

```
sudo ventoy -i -g -S -L {{LABEL_NAME}} {{/dev/sdX}}
```

- Install Ventoy and reserve space at the end of the disk:

```
sudo ventoy -i -r {{SIZE_MB}} {{/dev/sdX}}
```

- Force install Ventoy (overwrites existing installation):

```
sudo ventoy -I {{/dev/sdX}}
```

- Update Ventoy on a drive:

```
sudo ventoy -u {{/dev/sdX}}
```

- Display Ventoy information for a drive:

```
sudo ventoy -l {{/dev/sdX}}
```

- Try non-destructive installation if possible (Ventoy will not reformat the disk):

```
sudo ventoy -i -n {{/dev/sdX}}
```

veracrypt

Free and open source disk encryption software.

More information: https://arcanecode.com/2021/06/21/veracrypt-on-the-command-line-for-ubuntu-linux/.

- Create a new volume through a text user interface and use `/dev/urandom` as a source of random data:

```
veracrypt {{[-t|--text]}} {{[-c|--create]}} --random-source={{/dev/urandom}}
```

- Decrypt a volume interactively through a text user interface and mount it to a directory:

```
veracrypt {{[-t|--text]}} {{path/to/volume}} {{path/to/mount_point}}
```

- Decrypt a partition using a keyfile and mount it to a directory:

```
veracrypt {{[-k|--keyfiles]}} {{path/to/keyfile}} {{/dev/sdXN}} {{path/to/mount_point}}
```

- Dismount a volume on the directory it is mounted to:

```
veracrypt {{[-d|--dismount]}} {{path/to/mounted_point}}
```

vgcfgbackup

Back up volume group configuration metadata to files (not user data).

More information: https://manned.org/vgcfgbackup.

- Back up metadata for all volume groups (to `/etc/lvm/backup/` by default):

```
vgcfgbackup
```

- Back up metadata for a specific volume group:

```
vgcfgbackup {{vg_name}}
```

- Write the backup to a specific file:

```
vgcfgbackup {{[-f|--file]}} {{path/to/backup}} {{vg_name}}
```

- Back up multiple VGs using a filename template (`%s` becomes the VG name):

```
vgcfgbackup {{[-f|--file]}} {{/tmp/vg-backup-%s}} {{vg1
vg2 ...}}
```

- Increase verbosity (repeat `-v` for more detail):

```
vgcfgbackup {{[-v|--verbose]}} {{vg_name}}
```

vgcfgrestore

Restore volume group configuration (not user data) from a text back up file produced by **`vgcfgbackup`**.

More information: https://manned.org/vgcfgrestore.

- Restore VG metadata from last backup.:

```
vgcfgrestore {{vg_name}}
```

- Restore VG metadata from specified backup-file:

```
vgcfgrestore {{[-f|--file]}} {{path/to/file}} {{vg_name}}
```

- List all VG metadata backups:

```
vgcfgrestore {{[-l|--list]}} {{vg_name}}
```

- List one VG metadata backup file:

```
vgcfgrestore {{[-l|--list]}} {{[-f|--file]}} {{path/to/file}}
{{vg_name}}
```

vgchange

Change the attributes of a Logical Volume Manager (LVM) volume group.

See also: **`lvm`**.

More information: https://manned.org/vgchange.

- Change the activation status of logical volumes in all volume groups:

```
sudo vgchange {{[-a|--activate]}} {{y|n}}
```

- Change the activation status of logical volumes in the specified volume group (determine with `vgscan`):

```
sudo vgchange {{[-a|--activate]}} {{y|n}} {{volume_group}}
```

vgcreate

Create volume groups combining multiple mass-storage devices.

See also: **`lvm`**.

More information: https://manned.org/vgcreate.

- Create a new volume group called vg1 using the `/dev/sda1` device:

  ```
  vgcreate {{vg1}} {{/dev/sda1}}
  ```

- Create a new volume group called vg1 using multiple devices:

  ```
  vgcreate {{vg1}} {{/dev/sda1}} {{/dev/sdb1}} {{/dev/sdc1}}
  ```

vgdisplay

Display information about Logical Volume Manager (LVM) volume groups.

See also: **`lvm`**.

More information: https://manned.org/vgdisplay.

- Display information about all volume groups:

```
sudo vgdisplay
```

- Display information about volume group vg1:

```
sudo vgdisplay {{vg1}}
```

vgextend

Add one or more physical volumes to an existing volume group.

More information: https://manned.org/vgextend.

- Add a physical volume to an existing volume group:

```
vgextend {{vg1}} {{/dev/sda1}}
```

- Add multiple physical volumes to an existing volume group:

```
vgextend {{vg1}} {{/dev/sda1 /dev/sda2 ...}}
```

vgremove

Remove volume group(s) in LVM.

More information: https://manned.org/vgremove.

- Remove a volume group with confirmation:

```
vgremove {{volume_group}}
```

- Forcefully remove a volume group without confirmation:

```
vgremove {{[-f|--force]}} {{volume_group}}
```

- Set the debug level for detailed logging to level 2, (repeat `--debug` up to 6 times to increase the level):

```
vgremove {{[-d|--debug]}} {{[-d|--debug]}} {{volume_group}}
```

- Use a specific config setting to override defaults:

```
vgremove --config '{{global/
locking_type=1}}' {{volume_group}}
```

- Display help:

```
vgremove {{[-h|--help]}}
```

vgs

Display information about volume groups.

See also: **`lvm`**.

More information: https://manned.org/vgs.

- Display information about volume groups:

```
sudo vgs
```

- Display all volume groups:

```
sudo vgs {{[-a|--all]}}
```

- Change default display to show more details:

```
sudo vgs {{[-v|--verbose]}}
```

- Display only specific fields:

```
sudo vgs {{[-o|--options]}} {{field_name_1}},{{field_name_2}}
```

- Append field to default display:

```
sudo vgs {{[-o|--options]}} +{{field_name}}
```

- Suppress heading line:

```
sudo vgs --noheadings
```

- Use separator to separate fields:

```
sudo vgs --separator =
```

vgscan

Scan for volume groups on all supported Logical Volume Manager (LVM) block devices.

See also: **`lvm`**, **`vgchange`**.

More information: https://manned.org/vgscan.

- Scan for volume groups and print information about each group found:

  `sudo vgscan`

- Scan for volume groups and add the special files in `/dev`, if they don't already exist, needed to access the logical volumes in the found groups:

  `sudo vgscan --mknodes`

viewnior

Simple and elegant image viewer.

More information: https://manned.org/viewnior.

- View an image:

```
viewnior {{path/to/image.ext}}
```

- View in fullscreen mode:

```
viewnior --fullscreen {{path/to/image.ext}}
```

- View fullscreen in slideshow mode:

```
viewnior --slideshow {{path/to/image.ext}}
```

vigr

Edit the group file.

More information: https://manned.org/vigr.

- Edit the group file:

```
vigr
```

- Display version:

```
vigr {{[-V|--version]}}
```

vipw

Edit the password file.

More information: https://manned.org/vipw.

- Edit the password file:

```
vipw
```

- Display version:

```
vipw {{[-V|--version]}}
```

virt-manager

A desktop user interface for managing KVM and Xen virtual machines and LXC containers.

More information: https://manned.org/virt-manager.1.

- Launch the GUI:

```
virt-manager
```

- Connect to a hypervisor:

```
virt-manager --connect {{hypervisor_uri}}
```

- Don't fork virt-manager process into background on startup:

```
virt-manager --no-fork
```

- Print debug output:

```
virt-manager --debug
```

- Open the "New VM" wizard:

```
virt-manager --show-domain-creator
```

- Show domain details window for a specific virtual machine/container:

```
virt-manager --show-domain-editor {{name|id|uuid}}
```

- Show domain performance window for a specific virtual machine/container:

```
virt-manager --show-domain-performance {{name|id|uuid}}
```

- Show connection details window:

```
virt-manager --show-host-summary
```

virt-viewer

Minimal graphical interface for a virtual machine (VM).

Note: **`domain`** refers to the name, UUID or ID for the existing VMs.

See also: **`virsh`**.

More information: https://manned.org/virt-viewer.

- Launch `virt-viewer` with a prompt to select running virtual machines:

```
virt-viewer
```

- Launch `virt-viewer` for a specific virtual machine by ID, UUID or name:

```
virt-viewer "{{domain}}"
```

- Wait for a virtual machine to start and automatically reconnect if it shutdown and restarts:

```
virt-viewer --reconnect --wait "{{domain}}"
```

- Connect to a specific remote virtual machine over TLS:

```
virt-viewer --connect "xen//{{url}}" "{{domain}}"
```

- Connect to a specific remote virtual machine over SSH:

```
virt-viewer --connect "qemu+ssh//{{username}}@{{url}}/system"
"{{domain}}"
```

virt-what

Shell script that detects if you are running in a virtual machine.

Outputs a specific code for a virtual machine if detected.

More information: https://manned.org/virt-what.

- Detect if you are running in a virtual machine:

```
virt-what
```

- Log output of `virt-what` into a file:

```
virt-what > {{path/to/file}}
```

- Display version:

```
virt-what --version
```

virt-xml-validate

Validate **`libvirt`** XML files against a schema.

If a schema is not specified, the schema is determined by the root element in the XML file.

More information: https://libvirt.org/manpages/virt-xml-validate.html.

- Validate an XML file against a specific schema:

```
virt-xml-validate {{path/to/file.xml}} {{schema}}
```

- Validate the domain XML against the domain schema:

```
virt-xml-validate {{path/to/domain.xml}} domain
```

virt-xml

Edit libvirt Domain XML files with explicit command-line options.

Note: **`domain`** refers to the name, UUID or ID for the existing VMs.

See also: **`virsh`**.

More information: https://github.com/virt-manager/virt-manager/blob/main/man/virt-xml.rst.

- List all the suboptions for a specific option:

```
virt-xml --{{option}}=?
```

- List all the suboptions for disk, network, and boot:

```
virt-xml --disk=? --network=? --boot=?
```

- Edit a value for a specific domain:

```
virt-xml {{domain}} --edit --{{option}} {{suboption}}
={{new_value}}
```

- Change the description for a specific domain:

```
virt-xml {{domain}} --edit --metadata
description="{{new_description}}"
```

- Enable/Disable the boot device menu for a specific domain:

```
virt-xml {{domain}} --edit --boot bootmenu={{on|off}}
```

- Attach host USB hub to a running VM (run `lsusb` to see a list of USB devices with their IDs):

```
virt-xml {{domain}} --update --add-device --hostdev {{bus}}.
{{device}}
```

vivaldi-stable

This command is an alias of **`chromium`**.

More information: https://vivaldi.com.

- View documentation for the original command:

```
tldr chromium
```

vkpurge

List or remove old kernel versions left behind by **`xbps`**.

The **`version`** arguments support shell globs.

More information: https://man.voidlinux.org/vkpurge.8.

- List all removable kernel versions (or those matching `version` if the argument is specified):

```
vkpurge list {{version}}
```

- Remove all unused kernels:

```
vkpurge rm all
```

- Remove kernel versions matching `version`:

```
vkpurge rm {{version}}
```

vlock

Lock virtual console.

More information: https://manned.org/vlock.

- Lock current virtual console session:

```
vlock
```

- Lock current virtual console session via flag:

```
vlock {{[-c|--current]}}
```

- Lock all virtual console sessions:

```
vlock {{[-a|--all]}}
```

- Display help:

```
vlock {{[-h|--help]}}
```

- Display version:

```
vlock {{[-v|--version]}}
```

vmstat

Report information about processes, memory, paging, block IO, traps, disks and CPU activity.

More information: https://manned.org/vmstat.

- Display virtual memory statistics:

```
vmstat
```

- Display reports every 2 seconds for 5 times:

```
vmstat {{2}} {{5}}
```

vmware-checkvm

Check if the current host is a VMware VM or not.

More information: https://manned.org/vmware-checkvm.

- Return the current VMware software version (exit status determines whether the system is a VM or not):

```
vmware-checkvm
```

- Return the VMware hardware version:

```
vmware-checkvm -h
```

vncserver

Launch a VNC (Virtual Network Computing) desktop.

More information: https://manned.org/vncserver.1x.

- Launch a VNC Server on next available display:

```
vncserver
```

- Launch a VNC Server with specific screen geometry:

```
vncserver --geometry {{width}}x{{height}}
```

- Kill an instance of VNC Server running on a specific display:

```
vncserver --kill :{{display_number}}
```

vncviewer

Launch a VNC (Virtual Network Computing) client.

More information: https://manned.org/vncviewer.

- Launch a VNC client which connects to a host on a given display:

```
vncviewer {{host}}:{{display_number}}
```

- Launch in full-screen mode:

```
vncviewer -FullScreen {{host}}:{{display_number}}
```

- Launch a VNC client with a specific screen geometry:

```
vncviewer --geometry {{width}}x{{height}} {{host}}:
{{display_number}}
```

- Launch a VNC client which connects to a host on a given port:

```
vncviewer {{host}}::{{port}}
```

vnstat

A console-based network traffic monitor.

More information: https://manned.org/vnstat.

- Display traffic summary for all interfaces:

```
vnstat
```

- Display traffic summary for a specific network interface:

```
vnstat {{[-i|--iface]}} {{network_interface}}
```

- Display live stats for a specific network interface:

```
vnstat {{[-l|--live]}} {{[-i|--iface]}} {{network_interface}}
```

- Show traffic statistics on an hourly basis for the last 24 hours using a bar graph:

```
vnstat {{[-hg|--hoursgraph]}}
```

- Measure and show average traffic for 30 seconds:

```
vnstat {{[-tr|--traffic]}} {{30}}
```

vnstati

PNG image output support for vnStat.

More information: https://manned.org/vnstati.

- Output a summary of the last 2: months, days, and all-time:

```
vnstati --summary --iface {{network_interface}} --output
{{path/to/output.png}}
```

- Output the 10 most traffic-intensive days of all time:

```
vnstati --top 10 --iface {{network_interface}} --output
{{path/to/output.png}}
```

- Output monthly traffic statistics from the last 12 months:

```
vnstati --months --iface {{network_interface}} --output
{{path/to/output.png}}
```

- Output hourly traffic statistics from the last 24 hours:

```
vnstati --hours --iface {{network_interface}} --output
{{path/to/output.png}}
```

vpnc

A VPN client for the Cisco 3000 VPN Concentrator.

More information: https://manned.org/vpnc.

- Connect with a defined configuration file:

```
sudo vpnc {{config_file}}
```

- Terminate the previously created connection:

```
sudo vpnc-disconnect
```

vrms

Report non-free packages installed on Debian-based OSes.

More information: https://salsa.debian.org/debian/check-dfsg-status.

- List non-free and contrib packages (and their description):

```
vrms
```

- Only output the package names:

```
vrms --sparse
```

vso

Package manager, system updater and a task automator for Vanilla OS.

More information: https://github.com/Vanilla-OS/vanilla-system-operator.

- Check for system updates to the host system:

```
vso sys-upgrade check
```

- Upgrade the host system now:

```
vso sys-upgrade upgrade --now
```

- Initialize the Pico subsystem (used for package management):

```
vso pico-init
```

- Install applications inside the subsystem:

```
vso install {{package1 package2 ...}}
```

- Remove applications from the subsystem:

```
vso remove {{package1 package2 ...}}
```

- Enter the subsystem's shell:

```
vso shell
```

- Run an application from the subsystem:

```
vso run {{package}}
```

- Display VSO configuration:

```
vso config show
```

vzdump

Backup Utility for virtual machines and containers.

More information: https://pve.proxmox.com/pve-docs/vzdump.1.html.

- Dump a guest virtual machine into the default dump directory (usually `/var/lib/vz/dump/`), excluding snapshots:

```
vzdump {{vm_id}}
```

- Back up the guest virtual machines with the IDs 101, 102, and 103:

```
vzdump {{101 102 103}}
```

- Dump a guest virtual machine using a specific mode:

```
vzdump {{vm_id}} --mode {{suspend|snapshot}}
```

- Back up all guest systems and send an notification email to the root and admin users:

```
vzdump --all --mode {{suspend}} --mailto {{root}} --mailto {{admin}}
```

- Use snapshot mode (no downtime required) and a non-default dump directory:

```
vzdump {{vm_id}} --dumpdir {{path/to/directory}} --mode {{snapshot}}
```

- Back up all guest virtual machines excluding the IDs 101 and 102:

```
vzdump --mode {{suspend}} --exclude {{101, 102}}
```

waitpid

Wait for the termination of arbitrary processes.

See also: **`wait`**.

More information: https://manned.org/waitpid.1.

- Sleep until all processes whose PIDs have been specified have exited:

```
waitpid {{pid1 pid2 ...}}
```

- Sleep for at most n seconds:

```
waitpid {{[-t|--timeout]}} {{n}} {{pid1 pid2 ...}}
```

- Do not error if specified PIDs have already exited:

```
waitpid {{[-e|--exited]}} {{pid1 pid2 ...}}
```

- Sleep until n of the specified processes have exited:

```
waitpid {{[-c|--count]}} {{n}} {{pid1 pid2 ...}}
```

- Display help:

```
waitpid {{[-h|--help]}}
```

wajig

Simplified all-in-one-place system support tool for Debian-based systems.

More information: https://togaware.com/linux/survivor/wajig.html.

- Update the list of available packages and versions:

```
wajig update
```

- Install a package, or update it to the latest available version:

```
wajig install {{package}}
```

- Remove a package and its configuration files:

```
wajig purge {{package}}
```

- Perform an update and then a dist-upgrade:

```
wajig daily-upgrade
```

- Display the sizes of installed packages:

```
wajig sizes
```

- List the version and distribution for all installed packages:

```
wajig versions
```

- List versions of upgradable packages:

```
wajig toupgrade
```

- Display packages which have some form of dependency on the given package:

```
wajig dependents {{package}}
```

wal-telegram

Generate themes for Telegram based the colors generated by pywal/wal.

More information: https://github.com/guillaumeboehm/wal-telegram.

- Generate with wal's palette and the current wallpaper (feh only):

```
wal-telegram
```

- Generate with wal's palette and a specified background image:

```
wal-telegram --background={{path/to/image}}
```

- Generate with wal's palette and a colored background based on the palette:

```
wal-telegram --tiled
```

- Apply a gaussian blur on the background image:

```
wal-telegram -g
```

- Specify a location for the generated theme (default is `$XDG_CACHE_HOME/wal-telegram` or `~/.cache/wal-telegram`):

```
wal-telegram --destination={{path/to/destination}}
```

- Restart the telegram app after generation:

```
wal-telegram --restart
```

wall

Write a message on the terminals of users currently logged in.

More information: https://manned.org/wall.

- Send a message:

```
wall {{message}}
```

- Send a message to users that belong to a specific group:

```
wall {{[-g|--group]}} {{group_name}} {{message}}
```

- Send a message from a file:

```
wall {{file}}
```

- Send a message with timeout (default 300):

```
wall {{[-t|--timeout]}} {{seconds}} {{file}}
```

wami

An open-source and easy-to-use tool that recommends suitable programs for tasks.

More information: https://github.com/evait-security/wami.

- Find expanded results in all categories from the lake and sort them in the specified order:

```
wami {{[-a|--show-all]}} {{[-S|--sort]}} {{asc|desc}} {{[-s|--search-all]}} {{search_string}}
```

- Search GitHub to find expanded results, sorted in descending order:

```
wami {{[-a|--show-all]}} {{[-S|--sort]}} desc --github {{search_string}}
```

- Search GitHub for topics that match the search string:

```
wami --list-topics {{search_string}}
```

- Search the lake for a tool used in pentests to query for default credentials and sort the results in descending order:

```
wami {{[-S|--sort]}} desc {{[-s|--search-all]}} pentest credential default
```

warpd

A modal keyboard driven pointer manipulation program.

More information: https://github.com/rvaiya/warpd/blob/master/warpd.1.md.

- Run warpd in normal mode:

```
warpd --normal
```

- Run warpd in hint mode:

```
warpd --hint
```

- Move cursor left:

```
<h>
```

- Move cursor down:

```
<j>
```

- Move cursor up:

```
<k>
```

- Move cursor right:

```
<l>
```

- Emulate left click:

```
<m>
```

waydroid

A container-based approach to boot a full Android system on a regular Linux system like Ubuntu.

More information: https://docs.waydro.id/usage/waydroid-command-line-options.

- Start Waydroid:

```
waydroid show-full-ui
```

- Initialize Waydroid (required on first run or after reinstalling Android):

```
sudo waydroid init
```

- Install a new Android app from a file:

```
waydroid app install {{path/to/file.apk}}
```

- Launch an Android app by its package name:

```
waydroid app launch {{com.example.app}}
```

- Start or stop the Waydroid session:

```
waydroid session {{start|stop}}
```

- Manage the Waydroid container:

```
sudo waydroid container {{start|stop|restart|freeze|
unfreeze}}
```

- Open Waydroid shell:

```
sudo waydroid shell
```

- Adjust Waydroid window dimensions:

```
waydroid prop set persist.waydroid.{{width|height}}
{{number}}
```

waypipe

Remotely run graphical applications under a Wayland compositor.

More information: https://gitlab.freedesktop.org/mstoeckl/waypipe.

- Run a graphical program remotely and display it locally:

```
waypipe ssh {{user}}@{{server}} {{program}}
```

- Open an SSH tunnel to run any program remotely and display it locally:

```
waypipe ssh {{user}}@{{server}}
```

- Skip testing for Vulkan support:

```
waypipe --test-skip-vulkan ssh {{user}}@{{server}} {{program}}
```

- Display help:

```
waypipe {{[-h|--help]}}
```

wdctl

Show the hardware watchdog status.

More information: https://manned.org/wdctl.

- Display the watchdog status:

```
wdctl
```

- Display the watchdog status in a single line in key-value pairs:

```
wdctl {{[-O|--oneline]}}
```

- Display only specific watchdog flags (list is driver specific):

```
wdctl {{[-f|--flags]}} {{flag_list}}
```

wev

Print contents of Wayland events.

More information: https://manned.org/wev.

- Monitor all occurring Wayland events:

```
wev
```

- Print all events received by a specific Wayland interface:

```
wev -f {{wl_keyboard}}
```

- Print only specific events received by a Wayland interface:

```
wev -f {{wl_keyboard}}:{{key}}
```

- Print everything but the specified wayland events:

```
wev -F {{wl_keyboard}}:{{key}}
```

- Write the `wl_keyboards`'s keymap to a file:

```
wev -M {{path/to/file}}
```

- Print Wayland globals:

```
wev -g
```

wf-recorder

Screencast for Wayland optionally with audio.

By default you need to end the process with **`<Ctrl c>`**.

More information: https://github.com/ammen99/wf-recorder.

- Record storing to an MP4 file:

```
wf-recorder {{[-f|--file]}} {{output.mp4}}
```

- Record including audio, both with mic and system sounds:

```
wf-recorder {{[-a|--audio]}} {{[-f|--file]}} /{{path/to/file_with_audio.webm}}
```

- Select and record a portion of the screen using `slurp`, outputting to default `recording.mp4`:

```
wf-recorder {{[-g|--geometry]}} "$(slurp)"
```

wg-quick

Quickly set up WireGuard tunnels based on config files.

More information: https://manned.org/wg-quick.

- Set up a VPN tunnel:

```
wg-quick up {{interface_name}}
```

- Delete a VPN tunnel:

```
wg-quick down {{interface_name}}
```

wg

Manage the configuration of WireGuard interfaces.

More information: https://www.wireguard.com/quickstart/.

- Check status of currently active interfaces:

```
sudo wg
```

- Generate a new private key:

```
wg genkey
```

- Generate a public key from a private key:

```
wg < {{path/to/private_key}} pubkey > {{path/to/public_key}}
```

- Generate a public and private key:

```
wg genkey | tee {{path/to/private_key}} | wg pubkey > {{path/to/public_key}}
```

- Show the current configuration of a wireguard interface:

```
sudo wg showconf {{wg0}}
```

whatis

Display one-line descriptions from manual pages.

See also: **`whereis`**, **`which`**, **`type`**.

More information: https://manned.org/whatis.

- Display a description from a man page:

```
whatis {{command}}
```

- Don't cut the description off at the end of the line:

```
whatis --long {{command}}
```

- Display descriptions for all commands matching a glob:

```
whatis --wildcard {{net*}}
```

- Search man page descriptions with a `regex`:

```
whatis --regex '{{wish[0-9]\.[0-9]}}'
```

- Display descriptions in a specific language:

```
whatis --locale={{en}} {{command}}
```

whereis

Locate the binary, source, and manual page files for a command.

See also: **`which`**, **`whatis`**, **`type`**.

More information: https://manned.org/whereis.

- Locate binary, source and man pages for SSH:

```
whereis {{ssh}}
```

- Locate [b]inary and [m]an pages for ls:

```
whereis -bm {{ls}}
```

- Locate [s]ource of gcc and [m]an pages for Git:

```
whereis -s {{gcc}} -m {{git}}
```

- Locate [b]inaries for gcc in `/usr/bin/` only:

```
whereis -b -B {{/usr/bin/}} -f {{gcc}}
```

- Locate [u]nusual binaries (those that have more or less than one binary on the system):

```
whereis -u *
```

- Locate binaries that have [u]nusual [m]anual entries (binaries that have more or less than one manual installed):

```
whereis -u -m *
```

whiptail

Display text-based dialog boxes from shell scripts.

See also: **`dialog`**, **`gum`**.

More information: https://manned.org/whiptail.

- Display a simple message:

```
whiptail --title "{{title}}" --msgbox
"{{message}}" {{height_in_chars}} {{width_in_chars}}
```

- Display a boolean choice, returning the result through the exit code:

```
whiptail --title "{{title}}" --yesno
"{{message}}" {{height_in_chars}} {{width_in_chars}}
```

- Customise the text on the yes/no buttons:

```
whiptail --title "{{title}}" --yes-button "{{text}}" --no-
button "{{text}}" --yesno "{{message}}" {{height_in_chars}}
{{width_in_chars}}
```

- Display a text input box:

```
{{result_variable_name}}="$(whiptail --title "{{title}}" --
inputbox "{{message}}" {{height_in_chars}} {{width_in_chars}}
{{default_text}} 3>&1 1>&2 2>&3)"
```

- Display a password input box:

```
{{result_variable_name}}="$(whiptail --title "{{title}}" --
passwordbox "{{message}}" {{height_in_chars}}
{{width_in_chars}} 3>&1 1>&2 2>&3)"
```

- Display a multiple-choice menu:

```
{{result_variable_name}}=$(whiptail --title "{{title}}" --
menu "{{message}}" {{height_in_chars}} {{width_in_chars}}
{{menu_display_height}} {{"value_1" "display_text_1"
"value_2" "display_text_2" ...}} 3>&1 1>&2 2>&3)
```

wifi-menu

Interactively connect to a wireless network.

More information: https://manned.org/wifi-menu.

- Set up a wireless connection interactively:

```
wifi-menu
```

- Interactively set up a connection to a network and obscure the password:

```
wifi-menu {{[-o|--obscure]}}
```

- Display help:

```
wifi-menu {{[-h|--help]}}
```

wine

Run Windows executables on Unix-based systems.

More information: https://gitlab.winehq.org/wine/wine/-/wikis/Commands.

- Run a specific program inside the `wine` environment:

```
wine {{command}}
```

- Run a specific program in background:

```
wine start {{command}}
```

- Install/uninstall an MSI package:

```
wine msiexec /{{i|x}} {{path/to/package.msi}}
```

- Run `File Explorer`, `Notepad`, or `WordPad`:

```
wine {{explorer|notepad|write}}
```

- Run `Registry Editor`, `Control Panel`, or `Task Manager`:

```
wine {{regedit|control|taskmgr}}
```

- Run the configuration tool:

```
wine winecfg
```

winetricks

Manage Wine virtual Windows environments.

More information: https://wiki.winehq.org/Winetricks.

- Start a graphical setup at the default Wine location:

```
winetricks
```

- Specify a custom Wine directory to run Winetricks in:

```
WINEPREFIX={{path/to/wine_directory}} winetricks
```

- Install a Windows DLL or component to the default Wine directory:

```
winetricks {{package}}
```

wipefs

Wipe filesystem, raid, or partition-table signatures from a device.

More information: https://manned.org/wipefs.

- Display signatures for specified device:

```
sudo wipefs {{/dev/sdX}}
```

- Wipe all available signature types for a specific device with no recursion into partitions:

```
sudo wipefs {{[-a|--all]}} {{/dev/sdX}}
```

- Wipe all available signature types for the device and partitions using a glob pattern:

```
sudo wipefs {{[-a|--all]}} {{/dev/sdX}}*
```

- Perform dry run:

```
sudo wipefs {{[-a|--all]}} {{[-n|--no-act]}} {{/dev/sdX}}
```

- Force wipe, even if the filesystem is mounted:

```
sudo wipefs {{[-a|--all]}} {{[-f|--force]}} {{/dev/sdX}}
```

wireplumber

A modular session/policy manager for PipeWire and a GObject-based high-level library that wraps PipeWire's API.

See also: **`wpctl`**, **`pipewire`**.

More information: https://pipewire.pages.freedesktop.org/wireplumber/.

- Make WirePlumber start with the user session immediately (for systemd systems):

```
systemctl --user --now enable wireplumber
```

- Run WirePlumber, after `pipewire` is started (for non-systemd systems):

```
wireplumber
```

- Specify a different context configuration file:

```
wireplumber --config-file {{path/to/file}}
```

- Display help:

```
wireplumber --help
```

- Display version:

```
wireplumber --version
```

wl-copy

Clear and copy to Wayland clipboard.

See also: **`wl-paste`**, **`xclip`**.

More information: https://github.com/bugaevc/wl-clipboard.

- Copy the text to the clipboard:

```
wl-copy "{{text}}"
```

- Pipe the command (`ls`) output to the clipboard:

```
{{ls}} | wl-copy
```

- Copy for only one paste and then clear it:

```
wl-copy --paste-once "{{text}}"
```

- Copy an image:

```
wl-copy < {{path/to/image}}
```

- Clear the clipboard:

```
wl-copy --clear
```

wl-paste

Paste content in Wayland clipboard.

See also: **`wl-copy`**, **`xclip`**.

More information: https://github.com/bugaevc/wl-clipboard.

- Paste the contents of the clipboard:

```
wl-paste
```

- Paste the contents of the primary clipboard (highlighted text):

```
wl-paste --primary
```

- Write the contents of the clipboard to a file:

```
wl-paste > {{path/to/file}}
```

- Pipe the contents of the clipboard to a command:

```
wl-paste | {{command}}
```

wmctrl

CLI for X Window Manager.

More information: https://manned.org/wmctrl.

- List all windows, managed by the window manager:

```
wmctrl -l
```

- Switch to the first window whose (partial) title matches:

```
wmctrl -a {{window_title}}
```

- Move a window to the current workspace, raise it and give it focus:

```
wmctrl -R {{window_title}}
```

- Switch to a workspace:

```
wmctrl -s {{workspace_number}}
```

- Select a window and toggle fullscreen:

```
wmctrl -r {{window_title}} -b toggle,fullscreen
```

- Select a window and move it to a workspace:

```
wmctrl -r {{window_title}} -t {{workspace_number}}
```

wodim

Command (aliased as **`cdrecord`** on some systems) for recording data to CDs or DVDs.

Some invocations of wodim can cause destructive actions, such as erasing all the data on a disc.

More information: https://manned.org/wodim.

- Display optical drives available to `wodim`:

```
wodim --devices
```

- Record ("burn") an audio-only disc:

```
wodim dev={{/dev/optical_drive}} -audio {{track*.cdaudio}}
```

- Burn a file to a disc, ejecting the disc once done (some recorders require this):

```
wodim -eject dev={{/dev/optical_drive}} -data {{file.iso}}
```

- Burn a file to the disc in an optical drive, potentially writing to multiple discs in succession:

```
wodim -tao dev={{/dev/optical_drive}} -data {{file.iso}}
```

woeusb

Windows media creation tool.

More information: https://github.com/WoeUSB/WoeUSB.

- Format a USB then create a bootable Windows installation drive:

```
woeusb --device {{path/to/windows.iso}} {{/dev/sdX}}
```

- Copy Windows files to an existing partition of a USB storage device and make it bootable, without erasing the current data:

```
woeusb --partition {{path/to/windows.iso}} {{/dev/sdXN}}
```

wofi

An application launcher for wlroots-based Wayland compositors, similar to **`rofi`** and **`dmenu`**.

More information: https://manned.org/wofi.

- Show the list of apps:

```
wofi {{[-S|--show]}} drun
```

- Show the list of all commands:

```
wofi {{[-S|--show]}} run
```

- Pipe a list of items to `stdin` and print the selected item to `stdout`:

```
printf "{{Choice1\nChoice2\nChoice3}}" | wofi {{[-d|--dmenu]}}
```

wol

Client for sending Wake-on-LAN magic packets.

More information: https://sourceforge.net/projects/wake-on-lan/.

- Send a WoL packet to a device:

```
wol {{mac_address}}
```

- Send a WoL packet to a device in another subnet based on its IP:

```
wol {{[-i|--ipaddr]}} {{ip_address}} {{mac_address}}
```

- Send a WoL packet to a device in another subnet based on its hostname:

```
wol {{[-h|--host]}} {{hostname}} {{mac_address}}
```

- Send a WoL packet to a specific port on a host:

```
wol {{[-p|--port]}} {{port_number}} {{mac_address}}
```

- Read hardware addresses, IP addresses/hostnames, optional ports and SecureON passwords from a file:

```
wol {{[-f|--file]}} {{path/to/file}}
```

- Turn on verbose output:

```
wol {{[-v|--verbose]}} {{mac_address}}
```

wpa_cli

Add and configure wlan interfaces.

More information: https://manned.org/wpa_cli.

- Scan for available networks:

```
sudo wpa_cli scan
```

- Show scan results:

```
sudo wpa_cli scan_results
```

- Add a network:

```
sudo wpa_cli {{[add_n|add_network]}} {{number}}
```

- Set a network's SSID:

```
sudo wpa_cli {{[set_n|set_network]}} {{number}} ssid
"{{SSID}}"
```

- Enable network:

```
sudo wpa_cli {{[en|enable_network]}} {{number}}
```

- Save config:

```
sudo wpa_cli {{[sa|save_config]}}
```

wpa_passphrase

Generate a WPA-PSK key from an ASCII passphrase for a SSID.

More information: https://manned.org/wpa_passphrase.1.

- Compute and display the WPA-PSK key for a given SSID reading the passphrase from `stdin`:

```
wpa_passphrase {{SSID}}
```

- Compute and display WPA-PSK key for a given SSID specifying the passphrase as an argument:

```
wpa_passphrase {{SSID}} {{passphrase}}
```

wpctl

Manage WirePlumber, a session and policy manager for PipeWire.

Note: You can use the special name **`@DEFAULT_SINK@`** in place of **`id`** to operate on the default sink.

See also: **`pw-cli`**.

More information: https://pipewire.pages.freedesktop.org/wireplumber/.

- List all objects managed by WirePlumber:

```
wpctl status
```

- Print all properties of an object:

```
wpctl inspect {{id}}
```

- Set an object to be the default in its group:

```
wpctl set-default {{id}}
```

- Get the volume of a sink:

```
wpctl get-volume {{id}}
```

- Set the volume of a sink to `n` percent:

```
wpctl set-volume {{id}} {{n}}%
```

- Increase/Decrease the volume of a sink by `n` percent:

```
wpctl set-volume {{id}} {{n}}%{{+|-}}
```

- Increase the volume of a sink by `n` percent but limit the volume below 100%:

```
wpctl set-volume {{[-l|--limit]}} 1 {{id}} {{n}}%-
```

- Mute/Unmute a sink (1 is mute, 0 is unmute):

```
wpctl set-mute {{id}} {{1|0|toggle}}
```

wpexec

Run WirePlumber Lua scripts.

See also: **`wpctl`**, **`wireplumber`**.

More information: https://pipewire.pages.freedesktop.org/wireplumber/scripting/lua_api/lua_introduction.html.

- Run a WirePlumber script:

```
wpexec {{path/to/file.lua}}
```

- Display help:

```
wpexec --help
```

write

Write a message on the terminal of a specified logged in user (**<Ctrl c>** to stop writing messages).

Use the **who** command to find out all terminal_ids of all active users active on the system.

See also: **mesg**.

More information: https://manned.org/write.

- Send a message to a given user on a given terminal ID:

```
write {{username}} {{terminal_id}}
```

- Send message to "testuser" on terminal `/dev/tty/5`:

```
write {{testuser}} {{tty/5}}
```

- Send message to "johndoe" on pseudo terminal `/dev/pts/5`:

```
write {{johndoe}} {{pts/5}}
```

wtf

Show the expansions of acronyms.

More information: https://manned.org/wtf.6.

- Expand a given acronym:

```
wtf {{IMO}}
```

- Specify a computer related search type:

```
wtf -t {{comp}} {{WWW}}
```

wtype

Simulate keyboard input on Wayland, similar to **`xdotool type`** for X11.

See also: **`ydotool`**.

More information: https://manned.org/wtype.

- Simulate typing text:

```
wtype "{{Hello World}}"
```

- Type a specific key:

```
wtype -k {{Left}}
```

- Press a modifier:

```
wtype -M {{shift|ctrl|...}}
```

- Release a modifier:

```
wtype -m {{ctrl}}
```

- Wait between keystrokes (in milliseconds):

```
wtype -d {{500}} -- "{{text}}"
```

- Read text from `stdin`:

```
echo "{{text}}" | wtype -
```

x0vncserver

TigerVNC Server for X displays.

More information: https://tigervnc.org/doc/x0vncserver.html.

- Start a VNC server using a passwordfile:

```
x0vncserver -display {{:0}} -passwordfile {{path/to/file}}
```

- Start a VNC server using a specific port:

```
x0vncserver -display {{:0}} -rfbport {{port}}
```

x11vnc

A VNC server that will enable VNC on an existing display server.

By default, the server will automatically terminate once all clients disconnect from it.

More information: https://manned.org/x11vnc.

- Launch a VNC server that allows multiple clients to connect:

```
x11vnc -shared
```

- Launch a VNC server in view-only mode, and which won't terminate once the last client disconnects:

```
x11vnc -forever -viewonly
```

- Launch a VNC server on a specific display and screen (both starting at index zero):

```
x11vnc -display :{{display}}.{{screen}}
```

- Launch a VNC server on the third display's default screen:

```
x11vnc -display :{{2}}
```

- Launch a VNC server on the first display's second screen:

```
x11vnc -display :{{0}}.{{1}}
```

x86_64

This command is an alias of **`setarch x86_64`**.

- View documentation for the original command:

  `tldr setarch`

xauth

Edit and display the authorization information used in connecting to the X server.

More information: https://manned.org/xauth.

- Start interactive mode with a specific authority file (defaults to `~/.Xauthority`):

```
xauth -f {{path/to/file}}
```

- Display information about the authority file:

```
xauth info
```

- Display authorization entries for all the displays:

```
xauth list
```

- Add an authorization for a specific display:

```
xauth add {{display_name}} {{protocol_name}} {{key}}
```

- Remove the authorization for a specific display:

```
xauth remove {{display_name}}
```

- Print the authorization entry for the current display to `stdout`:

```
xauth extract - $DISPLAY
```

- Merge the authorization entries from a specific file into the authorization database:

```
cat {{path/to/file}} | xauth merge -
```

- Display help:

```
xauth --help
```

xbacklight

Utility to adjust backlight brightness using the RandR extension.

More information: https://manned.org/xbacklight.

- Get the current screen brightness as a percentage:

```
xbacklight
```

- Set the screen brightness to 40%:

```
xbacklight -set {{40}}
```

- Increase current brightness by 25%:

```
xbacklight -inc {{25}}
```

- Decrease current brightness by 75%:

```
xbacklight -dec {{75}}
```

- Increase backlight to 100%, over 60 seconds (value given in ms), using 60 steps:

```
xbacklight -set {{100}} -time {{60000}} -steps {{60}}
```

xbps-install

XBPS utility to (re)install and update packages.

See also: **xbps**.

More information: https://manned.org/xbps-install.1.

- Install a new package:

```
xbps-install {{package}}
```

- Synchronize and update all packages:

```
xbps-install {{[-S|--sync]}} {{[-u|--update]}}
```

xbps-query

XBPS utility to query for package and repository information.

See also: **xbps**.

More information: https://manned.org/xbps-query.1.

- Search for a package in remote repositories using a `regex` or a keyword (if `--regex` is omitted):

```
xbps-query {{[-s|--search]}} {{regex|keyword}} --repository --regex
```

- Show information about an installed package:

```
xbps-query {{[-S|--show]}} {{package}}
```

- Show information about a package in remote repositories:

```
xbps-query {{[-S|--show]}} {{package}} --repository
```

- List packages registered in the package database:

```
xbps-query {{[-l|--list-pkgs]}}
```

- List explicitly installed packages (i.e. not automatically installed as dependencies):

```
xbps-query {{[-m|--list-manual-pkgs]}}
```

xbps-remove

XBPS utility to remove packages.

See also: **xbps**.

More information: https://manned.org/xbps-remove.1.

- Remove a package:

```
xbps-remove {{package}}
```

- Remove a package and its dependencies:

```
xbps-remove {{[-R|--recursive]}} {{package}}
```

- Remove orphan packages (installed as dependencies but no longer required by any package):

```
xbps-remove {{[-o|--remove-orphans]}}
```

- Remove obsolete packages from the cache:

```
xbps-remove {{[-O|--clean-cache]}}
```

xbps

The X Binary Package System is the package manager used by Void Linux.

For equivalent commands in other package managers, see https://wiki.archlinux.org/title/Pacman/Rosetta.

More information: https://docs.voidlinux.org/xbps/index.html.

- View documentation for installing and updating packages:

```
tldr xbps-install
```

- View documentation for removing packages:

```
tldr xbps-remove
```

- View documentation for querying for package and repository information:

```
tldr xbps-query
```

xclip

X11 clipboard manipulation tool, similar to **`xsel`**.

Handles the X primary and secondary selections, plus the system clipboard (**`<Ctrl c>`**/**`<Ctrl v>`**).

See also: **`wl-copy`**.

More information: https://manned.org/xclip.

- Copy the output from a command to the X11 primary selection area (clipboard):

```
echo 123 | xclip
```

- Copy the output from a command to a given X11 selection area:

```
echo 123 | xclip {{[-se|-selection]}} {{primary|secondary|clipboard}}
```

- Copy the output from a command to the system clipboard, using short notation:

```
echo 123 | xclip {{[-se|-selection]}} {{[c|clipboard]}}
```

- Copy the contents of a file into the system clipboard:

```
xclip {{[-se|-selection]}} {{[c|clipboard]}} {{input_file.txt}}
```

- Copy the contents of a PNG into the system clipboard (can be pasted in other programs correctly):

```
xclip {{[-se|-selection]}} {{[c|clipboard]}} {{[-t|-target]}} image/png {{input_file.png}}
```

- Copy the user input in the console into the system clipboard:

```
xclip {{[-i|-in]}}
```

- Paste the contents of the X11 primary selection area to the console:

```
xclip {{[-o|-out]}}
```

- Paste the contents of the system clipboard to the console:

```
xclip {{[-o|-out]}} {{[-se|-selection]}} {{[c|clipboard]}}
```

xclock

Display the time in analog or digital form.

More information: https://manned.org/xclock.

- Display an analog clock:

```
xclock
```

- Display a 24-hour digital clock with the hour and minute fields only:

```
xclock -digital -brief
```

- Display a digital clock using an strftime format string (see strftime(3)):

```
xclock -digital -strftime {{format}}
```

- Display a 24-hour digital clock with the hour, minute and second fields that updates every second:

```
xclock -digital -strftime '%H:%M:%S' -update 1
```

- Display a 12-hour digital clock with the hour and minute fields only:

```
xclock -digital -twelve -brief
```

xcowsay

Display a cute cow and message on your Linux desktop.

The cow is displayed for either a fixed amount of time, or an amount of time calculated from the size of the text. Click on the cow to dismiss it immediately.

More information: https://manned.org/xcowsay.

- Display a cow saying "hello, world":

```
xcowsay "{{hello, world}}"
```

- Display a cow with output from another command:

```
ls | xcowsay
```

- Display a cow at the specified X and Y coordinates:

```
xcowsay --at {{X}},{{Y}}
```

- Display a different sized cow:

```
xcowsay --cow-size {{small|med|large}}
```

- Display a thought bubble instead of a speech bubble:

```
xcowsay --think
```

- Display a different image instead of the default cow:

```
xcowsay --image {{path/to/file}}
```

xcursorgen

Create an X cursor file from a collection of PNGs.

If **`--prefix`** is omitted, the image files must be located in the current working directory.

More information: https://manned.org/xcursorgen.

- Create an X cursor file using a configuration file:

```
xcursorgen {{path/to/config.cursor}} {{path/to/output_file}}
```

- Create an X cursor file using a configuration file and specify the path to the image files:

```
xcursorgen --prefix {{path/to/image_directory}}/ {{path/to/config.cursor}} {{path/to/output_file}}
```

- Create an X cursor file using a configuration file and write the output to `stdout`:

```
xcursorgen {{path/to/config.cursor}}
```

xdg-desktop-menu

Install or uninstall desktop menu items.

More information: https://manned.org/xdg-desktop-menu.

- Install an application to the desktop menu system:

```
xdg-desktop-menu install {{path/to/file.desktop}}
```

- Install an application to the desktop menu system with the vendor prefix check disabled:

```
xdg-desktop-menu install --novendor {{path/to/file.desktop}}
```

- Uninstall an application from the desktop menu system:

```
xdg-desktop-menu uninstall {{path/to/file.desktop}}
```

- Force an update of the desktop menu system:

```
xdg-desktop-menu forceupdate --mode {{user|system}}
```

xdg-mime

Query and manage MIME types according to the XDG standard.

More information: https://portland.freedesktop.org/doc/xdg-mime.html.

- Display the MIME type of a file:

```
xdg-mime query filetype {{path/to/file}}
```

- Display the default application for opening PNGs:

```
xdg-mime query default {{image/png}}
```

- Display the default application for opening a specific file:

```
xdg-mime query default $(xdg-mime query filetype {{path/to/file}})
```

- Set imv as the default application for opening PNG and JPEG images:

```
xdg-mime default {{imv.desktop}} {{image/png}} {{image/jpeg}}
```

xdg-open

Open a file or URL in the user's preferred application.

More information: https://portland.freedesktop.org/doc/xdg-open.html.

- Open the current directory in the default file explorer:

```
xdg-open .
```

- Open a URL in the default browser:

```
xdg-open {{https://example.com}}
```

- Open an image in the default image viewer:

```
xdg-open {{path/to/image}}
```

- Open a PDF in the default PDF viewer:

```
xdg-open {{path/to/pdf}}
```

- Display help:

```
xdg-open --help
```

xdg-settings

Manage settings of XDG-compatible desktop environments.

More information: https://portland.freedesktop.org/doc/xdg-settings.html.

- Print the default web browser:

```
xdg-settings get {{default-web-browser}}
```

- Set the default web browser to Firefox:

```
xdg-settings set {{default-web-browser}} {{firefox.desktop}}
```

- Set the default mail URL scheme handler to Evolution:

```
xdg-settings set {{default-url-scheme-handler}} {{mailto}}
{{evolution.desktop}}
```

- Set the default PDF document viewer:

```
xdg-settings set {{pdf-viewer.desktop}}
```

- Display help:

```
xdg-settings --help
```

xdg-user-dir

Retrieve XDG user directory locations.

See also: **`xdg-user-dirs-update`**.

More information: https://manned.org/xdg-user-dir.

- Display the home directory for the current user:

```
xdg-user-dir
```

- Display the location of a user directory:

```
xdg-user-dir {{DESKTOP|DOWNLOAD|TEMPLATES|PUBLICSHARE|
DOCUMENTS|MUSIC|PICTURES|VIDEOS}}
```

xdg-user-dirs-update

Update XDG user directories.

See also: **`xdg-user-dir`**.

More information: https://manned.org/xdg-user-dirs-update.

- Change XDG's DESKTOP directory to the specified directory (must be absolute):

```
xdg-user-dirs-update --set DESKTOP "{{path/to/directory}}"
```

- Write the result to the specified dry-run-file instead of the `user-dirs.dirs` file:

```
xdg-user-dirs-update --dummy-output "{{path/to/
dry_run_file}}" --set {{xdg_user_directory}} "{{path/to/
directory}}"
```

xdotool

Automate X11 actions.

More information: https://manned.org/xdotool.

- Retrieve the X-Windows window ID of the running Firefox window(s):

```
xdotool search --onlyvisible --name {{firefox}}
```

- Perform a mouse <RightClick>:

```
xdotool click {{3}}
```

- Get the ID of the currently active window:

```
xdotool getactivewindow
```

- Focus on the window with ID of 12345:

```
xdotool windowfocus --sync {{12345}}
```

- Type a message, with a 500ms delay for each letter:

```
xdotool type --delay {{500}} "Hello world"
```

- Press the <Enter> key:

```
xdotool key {{KP_Enter}}
```

xdp-filter

Load and manage an eBPF XDP packet filter.

Part of the xdp-tools collection.

More information: https://github.com/xdp-project/xdp-tools/tree/main/xdp-filter#running-xdp-filter.

- Load the filter on an interface in skb (generic) mode with default allow policy:

```
sudo xdp-filter load {{[-p|--policy]}} allow {{[-m|--mode]}} skb {{network_interface}}
```

- Unload the filter from an interface:

```
sudo xdp-filter unload {{network_interface}}
```

- Deny traffic to a specific destination port:

```
sudo xdp-filter port {{destination_port}}
```

- Deny traffic from a specific source IP address:

```
sudo xdp-filter ip {{[-m|--mode]}} src {{source_ip}}
```

- Deny traffic from a specific source MAC address:

```
sudo xdp-filter ether {{[-m|--mode]}} src {{mac_address}}
```

- Poll packets and show statistics every 10000 milliseconds:

```
sudo xdp-filter poll {{[-i|--interval]}} 10000
```

xed

Edit files in Cinnamon desktop environment.

More information: https://github.com/linuxmint/xed.

- Start the editor:

```
xed
```

- Open specific files:

```
xed {{path/to/file1 path/to/file2 ...}}
```

- Open files using a specific encoding:

```
xed --encoding {{WINDOWS-1252}} {{path/to/file1 path/to/
file2 ...}}
```

- Print all supported encodings:

```
xed --list-encodings
```

- Open a file and go to a specific line:

```
xed +{{10}} {{path/to/file}}
```

xeyes

Display eyes on the screen that follow the mouse cursor.

More information: https://manned.org/xeyes.

- Launch xeyes on the local machine's default display:

```
xeyes
```

- Launch xeyes on a remote machine's display 0, screen 0:

```
xeyes -display {{remote_host}}:{{0}}.{{0}}
```

xfce4-screenshooter

The XFCE4 screenshot tool.

More information: https://docs.xfce.org/apps/xfce4-screenshooter/start.

- Launch the screenshooter GUI:

```
xfce4-screenshooter
```

- Take a screenshot of the entire screen and launch the GUI to ask how to proceed:

```
xfce4-screenshooter {{[-f|--fullscreen]}}
```

- Take a screenshot of the entire screen and save it in the specified directory:

```
xfce4-screenshooter {{[-f|--fullscreen]}} {{[-s|--save]}} {{path/to/directory}}
```

- Wait some time before taking the screenshot:

```
xfce4-screenshooter {{[-d|--delay]}} {{seconds}}
```

- Take a screenshot of a region of the screen (select using the mouse):

```
xfce4-screenshooter {{[-r|--region]}}
```

- Take a screenshot of the active window, and copy it to the clipboard:

```
xfce4-screenshooter {{[-w|--window]}} {{[-c|--clipboard]}}
```

- Take a screenshot of the active window, and open it with a chosen program:

```
xfce4-screenshooter {{[-w|--window]}} {{[-o|--open]}} {{gimp}}
```

xfce4-terminal

The XFCE4 terminal emulator.

More information: https://docs.xfce.org/apps/xfce4-terminal/start.

- Open a new terminal window:

```
xfce4-terminal
```

- Set the initial title:

```
xfce4-terminal --initial-title "{{initial_title}}"
```

- Open a new tab in the current terminal window:

```
xfce4-terminal --tab
```

- Execute a command in a new terminal window:

```
xfce4-terminal --command "{{command_with_args}}"
```

- Keep the terminal around after the executed command finishes executing:

```
xfce4-terminal --command "{{command_with_args}}" --hold
```

- Open multiple new tabs, executing a command in each:

```
xfce4-terminal --tab --command "{{command1}}" --tab --command "{{command2}}"
```

xfreerdp

Free Remote Desktop Protocol implementation.

More information: https://github.com/FreeRDP/FreeRDP/wiki/CommandLineInterface-(possibly-not-up-to-date,-check-application-help-text-for-most-up-to-date-version).

- Connect to a FreeRDP server:

```
xfreerdp /u:{{username}} /p:{{password}} /v:{{ip_address}}
```

- Connect to a FreeRDP server and activate audio output redirection using `sys:alsa` device:

```
xfreerdp /u:{{username}} /p:{{password}} /v:{{ip_address}} /sound:{{sys:alsa}}
```

- Connect to a FreeRDP server with dynamic resolution:

```
xfreerdp /v:{{ip_address}} /u:{{username}} /p:{{password}} /dynamic-resolution
```

- Connect to a FreeRDP server with clipboard redirection:

```
xfreerdp /v:{{ip_address}} /u:{{username}} /p:{{password}} +clipboard
```

- Connect to a FreeRDP server ignoring any certificate checks:

```
xfreerdp /v:{{ip_address}} /u:{{username}} /p:{{password}} /cert:ignore
```

- Connect to a FreeRDP server with a shared directory:

```
xfreerdp /v:{{ip_address}} /u:{{username}} /p:{{password}} /drive:{{path/to/directory}},{{share_name}}
```

xfs_repair

Repair an XFS filesystem.

More information: https://manned.org/xfs_repair.

- Repair a partition:

```
sudo xfs_repair {{path/to/partition}}
```

xhost

Manage access control lists for X server connections.

More information: https://manned.org/xhost.

- Display the current access control list:

```
xhost
```

- Allow a specific host to connect to the X server:

```
xhost +{{hostname}}
```

- Deny a specific host from connecting to the X server:

```
xhost -{{hostname}}
```

- Allow all hosts to connect (disable access control - insecure):

```
xhost +
```

- Deny all hosts except those explicitly allowed (enable access control):

```
xhost -
```

- Remove a specific user or address using a family prefix (like `inet:hostname` or `si:localuser:username`):

```
xhost -{{family:name}}
```

xinput

List available input devices, query information about a device and change input device settings.

More information: https://manned.org/xinput.

- List all input devices:

```
xinput list
```

- Disable an input:

```
xinput disable {{id}}
```

- Enable an input:

```
xinput enable {{id}}
```

- Disconnect an input from its master:

```
xinput float {{id}}
```

- Reattach an input as slave to a master:

```
xinput reattach {{id}} {{master_id}}
```

- List settings of an input device:

```
xinput list-props {{id}}
```

- Change a setting of an input device:

```
xinput set-prop {{id}} {{setting_id}} {{value}}
```

xman

Manual page viewer for X Window System.

More information: https://manned.org/xman.

- Start `xman` in three-button window:

  ```
  xman
  ```

- Open the manual page output stored in a given file:

  ```
  xman -helpfile {{filename}}
  ```

- Show both manual page and directory:

  ```
  xman -bothshown
  ```

xmodmap

Utility for modifying keymaps and pointer button mappings in X.

More information: https://manned.org/xmodmap.

- Swap `<LeftClick>` and `<RightCLick>` on the pointer:

```
xmodmap -e 'pointer = 3 2 1'
```

- Reassign a key on the keyboard to another key:

```
xmodmap -e 'keycode {{keycode}} = {{keyname}}'
```

- Disable a key on the keyboard:

```
xmodmap -e 'keycode {{keycode}} ='
```

- Execute all xmodmap expressions in the specified file:

```
xmodmap {{path/to/file}}
```

xmount

Convert on-the-fly between multiple input and output hard disk image types with optional write cache support.

Create a virtual file system using FUSE (Filesystem in Userspace) that contains a virtual representation of the input image.

More information: https://manned.org/xmount.

- Mount a `.raw` image file into a DMG container file:

```
xmount --in {{raw}} {{path/to/image.dd}} --out {{dmg}} {{mountpoint}}
```

- Mount an EWF image file with write-cache support into a VHD file to boot from:

```
xmount --cache {{path/to/cache.ovl}} --in {{ewf}} {{path/to/image.E??}} --out {{vhd}} {{mountpoint}}
```

- Mount the first partition at sector 2048 into a new `.raw` image file:

```
xmount --offset {{2048}} --in {{raw}} {{path/to/image.dd}} --out {{raw}} {{mountpoint}}
```

xrandr

Set the size, orientation and/or reflection of the outputs for a screen.

More information: https://www.x.org/releases/current/doc/man/man1/xrandr.1.xhtml.

- Display the current state of the system (known screens, resolutions, ...):

```
xrandr {{[-q|--query]}}
```

- Disable disconnected outputs and enable connected ones with default settings:

```
xrandr --auto
```

- Change the resolution and update frequency of DisplayPort 1 to 1920x1080, 60Hz:

```
xrandr --output DP1 --mode 1920x1080 {{[-r|--rate]}} 60
```

- Set the resolution of HDMI2 to 1280x1024 and put it on the right of DP1:

```
xrandr --output HDMI2 --mode 1280x1024 --right-of DP1
```

- Disable the VGA1 output:

```
xrandr --output VGA1 --off
```

- Set the brightness for LVDS1 to 50%:

```
xrandr --output LVDS1 --brightness 0.5
```

- Display the current state of any X server:

```
xrandr {{[-d|--display]}} :{{0}} {{[-q|--query]}}
```

xrdb

X window server's resource database utility for Unix-like systems.

More information: https://www.x.org/releases/current/doc/man/man1/xrdb.1.xhtml.

- Start `xrdb` in interactive mode:

```
xrdb
```

- Load values (e.g. style rules) from a resource file:

```
xrdb -load {{~/.Xresources}}
```

- Query the resource database and print currently set values:

```
xrdb -query
```

xsel

X11 selection and clipboard manipulation tool.

More information: https://manned.org/xsel.

- Use a command's output as input of the clipboard (equivalent to <Ctrl c>):

```
echo 123 | xsel {{[-ib|--input --clipboard]}}
```

- Use the contents of a file as input of the clipboard:

```
cat {{path/to/file}} | xsel {{[-ib|--input --clipboard]}}
```

- Output the clipboard's contents into the terminal (equivalent to <Ctrl v>):

```
xsel {{[-ob|--output --clipboard]}}
```

- Output the clipboard's contents into a file:

```
xsel {{[-ob|--output --clipboard]}} > {{path/to/file}}
```

- Clear the clipboard:

```
xsel {{[-cb|--clear --clipboard]}}
```

- Output the X11 primary selection's contents into the terminal (equivalent to a mouse <MiddleClick>):

```
xsel {{[-op|--output --primary]}}
```

xset

User preference utility for X.

More information: https://manned.org/xset.

- Disable the screensaver:

```
xset s off
```

- Disable the bell sound:

```
xset b off
```

- Set the screensaver to start after 60 minutes of inactivity:

```
xset s 3600 3600
```

- Disable DPMS (Energy Star) features:

```
xset -dpms
```

- Enable DPMS (Energy Star) features:

```
xset +dpms
```

- Query information on any X server:

```
xset -display :{{0}} q
```

xsetwacom

Change settings for Wacom pen tablets at runtime.

More information: https://manned.org/xsetwacom.

- List all the available Wacom devices. The device name is in the first column:

```
xsetwacom list
```

- Set Wacom area to specific screen. Get name of the screen with `xrandr`:

```
xsetwacom set "{{device_name}}" MapToOutput {{screen}}
```

- Set mode to relative (like a mouse) or absolute (like a pen) mode:

```
xsetwacom set "{{device_name}}" Mode "{{Relative|Absolute}}"
```

- Rotate the input (useful for tablet-PC when rotating screen) by 0|90|180|270 degrees from "natural" rotation:

```
xsetwacom set "{{device_name}}" Rotate {{none|half|cw|ccw}}
```

- Set button to only work when the tip of the pen is touching the tablet:

```
xsetwacom set "{{device_name}}" TabletPCButton "on"
```

xterm

A terminal emulator for the X Window System.

More information: https://manned.org/xterm.

- Open the terminal with a title of `Example`:

```
xterm -T {{Example}}
```

- Open the terminal in fullscreen mode:

```
xterm -fullscreen
```

- Open the terminal with a dark blue background and yellow foreground (font color):

```
xterm -bg {{darkblue}} -fg {{yellow}}
```

- Open the terminal with 100 characters per line and 35 lines, in screen position x=200px, y=20px:

```
xterm -geometry {{100}}x{{35}}+{{200}}+{{20}}
```

- Open the terminal using a Serif font and a font size equal to 20:

```
xterm -fa '{{Serif}}' -fs {{20}}
```

xtrlock

Lock the X display until the user supplies their password.

More information: https://manned.org/xtrlock.

- Lock the display and show a padlock instead of the cursor:

```
xtrlock
```

- Display a blank screen as well as the padlock cursor:

```
xtrlock -b
```

- Fork the xtrlock process and return immediately:

```
xtrlock -f
```

xvfb-run

Run a command in a virtual X server environment.

More information: https://manned.org/xvfb-run.

- Run the specified command in a virtual X server:

```
xvfb-run {{command}}
```

- Try to get a free server number, if the default (99) is not available:

```
xvfb-run {{[-a|--auto-servernum]}} {{command}}
```

- Pass arguments to the Xvfb server:

```
xvfb-run {{[-s|--server-args]}} "{{-screen 0
1024x768x24}}" {{command}}
```

xwinwrap

Run a player or a program as desktop background.

More information: https://github.com/ujjwal96/xwinwrap.

- Run a video using mpv:

```
xwinwrap -b -nf -ov -- {{mpv}} --wid {{wid}} --loop --no-audio --no-resume-playback --panscan={{1.0}} {{path/to/video.mp4}}
```

- Run a video in fullscreen using mpv:

```
xwinwrap -b -nf -fs -ov -- {{mpv}} --wid {{wid}} --loop --no-audio --no-resume-playback --panscan={{1.0}} {{path/to/video.mp4}}
```

- Run a video using mpv with 80% opacity:

```
xwinwrap -b -nf -ov -o 0.8 --- {{mpv}} --wid {{wid}} --loop --no-audio --no-resume-playback --panscan={{1.0}} {{path/to/video.mp4}}
```

- Run a video using mpv in a second monitor 1600x900 with 1920 offset on X-axis:

```
xwinwrap -g 1600x900+1920 -b -nf -ov -- {{mpv}} --wid {{wid}} --loop --no-audio --no-resume-playback --panscan={{1.0}} {{path/to/video.mkv}}
```

xxhsum

Print or verify checksums using fast non-cryptographic algorithm xxHash.

More information: https://manned.org/xxhsum.

- Calculate the checksum for a file using a specific algorithm:

```
xxhsum -H{{0|32|64|128}} {{path/to/file}}
```

- Run benchmark:

```
xxhsum -b
```

yakuake

A drop-down terminal for KDE.

See also: **guake**.

More information: https://manned.org/yakuake.

- Start up Yakuake:

  `yakuake`

- Toggle Yakuake visibility:

  `<F12>`

- Toggle fullscreen mode:

  `<F11>`

- Open a new tab:

  `<Ctrl Shift t>`

- Switch between tabs:

  `<Shift {{ArrowLeft|ArrowRight}}>`

- Create terminal splits:

  `<Ctrl {{)|(}}>`

- Switch between splits:

  `{{<Ctrl Tab>|<Ctrl Shift Tab>}}`

yaourt

Arch Linux utility for building packages from the Arch User Repository.

More information: https://archlinux.fr/yaourt-en.

- Synchronize and update all packages (including AUR):

```
yaourt -Syua
```

- Install a new package (includes AUR):

```
yaourt -S {{package}}
```

- Remove a package and its dependencies (includes AUR packages):

```
yaourt -Rs {{package}}
```

- Search the package database for a keyword (including AUR):

```
yaourt -Ss {{query}}
```

- List installed packages, versions, and repositories (AUR packages will be listed under the repository name `local`):

```
yaourt -Q
```

yay

Yet Another Yogurt: build and install packages from the Arch User Repository.

See also: **pacman**.

More information: https://github.com/Jguer/yay.

- Interactively search and install packages from the repos and AUR:

```
yay {{package_name|search_term}}
```

- Synchronize and update all packages from the repos and AUR:

```
yay
```

- Install a new package from the repos and AUR and do not ask to confirm transactions:

```
yay -S {{package}} --noconfirm
```

- Remove an installed package and both its dependencies and configuration files:

```
yay -Rns {{package}}
```

- Search the package database for a keyword from the repos and AUR:

```
yay -Ss {{keyword}}
```

- Remove orphaned packages (installed as dependencies but not required by any package):

```
yay -Yc
```

- Clean `pacman` and `yay` caches (old package versions kept for rollback and downgrade purposes):

```
yay -Scc
```

- Show statistics for installed packages and system health:

```
yay -Ps
```

ydotool

Control keyboard and mouse inputs via commands in a way that is display server agnostic.

More information: https://github.com/ReimuNotMoe/ydotool.

- Start the ydotool daemon in the background:

```
ydotoold
```

- Perform a `<LeftClick>` input:

```
ydotool click 0xC0
```

- Perform a `<RightClick>` input:

```
ydotool click 0xC1
```

- Input `<Alt F4>`:

```
ydotool key 56:1 62:1 62:0 56:0
```

yetris

Clone of the game Tetris in the terminal.

More information: https://github.com/alexdantas/yetris.

- Start a Tetris game:

  `yetris`

- Navigate the piece horizontally:

  `{{<ArrowLeft>|<ArrowRight>}}`

- Rotate the piece clockwise or counterclockwise:

  `{{<x>|<z>}}`

- Hold a piece (only one allowed at a time):

  `<c>`

- Soft drop the piece:

  `<ArrowDown>`

- Hard drop the piece:

  `<Space>`

- Pause/unpause the game:

  `<p>`

- Quit the game:

  `<q>`

yplan

Generate LaTeX code for a two-page vertical daily planner for any chosen year.

The generated output can be converted or printed using conversion tools such as **`pandoc`**, **`pdflatex`**, or **`xetex`**.

More information: https://www.ctan.org/tex-archive/macros/latex/contrib/yplan.

- Create a daily planner with specified language, lettercase (uppercase or lowercase) and year:

```
yplan {{language}} {{lettercase}} {{year}} > {{path/to/file.tex}}
```

ytfzf

Find and download videos and music. Written in POSIX shell.

See also: **`youtube-dl`**, **`yt-dlp`**, **`instaloader`**.

More information: https://github.com/pystardust/ytfzf.

- Search for videos on YouTube with thumbnail previews:

```
ytfzf --show-thumbnails {{search_pattern}}
```

- Play only the audio of the first item in a loop:

```
ytfzf --audio-only --auto-select --loop {{search_pattern}}
```

- Download a video from the history:

```
ytfzf --download --choose-from-history
```

- Play all the music found in a search:

```
ytfzf --audio-only --select-all {{search_pattern}}
```

- See the trending videos in an external menu:

```
ytfzf --trending --ext-menu {{search_pattern}}
```

- Search on PeerTube instead of YouTube:

```
ytfzf --peertube {{search_pattern}}
```

yum config-manager

This command is an alias of **`dnf config-manager`**.

- View documentation for the original command:

  ```
  tldr dnf config-manager
  ```

yum

This command is an alias of **`dnf`** on modern distros.

For older distros (e.g. CentOS 7), **`yum`** is a distinct program from **`dnf`**. Some subcommands and options may be different.

- View documentation for the original command:

```
tldr dnf
```

yumdownloader

Historical YUM package downloader for Fedora installations; now deprecated.

This command is an alias of **`dnf download`**.

- View documentation for the original command:

  ```
  tldr dnf download
  ```

zathura

A Vim-like modal document viewer, with an integrated command-line.

Make sure a backend is installed (poppler, PostScript, or DjVu).

More information: https://pwmt.org/projects/zathura/.

- Open a file:

```
zathura {{path/to/file}}
```

- Navigate left/up/down/right:

```
{{<h>|<j>|<k>|<l>|<ArrowKeys>}}
```

- Rotate:

```
<r>
```

- Invert Colors:

```
<Ctrl r>
```

- Search for text by a given string:

```
</>{{string}}
```

- Create/delete bookmarks:

```
<:>{{bmark|bdelete}} {{bookmark_name}}<Enter>
```

- List bookmarks:

```
<:>blist<Enter>
```

zbarcam

Scan and decode barcodes (and QR codes) from a video device.

More information: https://manned.org/zbarcam.

- Continuously read barcodes and print them to `stdout`:

```
zbarcam
```

- Disable output video window while scanning:

```
zbarcam --nodisplay
```

- Print barcodes without type information:

```
zbarcam --raw
```

- Define capture device:

```
zbarcam {{/dev/video_device}}
```

zdump

Print timezone information.

More information: https://manned.org/zdump.

- Print the current time in a timezone:

```
zdump {{timezone}}
```

- Display help:

```
zdump --help
```

zenity

Display dialogs from the command-line/shell scripts.

Return user-inserted values or 1 if error.

More information: https://manned.org/zenity.

- Display the default question dialog:

```
zenity --question
```

- Display an info dialog displaying the text "Hello!":

```
zenity --info --text="{{Hello!}}"
```

- Display a name/password form and output the data separated by ";":

```
zenity --forms --add-entry="{{Name}}" --add-
password="{{Password}}" --separator="{{;}}"
```

- Display a file selection form in which the user can only select directories:

```
zenity --file-selection --directory
```

- Display a progress bar which updates its message every second and show a progress percent:

```
{{(echo "#1"; sleep 1; echo "50"; echo "#2"; sleep 1; echo
"100")}} | zenity --progress
```

zforce

Add a **`.gz`** extension to files compressed using **`gzip`**.

More information: https://manned.org/zforce.

- Add a `.gz` extension to the supplied Gzip files (Note: Other files are ignored):

```
zforce {{path/to/file1 path/to/file2 ...}}
```

zic

Compile timezones into binary files.

More information: https://manned.org/zic.

- Compile a timezone file from a directory:

```
zic -d {{path/to/directory}}
```

- Report warnings during compilation of a specific file:

```
zic -v {{path/to/file}}.infile
```

zile

A lightweight clone of the Emacs text editor.

More information: https://manned.org/zile.

- Start a buffer for temporary notes, which won't be saved:

```
zile
```

- Open a file:

```
zile {{path/to/file}}
```

- Save a file:

```
<Ctrl x><Ctrl s>
```

- Quit:

```
<Ctrl x><Ctrl c>
```

- Open a file at a specified line number:

```
zile +{{line_number}} {{path/to/file}}
```

- Undo changes:

```
<Ctrl x><u>
```

zipsplit

Split a Zip archive into smaller Zip archives.

More information: https://manned.org/zipsplit.

- Split Zip archive into parts that are no larger than 36000 bytes (36 MB):

```
zipsplit {{path/to/archive.zip}}
```

- Use a given [n]umber of bytes as the part limit:

```
zipsplit -n {{size}} {{path/to/archive.zip}}
```

- [p]ause between the creation of each part:

```
zipsplit -p -n {{size}} {{path/to/archive.zip}}
```

- Output the smaller Zip archives into a given directory:

```
zipsplit -b {{path/to/output_directory}} -n {{size}} {{path/to/archive.zip}}
```

znc

IRC bouncer.

More information: https://manned.org/znc.

- Run the initial setup:

```
znc {{[-c|--makeconf]}}
```

- Start the IRC bouncer daemon:

```
znc
```

- Setup `znc` for systemd:

```
sudo {{[-u|--user]}} znc znc {{[-c|--makeconf]}} {{[-d|--datadir]}} /var/lib/znc
```

- Enable `znc` to start on boot and start it now:

```
systemctl enable znc --now
```

zramctl

Setup and control zram devices.

Use **`mkfs`** or **`mkswap`** to format zram devices to partitions.

More information: https://manned.org/zramctl.

- Check if zram is enabled:

```
lsmod | grep {{[-i|--ignore-case]}} zram
```

- Enable zram with a dynamic number of devices (use `zramctl` to configure devices further):

```
sudo modprobe zram
```

- Enable zram with exactly 2 devices:

```
sudo modprobe zram num_devices={{2}}
```

- Find and initialize the next free zram device to a 2 GB virtual drive using LZ4 compression:

```
sudo zramctl {{[-f|--find]}} {{[-s|--size]}} {{2GB}} {{[-a|--algorithm]}} {{lz4}}
```

- List currently initialized devices:

```
sudo zramctl
```

zsync

Partial/differential file downloader.

HTTPS is not supported - use HTTP URLs only.

See also: **`rsync`**.

More information: https://manned.org/zsync.

- Download a file using a `.zsync` control file:

```
zsync {{path/to/url.zsync}}
```

- Use a local file as a seed to avoid re-downloading unchanged parts:

```
zsync -i {{path/to/existing_file}} {{path/to/url.zsync}}
```

- Save the updated file under a specific name:

```
zsync -i {{path/to/existing_file}} -o {{path/to/new_file}} {{path/to/url.zsync}}
```

- Resume a partial download and keep the temporary file:

```
zsync -k {{path/to/url.zsync}}
```

- Run in quiet mode with minimal output (no progress bar, download rate, or ETA display):

```
zsync -q {{path/to/url.zsync}}
```

zypper

SUSE & openSUSE package management utility.

For equivalent commands in other package managers, see https://wiki.archlinux.org/title/Pacman/Rosetta.

More information: https://en.opensuse.org/SDB:Zypper_manual.

- Synchronize list of packages and versions available:

```
sudo zypper {{[ref|refresh]}}
```

- Install a new package:

```
sudo zypper {{[in|install]}} {{package}}
```

- Remove a package:

```
sudo zypper {{[rm|remove]}} {{package}}
```

- Upgrade installed packages to the newest available versions:

```
sudo zypper {{[up|update]}}
```

- Search package via keyword:

```
zypper {{[se|search]}} {{keyword}}
```

- Show information related to configured repositories:

```
zypper {{[lr|repos]}} --sort-by-priority
```

Netbsd

cal

Display a calendar.

More information: https://man.netbsd.org/cal.1.

- Display a calendar for the current month:

```
cal
```

- Display a calendar for a specific year:

```
cal {{year}}
```

- Display a calendar for a specific month and year:

```
cal {{month}} {{year}}
```

- Display the whole calendar for the current year using [j]ulian days (one-based, numbered from January 1):

```
cal -y -j
```

- [h]ighlight today and display [3] months spanning the date:

```
cal -h -3 {{month}} {{year}}
```

- Display the 2 months [B]efore and 3 [A]fter a specific [m]onth of the current year:

```
cal -A 3 -B 2 {{month}}
```

- Display a specific number of months before and after ([C]ontext) the specified month:

```
cal -C {{months}} {{month}}
```

- Specify the starting [d]ay of the week (0: Sunday, 1: Monday, ..., 6: Saturday):

```
cal -d {{0..6}}
```

chfn

This command is an alias of **chpass**.

- View documentation for the original command:

```
tldr chpass
```

chpass

Add or change user database information, including login shell and password.

See also: **passwd**.

More information: https://man.netbsd.org/chpass.

- Set a specific login shell for the current user interactively:

```
su -c chpass
```

- Set a specific login [s]hell for the current user:

```
chpass -s {{path/to/shell}}
```

- Set a login [s]hell for a specific user:

```
chpass -s {{path/to/shell}} {{username}}
```

- Specify a user database entry in the `passwd` file format:

```
su -c 'chpass -a {{username:encrypted_password:uid:gid:...}}
-s {{path/to/shell}}' {{username}}
```

- Only update the [l]ocal password file:

```
su -c 'chpass -l -s {{path/to/shell}}' {{username}}
```

- Forcedly change the database [y]P password database entry:

```
su -c 'chpass -y -s {{path/to/shell}}' {{username}}
```

chsh

This command is an alias of **chpass**.

- View documentation for the original command:

```
tldr chpass
```

df

Display an overview of the filesystem disk space usage.

More information: https://man.netbsd.org/df.1.

- Display all filesystems and their disk usage (using 512-byte units):

```
df
```

- Use [h]uman-readable units (based on powers of 1024):

```
df -h
```

- Display all the fields of the structure(s) returned by `statvfs`:

```
df -G
```

- Display the filesystem containing the specified file or directory:

```
df {{path/to/file_or_directory}}
```

- Include statistics on the number of free and used [i]nodes:

```
df -i
```

- Use [k]ibibyte (1024 byte) units when showing size figures:

```
df -k
```

- Display information in a [P]ortable way:

```
df -P
```

pkgin

Manage **`pkgsrc`** binary packages on NetBSD.

More information: https://pkgin.net/#usage.

- Install a package:

```
pkgin install {{package}}
```

- Remove a package and its dependencies:

```
pkgin remove {{package}}
```

- Upgrade all packages:

```
pkgin full-upgrade
```

- Search for a package:

```
pkgin search {{keyword}}
```

- List installed packages:

```
pkgin list
```

- Remove unneeded dependencies:

```
pkgin autoremove
```

sed

Edit text in a scriptable manner.

See also: **awk**, **ed**.

More information: https://man.netbsd.org/sed.1.

- Replace all `apple` (basic `regex`) occurrences with `mango` (basic `regex`) in all input lines and print the result to `stdout`:

```
{{command}} | sed 's/apple/mango/g'
```

- Execute a specific script [f]ile and print the result to `stdout`:

```
{{command}} | sed -f {{path/to/script.sed}}
```

- Delay opening each file until a command containing the related `w` function or flag is applied to a line of input:

```
{{command}} | sed -fa {{path/to/script.sed}}
```

- Turn on GNU re[g]ex extension:

```
{{command}} | sed -fg {{path/to/script.sed}}
```

- Replace all `apple` (extended `regex`) occurrences with `APPLE` (extended `regex`) in all input lines and print the result to `stdout`:

```
{{command}} | sed -E 's/(apple)/\U\1/g'
```

- Print just a first line to `stdout`:

```
{{command}} | sed -n '1p'
```

- Replace all `apple` (basic `regex`) occurrences with `mango` (basic `regex`) in a specific file and overwrite the original file in place:

```
sed -i 's/apple/mango/g' {{path/to/file}}
```

sockstat

List open Internet or UNIX domain sockets.

Note: This program is a rewrite for NetBSD 3.0 from FreeBSD's **`sockstat`**.

See also: **`netstat`**.

More information: https://man.netbsd.org/sockstat.1.

- Show information for IPv4, IPv6 and Unix sockets for both listening and connected sockets:

```
sockstat
```

- Show information for IPv[4]/IPv[6] sockets [l]istening on specific [p]orts using a specific [P]rotocol:

```
sockstat -{{4|6}} -l -P {{tcp|udp|sctp|divert}} -p {{port1,port2...}}
```

- Also show [c]onnected sockets, showing [u]nix sockets:

```
sockstat -cu
```

- Only show [n]umeric output, without resolving symbolic names for addresses and ports:

```
sockstat -n
```

- Only list sockets of the specified address [f]amily:

```
sockstat -f {{inet|inet6|local|unix}}
```

Openbsd

cal

Display a calendar with the current day highlighted.

More information: https://man.openbsd.org/cal.

- Display a calendar for the current month:

```
cal
```

- Display a calendar for a specific year:

```
cal {{year}}
```

- Display a calendar for a specific month and year:

```
cal {{month}} {{year}}
```

- Display a calendar for the current [y]ear:

```
cal -y
```

- Display [j]ulian days (starting from one, numbered from January 1):

```
cal -j
```

- Use [m]onday as week start instead of Sunday:

```
cal -m
```

- Number [w]eek numbers (incompatible with `-j`):

```
cal -w
```

chfn

This command is an alias of **chpass**.

- View documentation for the original command:

```
tldr chpass
```

chpass

Add or change user database information, including login shell and password.

See also: **passwd**.

More information: https://man.openbsd.org/chpass.

- Set a specific login shell for the current user interactively:

```
doas chpass
```

- Set a specific login [s]hell for the current user:

```
doas chpass -s {{path/to/shell}}
```

- Set a login [s]hell for a specific user:

```
doas chpass -s {{path/to/shell}} {{username}}
```

- Specify a user database entry in the `passwd` file format:

```
doas chpass -a {{username:encrypted_password:uid:gid:...}}
```

chsh

This command is an alias of **chpass**.

- View documentation for the original command:

```
tldr chpass
```

df

Display an overview of the filesystem disk space usage.

More information: https://man.openbsd.org/df.1.

- Display all filesystems and their disk usage (using 512-byte units):

```
df
```

- Display all filesystems in [h]uman-readable form (based on powers of 1024):

```
df -h
```

- Display the filesystem containing the specified file or directory:

```
df {{path/to/file_or_directory}}
```

- Include statistics on the number of free and used [i]nodes:

```
df -i
```

- Use [k]ibibyte (1024 byte) units when showing size figures:

```
df -k
```

- Display information in a [P]ortable way:

```
df -P
```

pkg

OpenBSD package manager utility.

More information: https://www.openbsd.org/faq/faq15.html.

- View documentation for installing/updating packages:

```
tldr pkg_add
```

- View documentation for removing packages:

```
tldr pkg_delete
```

- View documentation for viewing information about packages:

```
tldr pkg_info
```

pkg_add

Install/update packages in OpenBSD.

See also: **`pkg_info`**, **`pkg_delete`**.

More information: https://man.openbsd.org/pkg_add.

- Update all packages, including dependencies:

```
pkg_add -u
```

- Install a new package:

```
pkg_add {{package}}
```

- Install packages from the raw output of `pkg_info`:

```
pkg_add -l {{path/to/file}}
```

pkg_delete

Remove packages in OpenBSD.

See also: **`pkg_add`**, **`pkg_info`**.

More information: https://man.openbsd.org/pkg_delete.

- Delete a package:

```
pkg_delete {{package}}
```

- Delete a package, including its unused dependencies:

```
pkg_delete -a {{package}}
```

- Dry-run deletion of a package:

```
pkg_delete -n {{package}}
```

pkg_info

View information about packages in OpenBSD.

See also: **`pkg_add`**, **`pkg_delete`**.

More information: https://man.openbsd.org/pkg_info.

- Search for a package using the package name:

```
pkg_info -Q {{package}}
```

- Output a list of installed packages for use with `pkg_add -l`:

```
pkg_info -mz
```

sed

Edit text in a scriptable manner.

See also: **awk**, **ed**.

More information: https://man.openbsd.org/sed.1.

- Replace all `apple` (basic `regex`) occurrences with `mango` (basic `regex`) in all input lines and print the result to `stdout`:

```
{{command}} | sed 's/apple/mango/g'
```

- Execute a specific script [f]ile and print the result to `stdout`:

```
{{command}} | sed -f {{path/to/script.sed}}
```

- Delay opening each file until a command containing the related `w` function or flag is applied to a line of input:

```
{{command}} | sed -fa {{path/to/script.sed}}
```

- Replace all `apple` (extended `regex`) occurrences with `APPLE` (extended `regex`) in all input lines and print the result to `stdout`:

```
{{command}} | sed -E 's/(apple)/\U\1/g'
```

- Print just a first line to `stdout`:

```
{{command}} | sed -n '1p'
```

- Replace all `apple` (basic `regex`) occurrences with `mango` (basic `regex`) in a specific file and overwrite the original file in place:

```
sed -i 's/apple/mango/g' {{path/to/file}}
```

Osx

aa

This command is an alias of **yaa**.

- View documentation for the original command:

```
tldr yaa
```

afinfo

Audio file metadata parser for OS X.

Built-in command of OS X.

More information: https://keith.github.io/xcode-man-pages/afinfo.1.html.

- Display info of a given audio file:

```
afinfo {{path/to/file}}
```

- Print a one line description of the audio file:

```
afinfo --brief {{path/to/file}}
```

- Print metadata info and contents of the audio file's InfoDictionary:

```
afinfo --info {{path/to/file}}
```

- Print output in XML format:

```
afinfo --xml {{path/to/file}}
```

- Print warnings for the audio file if any:

```
afinfo --warnings {{path/to/file}}
```

- Display help:

```
afinfo --help
```

afplay

Command-line audio player.

More information: https://keith.github.io/xcode-man-pages/afplay.1.html.

- Play a sound file (waits until playback ends):

```
afplay {{path/to/file}}
```

- Play a sound file at 2x speed (playback rate):

```
afplay --rate {{2}} {{path/to/file}}
```

- Play a sound file at half speed:

```
afplay --rate {{0.5}} {{path/to/file}}
```

- Play the first n seconds of a sound file:

```
afplay --time {{n}} {{path/to/file}}
```

aiac

Use OpenAI to generate IaC configurations, utilities, queries and more.

More information: https://github.com/gofireflyio/aiac.

- Generate Terraform for Azure storage account:

```
aiac get terraform {{for an azure storage account}}
```

- Generate a Dockerfile for nginx:

```
aiac get dockerfile {{for a secured nginx}}
```

- Generate GitHub action that applies Terraform:

```
aiac get github action {{that plans and applies terraform}}
```

- Generate a port scanner in Python:

```
aiac get python {{code that scans all open ports in my network}}
```

- Generate a MongoDB query:

```
aiac get mongo {{query that aggregates all documents by created date}}
```

airport

Wireless network configuration utility.

More information: https://ss64.com/mac/airport.html.

- Show current wireless status information:

```
airport --getinfo
```

- Sniff wireless traffic on channel 1:

```
airport sniff {{1}}
```

- Scan for available wireless networks:

```
airport --scan
```

- Disassociate from current airport network:

```
sudo airport --disassociate
```

airportd

Manage wireless interfaces.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/airportd.8.html.

- Start the daemon:

```
airportd
```

apachectl

Apache HTTP Server control interface for macOS.

More information: https://keith.github.io/xcode-man-pages/apachectl.8.html.

- Start the `org.apache.httpd` launchd job:

  `apachectl start`

- Stop the launchd job:

  `apachectl stop`

- Stop, then start launchd job:

  `apachectl restart`

applecamerad

Camera manager.

It should not be invoked manually.

More information: https://www.theiphonewiki.com/wiki/Services.

- Start the daemon:

```
applecamerad
```

appsleepd

Start app sleep services.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/appsleepd.8.html.

- Start the daemon:

```
appsleepd
```

arch

Display the name of the system architecture, or run a command under a different architecture.

See also: **uname**.

More information: https://keith.github.io/xcode-man-pages/arch.1.html.

- Display the system's architecture:

```
arch
```

- Run a command using x86_64:

```
arch -x86_64 "{{command}}"
```

- Run a command using arm:

```
arch -arm64 "{{command}}"
```

archey

Stylishly display system information.

More information: https://github.com/joshfinnie/archey-osx.

- Show system information:

```
archey
```

- Show system information without colored output:

```
archey --nocolor
```

- Show system information, using MacPorts instead of Homebrew:

```
archey --macports
```

- Show system information without IP address check:

```
archey --offline
```

as

Portable GNU assembler.

Primarily intended to assemble output from **`gcc`** to be used by **`ld`**.

More information: https://keith.github.io/xcode-man-pages/as.1.html.

- Assemble a file, writing the output to `a.out`:

```
as {{path/to/file.s}}
```

- Assemble the output to a given file:

```
as {{path/to/file.s}} -o {{path/to/output_file.o}}
```

- Generate output faster by skipping whitespace and comment preprocessing. (Should only be used for trusted compilers):

```
as -f {{path/to/file.s}}
```

- Include a given path to the list of directories to search for files specified in `.include` directives:

```
as -I {{path/to/directory}} {{path/to/file.s}}
```

asr

Restore (copy) a disk image onto a volume.

The command name stands for Apple Software Restore.

More information: https://keith.github.io/xcode-man-pages/asr.8.html.

- Restore a disk image to a target volume:

```
sudo asr restore --source {{image_file.dmg}} --target {{path/to/volume_file}}
```

- Erase the target volume before restoring:

```
sudo asr restore --source {{image_file.dmg}} --target {{path/to/volume_file}} --erase
```

- Skip verification after restoring:

```
sudo asr restore --source {{image_file.dmg}} --target {{path/to/volume_file}} --noverify
```

- Clone volumes without using an intermediate disk image:

```
sudo asr restore --source {{path/to/volume_file}} --target {{path/to/volume_file}}
```

autofsd

Run **`automount`** on startup and network configuration change events.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/autofsd.8.html.

- Start the daemon:

```
autofsd
```

automount

Read the **`/etc/auto_master`** file and mount **`autofs`** on the appropriate mount points to trigger the on-demand mounting of directories. Essentially, it's a way to manually initiate the system's automounting process.

Note: You'll most likely need to run with **`sudo`** if you don't have the necessary permissions.

More information: https://keith.github.io/xcode-man-pages/automount.8.html.

- Run automount, flush the cache(`-c`) beforehand, and be verbose(`-v`) about it (most common use):

  ```
  automount -cv
  ```

- Automatically unmount after 5 minutes (300 seconds) of inactivity:

  ```
  automount -t 300
  ```

- Unmount anything previously mounted by automount and/or defined in `/etc/auto_master`:

  ```
  automount -u
  ```

automountd

An automatic mount/unmount daemon for **`autofs`**. Started on demand by **`launchd`**.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/automountd.8.html.

- Start the daemon:

```
automountd
```

- Log more details to `syslog`:

```
automountd -v
```

autoraise

Automatically raise and/or focus a window when hovering over it with the mouse.

More information: https://github.com/sbmpost/AutoRaise.

- Run AutoRaise in the background:

```
autoraise &
```

auvaltool

AudioUnit validation tool for Mac.

More information: https://keith.github.io/xcode-man-pages/auvaltool.1.html.

- List all [a]vailable AudioUnits of any type:

```
auvaltool -a
```

- List all [a]vailable AudioUnits of any type with their [l]ocation:

```
auvaltool -al
```

avbdeviced

A service for managing Audio Video Bridging (AVB) devices.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/avbdeviced.1.html.

- Start the daemon:

```
avbdeviced
```

backupd

Create Time Machine backups and manages its backup history.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/backupd.8.html.

- Start the daemon:

```
backupd
```

base64

Encode or decode file or **`stdin`** to/from base64, to **`stdout`** or another file.

More information: https://keith.github.io/xcode-man-pages/bintrans.1.

- Encode a file to `stdout`:

```
base64 {{[-i|--input]}} {{path/to/file}}
```

- Encode a file to the specified output file:

```
base64 {{[-i|--input]}} {{path/to/input_file}} {{[-o|--output]}} {{path/to/output_file}}
```

- Wrap encoded output at a specific width (`0` disables wrapping):

```
base64 {{[-b|--break]}} {{0|76|...}} {{path/to/file}}
```

- Decode a file to `stdout`:

```
base64 {{[-d|--decode]}} {{[-i|--input]}} {{path/to/file}}
```

- Encode from `stdin` to `stdout`:

```
{{command}} | base64
```

- Decode from `stdin` to `stdout`:

```
{{command}} | base64 {{[-d|--decode]}}
```

bc

An arbitrary precision calculator language.

See also: **dc**.

More information: https://keith.github.io/xcode-man-pages/bc.1.html.

- Start an interactive session:

```
bc
```

- Start an interactive session with the standard math library enabled:

```
bc --mathlib
```

- Calculate an expression:

```
bc --expression '{{5 / 3}}'
```

- Execute a script:

```
bc {{path/to/script.bc}}
```

- Calculate an expression with the specified scale:

```
bc --expression '{{scale = 10; 5 / 3}}'
```

- Calculate a sine/cosine/arctangent/natural logarithm/exponential function using `mathlib`:

```
bc --mathlib --expression '{{s|c|a|l|e}}({{1}})'
```

bclm

Set a custom charge limit on MacBooks.

More information: https://github.com/zackelia/bclm.

- Set the charge limit to about 80% (for Intel machines, the battery charge level may be slightly lower than the set value):

```
sudo bclm write {{77}}
```

- Read the current charge limit:

```
bclm read
```

- Keep the charge limit after rebooting/smc reset:

```
sudo bclm persist
```

- Remove the persistent charge limit:

```
sudo bclm unpersist
```

biomesyncd

Synchronizes data between devices registered to the same account.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/biomesyncd.8.html.

- Start the daemon:

```
biomesyncd
```

biometrickitd

Get support for biometric operations.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/biometrickitd.8.html.

- Start the daemon:

```
biometrickitd
```

bird

This supports the syncing of iCloud and iCloud Drive.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/bird.8.html.

- Start the daemon:

```
bird
```

bless

Set volume boot capability and startup disk options.

More information: https://keith.github.io/xcode-man-pages/bless.8.html.

- Bless a volume with only Mac OS X or Darwin, and create the BootX and `boot.efi` files as needed:

```
bless --folder {{/Volumes/Mac OS X/System/Library/
CoreServices}} --bootinfo --bootefi
```

- Set a volume containing either Mac OS 9 and Mac OS X to be the active volume:

```
bless --mount {{/Volumes/Mac OS}} --setBoot
```

- Set the system to NetBoot and broadcast for an available server:

```
bless --netboot --server {{bsdp://255.255.255.255}}
```

- Gather information about the currently selected volume (as determined by the firmware), suitable for piping to a program capable of parsing Property Lists:

```
bless --info --plist
```

bnepd

A service that handles all Bluetooth network connections.

It should not be invoked manually.

More information: https://www.manpagez.com/man/8/bnepd/.

- Start the daemon:

```
bnepd
```

brightness

Get and set the brightness level of all internal and certain external displays.

More information: https://github.com/nriley/brightness.

- Show current brightness:

```
brightness -l
```

- Set a specific brightness:

```
brightness {{0..1}}
```

- Set the brightness to 50%:

```
brightness {{0.5}}
```

caffeinate

Prevent macOS from sleeping.

More information: https://keith.github.io/xcode-man-pages/caffeinate.8.html.

- Prevent the display from sleeping:

```
caffeinate -d
```

- Prevent from sleeping for 1 hour (3600 seconds):

```
caffeinate -u -t {{3600}}
```

- Fork a process, exec "make" in it, and prevent sleep as long as that process is running:

```
caffeinate -i make
```

- Prevent from sleeping until a process with the specified PID completes:

```
caffeinate -w {{pid}}
```

- Prevent disk from sleeping (use `<Ctrl c>` to exit):

```
caffeinate -m
```

cal

Print calendar information.

More information: https://keith.github.io/xcode-man-pages/cal.1.html.

- Display a calendar for the current month:

```
cal
```

- Display previous, current, and next month:

```
cal -3
```

- Display a calendar for a specific month (1-12 or name):

```
cal -m {{month}}
```

- Display a calendar for the current year:

```
cal -y
```

- Display a calendar for a specific year (4 digits):

```
cal {{year}}
```

- Display a calendar for a specific month and year:

```
cal {{month}} {{year}}
```

- Display date of Easter (Western Christian churches) in a given year:

```
ncal -e {{year}}
```

carthage

A dependency management tool for Cocoa applications.

More information: https://github.com/Carthage/Carthage.

- Download the latest version of all dependencies mentioned in Cartfile, and build them:

  ```
  carthage update
  ```

- Update dependencies, but only build for iOS:

  ```
  carthage update --platform ios
  ```

- Update dependencies, but don't build any of them:

  ```
  carthage update --no-build
  ```

- Download and rebuild the current version of dependencies (without updating them):

  ```
  carthage bootstrap
  ```

- Rebuild a specific dependency:

  ```
  carthage build {{dependency}}
  ```

cat

Print and concatenate files.

More information: https://keith.github.io/xcode-man-pages/cat.1.html.

- Print the contents of a file to `stdout`:

```
cat {{path/to/file}}
```

- Concatenate several files into an output file:

```
cat {{path/to/file1 path/to/file2 ...}} > {{path/to/output_file}}
```

- Append several files to an output file:

```
cat {{path/to/file1 path/to/file2 ...}} >> {{path/to/output_file}}
```

- Copy the contents of a file into an output file without buffering:

```
cat -u {{/dev/tty12}} > {{/dev/tty13}}
```

- Write `stdin` to a file:

```
cat - > {{path/to/file}}
```

- Number all output lines:

```
cat -n {{path/to/file}}
```

- Display non-printable and whitespace characters (with `M-` prefix if non-ASCII):

```
cat -v -t -e {{path/to/file}}
```

cfprefsd

Start preferences services (**`CFPreferences`**, **`NSUserDefaults`**).

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/cfprefsd.8.html.

- Start the daemon:

```
cfprefsd
```

chflags

Change file or directory flags.

More information: https://keith.github.io/xcode-man-pages/chflags.1.html.

- Set the `hidden` flag for a file:

```
chflags {{hidden}} {{path/to/file}}
```

- Unset the `hidden` flag for a file:

```
chflags {{nohidden}} {{path/to/file}}
```

- Recursively set the `uchg` flag for a directory:

```
chflags -R {{uchg}} {{path/to/directory}}
```

- Recursively unset the `uchg` flag for a directory:

```
chflags -R {{nouchg}} {{path/to/directory}}
```

chpass

Add or change user database information, including login shell and password.

Note: It's not possible to change the user's password on Open Directory systems, use **passwd** instead.

See also: **passwd**.

More information: https://man.freebsd.org/cgi/man.cgi?chpass.

- Add or change user database information for the current user interactively:

```
su -c chpass
```

- Set a specific login [s]hell for the current user:

```
chpass -s {{path/to/shell}}
```

- Set a login [s]hell for a specific user:

```
chpass -s {{path/to/shell}} {{username}}
```

- Edit the user record on the directory node at the given [l]ocation:

```
chpass -l {{location}} -s {{path/to/shell}} {{username}}
```

- Use the given [u]sername when authenticating to the directory node containing the user:

```
chpass -u {{authname}} -s {{path/to/shell}} {{username}}
```

cloudd

Backs the CloudKit feature.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/cloudd.8.html.

- Start the daemon:

```
cloudd
```

cloudphotod

This synchronizes iCloud Photos.

It should not be invoked manually.

More information: https://www.manpagez.com/man/8/cloudphotosd/.

- Start the daemon:

```
cloudphotod
```

codesign

Create and manipulate code signatures for macOS.

More information: https://keith.github.io/xcode-man-pages/codesign.1.html.

- Sign an application with a certificate:

```
codesign --sign "{{My Company Name}}" {{path/to/application_file.app}}
```

- Verify the certificate of an application:

```
codesign --verify {{path/to/application_file.app}}
```

contactsd

Manages the information in your contacts database.

It provides functionality to apps using the Contacts API and performs various background maintenance tasks.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/contactsd.8.html.

- Start the daemon:

```
contactsd
```

coreaudiod

Service for Core Audio, Apple's audio system.

It should not be invoked manually.

More information: https://developer.apple.com/library/archive/documentation/MusicAudio/Conceptual/CoreAudioOverview/WhatisCoreAudio/WhatisCoreAudio.html.

- Start the daemon:

```
coreaudiod
```

coreautha

A system agent providing the **`LocalAuthentication`** framework.

It should not be invoked manually.

See also: **`coreauthd`**.

More information: https://keith.github.io/xcode-man-pages/coreautha.8.html.

- Start the agent:

```
coreautha
```

coreauthd

A system daemon providing the **`LocalAuthentication`** framework.

It should not be invoked manually.

See also: **`coreautha`**.

More information: https://keith.github.io/xcode-man-pages/coreauthd.8.html.

- Start the agent:

```
coreauthd
```

corebrightnessd

Manage Night Shift.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/corebrightnessd.8.html.

- Start the daemon:

```
corebrightnessd
```

coredatad

Schedules CloudKit operations for clients of NSPersistentCloudKitContainer.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/coredatad.8.html.

- Start the daemon:

```
coredatad
```

cot

The Plain-Text Editor for macOS.

More information: https://coteditor.com/cot.

- Start CotEditor:

```
cot
```

- Open specific files:

```
cot {{path/to/file1 path/to/file2 ...}}
```

- Open a new blank document:

```
cot --new
```

- Open a specific file and block the terminal until it is closed:

```
cot --wait {{path/to/file}}
```

- Open a specific file with the cursor at a specific line and column:

```
cot --line {{1}} --column {{80}} {{path/to/file}}
```

csrutil

Manage the System Integrity Protection configuration.

More information: https://keith.github.io/xcode-man-pages/csrutil.8.html.

- Display the System Integrity Protection status:

```
csrutil status
```

- Disable the System Integrity Protection:

```
csrutil disable
```

- Enable the System Integrity Protection:

```
csrutil enable
```

- Display the list of allowed NetBoot sources:

```
csrutil netboot list
```

- Add an IPv4 address to the list of allowed NetBoot sources:

```
csrutil netboot add {{ip}}
```

- Reset the System Integrity Protection status and clear the NetBoot list:

```
csrutil clear
```

csshX

Cluster SSH tool for macOS.

More information: https://github.com/brockgr/csshx.

- Connect to multiple hosts:

```
csshX {{hostname1}} {{hostname2}}
```

- Connect to multiple hosts with a given SSH key:

```
csshX {{user@hostname1}} {{user@hostname2}} --ssh_args "-i
{{path/to/key_file.pem}}"
```

- Connect to a pre-defined cluster from `/etc/clusters`:

```
csshX cluster1
```

ctkd

SmartCard daemon.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/ctkd.8.html.

- Start the daemon:

```
ctkd
```

cut

Cut out fields from **stdin** or files.

More information: https://keith.github.io/xcode-man-pages/cut.1.html.

- Print a specific character/field range of each line:

```
{{command}} | cut -{{c|f}} {{1|1,10|1-10|1-|-10}}
```

- Print a field range of each line with a specific delimiter:

```
{{command}} | cut -d "{{,}}" -f {{1}}
```

- Print a character range of each line of a specific file:

```
cut -c {{1}} {{path/to/file}}
```

cvfsck

Check and repair an Xsan file system volume.

Part of the Xsan file system utilities on macOS.

More information: https://www.manpagez.com/man/1/cvfsck/.

- Check an Xsan volume for metadata corruption (read-only mode):

```
sudo cvfsck {{/Volumes/XsanVolume}}
```

- Repair a corrupted Xsan volume (make modifications to fix problems):

```
sudo cvfsck -w {{/Volumes/XsanVolume}}
```

- Log problems to the system log (used mainly during automatic startup checks):

```
sudo cvfsck -l {{/Volumes/XsanVolume}}
```

- Report free-space fragmentation on an Xsan volume:

```
sudo cvfsck -f {{/Volumes/XsanVolume}}
```

dark-mode

Control macOS dark mode.

More information: https://github.com/sindresorhus/dark-mode.

- Toggle dark mode (turn it on if it's currently off, off if it's currently on):

```
dark-mode
```

- Turn dark mode on:

```
dark-mode on
```

- Turn dark mode off:

```
dark-mode off
```

- Check if dark mode is on:

```
dark-mode status
```

darwin-rebuild

Rebuild and switch to a Nix-based Darwin (macOS) system configuration.

More information: https://github.com/LnL7/nix-darwin.

- Rebuild and switch to the specified Darwin configuration:

```
darwin-rebuild switch --flake {{path/to/flake}}
```

- Build the configuration but don't switch to it:

```
darwin-rebuild build --flake {{path/to/flake}}
```

- Display help:

```
darwin-rebuild --help
```

date

Set or display the system date.

More information: https://keith.github.io/xcode-man-pages/date.1.html.

- Display the current date using the default locale's format:

```
date +%c
```

- Display the current date in UTC and ISO 8601 format:

```
date -u +%Y-%m-%dT%H:%M:%SZ
```

- Display the current date as a Unix timestamp (seconds since the Unix epoch):

```
date +%s
```

- Display a specific date (represented as a Unix timestamp) using the default format:

```
date -r {{1473305798}}
```

- Display a date relative to the current date using the default format:

```
date -v {{+1d}} -v {{-20m}}
```

dd

Convert and copy a file.

More information: https://keith.github.io/xcode-man-pages/dd.1.html.

- Make a bootable USB drive from an isohybrid file (such like `archlinux-xxx.iso`) and show the progress:

```
dd if={{path/to/file.iso}} of={{/dev/usb_drive}} status=progress
```

- Clone a drive to another drive with 4 MB block, ignore error and show the progress:

```
dd bs=4m conv=noerror if={{/dev/source_drive}} of={{/dev/dest_drive}} status=progress
```

- Generate a file with a specific number of random bytes by using kernel random driver:

```
dd bs={{100}} count={{1}} if=/dev/urandom of={{path/to/random_file}}
```

- Benchmark the write performance of a disk:

```
dd bs={{1024}} count={{1000000}} if=/dev/zero of={{path/to/1GB_file}}
```

- Create a system backup, save it into an IMG file (can be restored later by swapping `if` and `of`), and show the progress:

```
dd if={{/dev/drive_device}} of={{path/to/file.img}} status=progress
```

- Check the progress of an ongoing `dd` operation (run this command from another shell):

```
kill -USR1 $(pgrep ^dd)
```

defaults

Read and write macOS user configuration for applications.

More information: https://keith.github.io/xcode-man-pages/defaults.1.html.

- Read system defaults for an application option:

```
defaults read "{{application}}" "{{option}}"
```

- Read default values for an application option:

```
defaults read -app "{{application}}" "{{option}}"
```

- Search for a keyword in domain names, keys, and values:

```
defaults find "{{keyword}}"
```

- Write the default value of an application option:

```
defaults write "{{application}}" "{{option}}" {{-type}} {{value}}
```

- Speed up Mission Control animations:

```
defaults write com.apple.Dock expose-animation-duration -float 0.1
```

- Delete all defaults of an application:

```
defaults delete "{{application}}"
```

deleted

Keeps track of purgeable space and asks clients to purge when space is low.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/deleted.8.html.

- Start the daemon:

```
deleted
```

df

Display an overview of the filesystem disk space usage.

More information: https://keith.github.io/xcode-man-pages/df.1.html.

- Display all filesystems and their disk usage (using 512-byte units):

```
df
```

- Use [h]uman-readable units (based on powers of 1024) and display a grand total:

```
df -h -c
```

- Use [H]uman-readable units (based on powers of 1000):

```
df {{[-H|--si]}}
```

- Display the filesystem containing the specified file or directory:

```
df {{path/to/file_or_directory}}
```

- Include statistics on the number of free and used [i]nodes, including the filesystem t[Y]pes:

```
df -iY
```

- Use [k]ibibyte (1024 byte) units when showing size figures:

```
df -k
```

- Display information in a [P]ortable way:

```
df -P
```

dhcp6d

Stateless DHCPv6 server.

See also: **`InternetSharing`**.

It should not be invoked manually.

More information: https://www.manpagez.com/man/8/dhcp6d/.

- Start the daemon:

```
dhcp6d
```

- Use a custom configuration:

```
dhcp6d {{path/to/config_file}}
```

diskutil partitionDisk

Utility to manage partitions inside disks and volumes.

Part of **`diskutil`**.

APM is only supported for macOS, MBR is optimized for DOS, while GPT is compatible for most modern systems.

More information: https://keith.github.io/xcode-man-pages/diskutil.8.html.

- Reformat a volume using APM/MBR/GPT partitioning scheme, leaving no partitions inside (this will erase all data on the volume):

```
diskutil partitionDisk {{/dev/disk_device}} 0 {{APM|MBR|GPT}}
```

- Reformat a volume, then create a single partition using a specific filesystem filling up all free space:

```
diskutil partitionDisk {{/dev/disk_device}} 1 {{APM|MBR|GPT}}
{{partition_filesystem}} {{partition_name}}
```

- Reformat a volume, then create a single partition using a specific filesystem under specific size (e.g. `16G` for 16GB or `50%` to fill half of total volume size):

```
diskutil partitionDisk {{/dev/disk_device}} 1 {{APM|MBR|GPT}}
{{partition_filesystem}} {{partition_name}}
{{partition_size}}
```

- Reformat a volume, then create multiple partitions:

```
diskutil partitionDisk {{/dev/disk_device}}
{{number_of_partitions}} {{APM|MBR|GPT}}
{{partition_filesystem1}} {{partition_name1}}
{{partition_size1}} {{partition_filesystem2}}
{{partition_name2}} {{partition_size2}} ...
```

- List all supported file systems for partitioning:

```
diskutil listFilesystems
```

diskutil

Utility to manage local disks and volumes.

Some subcommands such as **`partitiondisk`** have their own usage documentation.

More information: https://keith.github.io/xcode-man-pages/diskutil.8.html.

- List all currently available disks, partitions and mounted volumes:

```
diskutil list
```

- Repair the filesystem data structures of a volume:

```
diskutil repairVolume {{/dev/disk}}
```

- Unmount a volume:

```
diskutil unmountDisk {{/dev/disk}}
```

- Eject a CD/DVD (unmount first):

```
diskutil eject {{/dev/diskX}}
```

distnoted

Start distributed notification services.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/distnoted.8.html.

- Start the daemon:

```
distnoted
```

ditto

Copy files and directories.

More information: https://keith.github.io/xcode-man-pages/ditto.1.html.

- Overwrite contents of destination directory with contents of source directory:

```
ditto {{path/to/source_directory}} {{path/to/
destination_directory}}
```

- Print a line to the Terminal window for every file that's being copied:

```
ditto -V {{path/to/source_directory}} {{path/to/
destination_directory}}
```

- Copy a given file or directory, while retaining the original file permissions:

```
ditto -rsrc {{path/to/source_directory}} {{path/to/
destination_directory}}
```

dmesg

Write the kernel messages to **`stdout`**.

More information: https://keith.github.io/xcode-man-pages/dmesg.8.html.

- Show kernel messages:

```
dmesg
```

- Show how much physical memory is available on this system:

```
dmesg | grep -i memory
```

- Show kernel messages 1 page at a time:

```
dmesg | less
```

dockutil

Manage macOS dock items.

More information: https://github.com/kcrawford/dockutil.

- Add an application to the end of the current user's dock:

```
dockutil --add {{path/to/application}}
```

- Replace one application with another in the current user's dock:

```
dockutil --add /{{path/to/application}} --replacing
'{{dock_item_label}}'
```

- Add a directory with view options and display it as a folder icon or stack:

```
dockutil --add /{{path/to/directory}} --view {{grid|fan|list|
auto}} --display {{folder|stack}}
```

- Add a URL dock item after another item:

```
dockutil --add {{vnc://example_server.local}} --label
'{{Example VNC}}' --after {{dock_item_label}}
```

- Remove an application from the dock given its dock label name:

```
dockutil --remove '{{dock_item_label}}'
```

- Add a spacer in a section after an application:

```
dockutil --add '' --type {{spacer|small-spacer|flex-spacer}}
--section {{apps}} --after {{dock_item_label}}
```

- Remove all spacer tiles:

```
dockutil --remove spacer-tiles
```

dot_clean

Merge ._* files with corresponding native files.

More information: https://keith.github.io/xcode-man-pages/dot_clean.1.html.

- Merge all `._*` files recursively:

```
dot_clean {{path/to/directory}}
```

- Don't recursively merge all `._*` in a directory (flat merge):

```
dot_clean -f {{path/to/directory}}
```

- Merge and delete all `._*` files:

```
dot_clean -m {{path/to/directory}}
```

- Only delete `._*` files if there's a matching native file:

```
dot_clean -n {{path/to/directory}}
```

- Follow symlinks:

```
dot_clean -s {{path/to/directory}}
```

- Print verbose output:

```
dot_clean -v {{path/to/directory}}
```

drutil

Interact with DVD burners.

More information: https://keith.github.io/xcode-man-pages/drutil.1.html.

- Eject a disk from the drive:

```
drutil eject
```

- Burn a directory as an ISO9660 filesystem onto a DVD. Don't verify and eject when complete:

```
drutil burn -noverify -eject -iso9660
```

dtrace

A simple interface to invoke the D language compiler, retrieve buffered trace, and print traced data from the DTrace kernel facility.

Generic front-end to DTrace facility, requiring root privileges.

More information: https://keith.github.io/xcode-man-pages/dtrace.1.html.

- Set target data model for a specific architecture:

```
dtrace -arch {{arch_name}}
```

- Claim [a]nonymous tracing state and display the traced data:

```
dtrace -a
```

- Set principal trace buffer size. Supported units are `k`, `m`, `g`, or `t`:

```
dtrace -b {{2g}}
```

- Compile the specified D Program [s]ource file:

```
dtrace -s {{D_script}}
```

- Run the specified [c]ommand and exit upon its completion:

```
dtrace -c {{command}}
```

- Specify [f]unction name to trace or list (-l option). The corresponding argument can include any of the probe description forms like `provider:module:function`, `module:function` or `function`:

```
dtrace -f {{function}}
```

- Grad the specified [p]rocess ID, cache its symbol table, and exit upon completion:

```
dtrace -p {{pid}}
```

- Combine different options for tracing function in a process:

```
dtrace -a -b {{buffer_size}} -f {{function}} -p {{pid}}
```

du

Disk usage: estimate and summarize file and directory space usage.

More information: https://keith.github.io/xcode-man-pages/du.1.html.

- List the sizes of a directory and any subdirectories, in the given unit (KiB/MiB/GiB):

```
du -{{k|m|g}} {{path/to/directory}}
```

- List the sizes of a directory and any subdirectories, in human-readable form (i.e. auto-selecting the appropriate unit for each size):

```
du -h {{path/to/directory}}
```

- Show the size of a single directory, in human-readable units:

```
du -sh {{path/to/directory}}
```

- List the human-readable sizes of a directory and of all the files and directories within it:

```
du -ah {{path/to/directory}}
```

- List the human-readable sizes of a directory and any subdirectories, up to `n` levels deep:

```
du -h -d {{n}} {{path/to/directory}}
```

- List the human-readable size of all `.jpg` files in subdirectories of the current directory, and show a cumulative total at the end:

```
du -ch */*.jpg
```

duti

Set default applications for document types and URL schemes on macOS.

See also: **`osascript`**.

More information: https://github.com/moretension/duti.

- Set Safari as the default handler for HTML documents:

```
duti -s {{com.apple.Safari}} {{public.html}} all
```

- Set VLC as the default viewer for files with `.m4v` extensions:

```
duti -s {{org.videolan.vlc}} {{m4v}} viewer
```

- Set Finder as the default handler for the ftp:// URL scheme:

```
duti -s {{com.apple.Finder}} "{{ftp}}"
```

- Display information about the default application for a given extension:

```
duti -x {{ext}}
```

- Display the default handler for a given UTI:

```
duti -d {{uti}}
```

- Display all handlers of a given UTI:

```
duti -l {{uti}}
```

emond

Event Monitor service that accepts events from various services, runs them through a simple rules engine, and takes action.

The actions can run commands, send email, or SMS messages.

More information: https://manpagez.com/man/8/emond/.

- Start the daemon:

```
emond
```

- Specify rules for emond to process by giving a path to a file or directory:

```
emond -r {{path/to/file_or_directory}}
```

- Use a specific configuration file:

```
emond -c {{path/to/config_file}}
```

fdesetup

Set and retrieve FileVault related information.

More information: https://keith.github.io/xcode-man-pages/fdesetup.8.html.

- List current FileVault enabled users:

```
sudo fdesetup list
```

- Get current FileVault status:

```
fdesetup status
```

- Add FileVault enabled user:

```
sudo fdesetup add -usertoadd {{user1}}
```

- Enable FileVault:

```
sudo fdesetup enable
```

- Disable FileVault:

```
sudo fdesetup disable
```

filecoordinationd

Coordinates access to files by multiple processes (**`NSFileCoordinator`**, **`NSFilePresenter`**).

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/filecoordinationd.8.html.

- Start the daemon:

```
filecoordinationd
```

fileicon

Manage custom file and folder icons.

More information: https://github.com/mklement0/fileicon.

- Set a custom icon for a specific file or directory:

```
fileicon set {{path/to/file_or_directory}} {{path/to/icon_file.png}}
```

- Remove a custom icon from a specific file or directory:

```
fileicon rm {{path/to/file_or_directory}}
```

- Save the custom icon of a file or directory as a `.icns` file into the current directory:

```
fileicon get {{path/to/file_or_directory}}
```

- Test if a specific file or directory has a custom icon:

```
fileicon test {{path/to/file_or_directory}}
```

fontd

Make fonts available to the system.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/fontd.8.html.

- Start the daemon:

```
fontd
```

fsck

Check the integrity of a filesystem or repair it. The filesystem should be unmounted at the time the command is run.

It is a wrapper that calls **`fsck_hfs`**, **`fsck_apfs`**, **`fsck_msdos`**, **`fsck_exfat`**, and **`fsck_udf`** as needed.

More information: https://keith.github.io/xcode-man-pages/fsck.8.html.

- Check filesystem `/dev/sdX`, reporting any damaged blocks:

  ```
  fsck {{/dev/sdX}}
  ```

- Check filesystem `/dev/sdX` only if it is clean, reporting any damaged blocks and interactively letting the user choose to repair each one:

  ```
  fsck -f {{/dev/sdX}}
  ```

- Check filesystem `/dev/sdX` only if it is clean, reporting any damaged blocks and automatically repairing them:

  ```
  fsck -fy {{/dev/sdX}}
  ```

- Check filesystem `/dev/sdX`, reporting whether it has been cleanly unmounted:

  ```
  fsck -q {{/dev/sdX}}
  ```

ftxdiff

Compare differences between two fonts.

More information: https://developer.apple.com/fonts.

- Output differences to a specific text file:

```
ftxdiff --output {{path/to/fontdiff_file.txt}} {{path/to/font_file1.ttc}} {{path/to/font_file2.ttc}}
```

- Include glyph names in output:

```
ftxdiff --include-glyph-names
```

- Include unicode names in output:

```
ftxdiff --include-unicode-names
```

fuser

Display process IDs currently using files.

More information: https://keith.github.io/xcode-man-pages/fuser.1.html.

- Show PIDs of processes accessing a file or directory:

```
fuser {{path/to/file_or_directory}}
```

- Show PIDs and usernames of processes accessing a file or directory:

```
fuser -u {{path/to/file_or_directory}}
```

g[

This command is an alias of GNU `[`.

- View documentation for the original command:

```
tldr [
```

gb2sum

This command is an alias of GNU **`b2sum`**.

- View documentation for the original command:

```
tldr b2sum
```

gbase32

This command is an alias of GNU **base32**.

- View documentation for the original command:

```
tldr base32
```

gbase64

This command is an alias of GNU **base64**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common base64
```

gbasename

This command is an alias of GNU **`basename`**.

- View documentation for the original command:

```
tldr basename
```

gbasenc

This command is an alias of GNU **`basenc`**.

- View documentation for the original command:

```
tldr basenc
```

gcat

This command is an alias of GNU **cat**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux cat
```

gchcon

This command is an alias of GNU **chcon**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux chcon
```

gchgrp

This command is an alias of GNU **`chgrp`**.

- View documentation for the original command:

  ```
  tldr chgrp
  ```

gchmod

This command is an alias of GNU **chmod**.

- View documentation for the original command:

  ```
  tldr chmod
  ```

gchown

This command is an alias of GNU **chown**.

- View documentation for the original command:

  ```
  tldr chown
  ```

gchroot

This command is an alias of GNU **`chroot`**.

- View documentation for the original command:

  ```
  tldr chroot
  ```

gcksum

This command is an alias of GNU **`cksum`**.

- View documentation for the original command:

```
tldr cksum
```

gcomm

This command is an alias of GNU **comm**.

- View documentation for the original command:

```
tldr comm
```

gcp

This command is an alias of GNU **cp**.

- View documentation for the original command:

```
tldr cp
```

gcrane completion

Generate the autocompletion script for gcrane for the specified shell.

The available shells are Bash, fish, PowerShell, and Zsh.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/gcrane/README.md.

- Generate the autocompletion script for your shell:

```
gcrane completion {{shell_name}}
```

- Disable completion descriptions:

```
gcrane completion {{shell_name}} --no-descriptions
```

- Load completions in your current shell session (Bash/Zsh):

```
source <(gcrane completion bash/zsh)
```

- Load completions in your current shell session (fish):

```
gcrane completion fish | source
```

- Load completions for every new session (Bash):

```
gcrane completion bash > $(brew --prefix)/etc/
bash_completion.d/gcrane
```

- Load completions for every new session (Zsh):

```
gcrane completion zsh > $(brew --prefix)/share/zsh/site-
functions/_gcrane
```

- Load completions for every new session (fish):

```
gcrane completion fish > ~/.config/fish/completions/
gcrane.fish
```

- Display help:

```
gcrane completion {{shell_name}} {{[-h|--help]}}
```

gcsplit

This command is an alias of GNU **`csplit`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux csplit
```

gcut

This command is an alias of GNU **`cut`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common cut
```

gdate

This command is an alias of GNU **`date`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common date
```

gdd

This command is an alias of GNU **dd**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux dd
```

gdf

This command is an alias of GNU **df**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux df
```

gdir

This command is an alias of GNU **`dir`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux dir
```

gdircolors

This command is an alias of GNU **`dircolors`**.

- View documentation for the original command:

  ```
  tldr dircolors
  ```

gdirname

This command is an alias of GNU **`dirname`**.

- View documentation for the original command:

  ```
  tldr dirname
  ```

gdnsdomainname

This command is an alias of GNU **dnsdomainname**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux dnsdomainname
```

gecho

This command is an alias of GNU **echo**.

- View documentation for the original command:

  ```
  tldr echo
  ```

ged

This command is an alias of GNU **ed**.

- View documentation for the original command:

```
tldr ed
```

gegrep

This command is an alias of GNU **`egrep`**.

- View documentation for the original command:

  ```
  tldr egrep
  ```

genv

This command is an alias of GNU **env**.

- View documentation for the original command:

  ```
  tldr env
  ```

GetFileInfo

Get information about a file in an HFS+ directory.

More information: https://keith.github.io/xcode-man-pages/GetFileInfo.1.html.

- Display information about a given file:

```
GetFileInfo {{path/to/file}}
```

- Display the [d]ate and time a given file was created:

```
GetFileInfo -d {{path/to/file}}
```

- Display the date and time a given file was last [m]odified:

```
GetFileInfo -m {{path/to/file}}
```

- Display the [c]reator of a given file:

```
GetFileInfo -c {{path/to/file}}
```

gexpand

This command is an alias of GNU **expand**.

- View documentation for the original command:

  `tldr expand`

gexpr

This command is an alias of GNU **expr**.

- View documentation for the original command:

```
tldr expr
```

gfactor

This command is an alias of GNU **`factor`**.

- View documentation for the original command:

```
tldr factor
```

gfalse

This command is an alias of GNU **`false`**.

- View documentation for the original command:

  `tldr false`

gfgrep

This command is an alias of GNU **`fgrep`**.

- View documentation for the original command:

```
tldr fgrep
```

gfind

This command is an alias of GNU **`find`**.

- View documentation for the original command:

  ```
  tldr find
  ```

gfmt

This command is an alias of GNU **fmt**.

- View documentation for the original command:

```
tldr fmt
```

gfold

This command is an alias of GNU **`fold`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux fold
```

gftp

This command is an alias of GNU **`ftp`**.

- View documentation for the original command:

```
tldr ftp
```

ggrep

This command is an alias of GNU **`grep`**.

- View documentation for the original command:

```
tldr grep
```

ggroups

This command is an alias of GNU **`groups`**.

- View documentation for the original command:

  ```
  tldr groups
  ```

ghead

This command is an alias of GNU **head**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux head
```

ghostid

This command is an alias of GNU **hostid**.

- View documentation for the original command:

```
tldr hostid
```

ghostname

This command is an alias of GNU **hostname**.

- View documentation for the original command:

```
tldr hostname
```

gid

This command is an alias of GNU **id**.

- View documentation for the original command:

```
tldr id
```

gifconfig

This command is an alias of GNU **`ifconfig`**.

- View documentation for the original command:

  ```
  tldr ifconfig
  ```

gindent

This command is an alias of GNU **`indent`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common indent
```

ginstall

This command is an alias of GNU **`install`**.

- View documentation for the original command:

  `tldr install`

gjoin

This command is an alias of GNU **`join`**.

- View documentation for the original command:

```
tldr join
```

gkill

This command is an alias of GNU **`kill`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux kill
```

glibtool

This command is an alias of GNU **`libtool`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux libtool
```

glibtoolize

This command is an alias of GNU **`libtoolize`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux libtoolize
```

glink

This command is an alias of GNU **`link`**.

- View documentation for the original command:

  ```
  tldr link
  ```

gln

This command is an alias of GNU **`ln`**.

- View documentation for the original command:

  ```
  tldr ln
  ```

glocate

This command is an alias of GNU **`locate`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux locate
```

glogger

This command is an alias of GNU **`logger`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux logger
```

glogname

This command is an alias of GNU **`logname`**.

- View documentation for the original command:

  ```
  tldr logname
  ```

gls

This command is an alias of GNU **`ls`**.

- View documentation for the original command:

  ```
  tldr ls
  ```

gmake

This command is an alias of GNU **make**.

- View documentation for the original command:

```
tldr make
```

gmd5sum

This command is an alias of GNU **`md5sum`**.

- View documentation for the original command:

  ```
  tldr md5sum
  ```

gmkdir

This command is an alias of GNU **`mkdir`**.

- View documentation for the original command:

```
tldr mkdir
```

gmkfifo

This command is an alias of GNU **`mkfifo`**.

- View documentation for the original command:

  ```
  tldr mkfifo
  ```

gmknod

This command is an alias of GNU **mknod**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux mknod
```

gmktemp

This command is an alias of GNU **mktemp**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux mktemp
```

gmv

This command is an alias of GNU **mv**.

- View documentation for the original command:

```
tldr mv
```

gnice

This command is an alias of GNU **`nice`**.

- View documentation for the original command:

  ```
  tldr nice
  ```

gnl

This command is an alias of GNU **`nl`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux nl
```

gnohup

This command is an alias of GNU **nohup**.

- View documentation for the original command:

```
tldr nohup
```

gnproc

This command is an alias of GNU **`nproc`**.

- View documentation for the original command:

  ```
  tldr nproc
  ```

gnumfmt

This command is an alias of GNU **`numfmt`**.

- View documentation for the original command:

  `tldr numfmt`

god

This command is an alias of GNU **od**.

- View documentation for the original command:

```
tldr od
```

goku

Manage Karabiner configuration.

More information: https://github.com/yqrashawn/GokuRakuJoudo.

- Generate `karabiner.json` using the default configuration:

```
goku
```

- Generate `karabiner.json` using the specific `config.edn` file:

```
goku --config {{path/to/config.edn}}
```

- Dry run the new configuration into `stdout` instead of updating `karabiner.json`:

```
goku --dry-run
```

- Dry run the whole configuration into `stdout` instead of updating `karabiner.json`:

```
goku --dry-run-all
```

- Display help:

```
goku --help
```

- Display version:

```
goku --version
```

gpaste

This command is an alias of GNU **`paste`**.

- View documentation for the original command:

```
tldr paste
```

gpathchk

This command is an alias of GNU **pathchk**.

- View documentation for the original command:

```
tldr pathchk
```

gping

This command is an alias of GNU **`ping`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common ping
```

gping6

This command is an alias of GNU **`ping6`**.

- View documentation for the original command:

```
tldr ping6
```

gpinky

This command is an alias of GNU **`pinky`**.

- View documentation for the original command:

```
tldr pinky
```

gpr

This command is an alias of GNU **pr**.

- View documentation for the original command:

```
tldr pr
```

gprintenv

This command is an alias of GNU **`printenv`**.

- View documentation for the original command:

  ```
  tldr printenv
  ```

gprintf

This command is an alias of GNU **`printf`**.

- View documentation for the original command:

  ```
  tldr printf
  ```

gptx

This command is an alias of GNU **ptx**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux ptx
```

gpwd

This command is an alias of GNU **pwd**.

- View documentation for the original command:

```
tldr pwd
```

grcp

This command is an alias of GNU **rcp**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux rcp
```

greadlink

This command is an alias of GNU **`readlink`**.

- View documentation for the original command:

```
tldr readlink
```

grealpath

This command is an alias of GNU **`realpath`**.

- View documentation for the original command:

```
tldr realpath
```

grexec

This command is an alias of GNU **rexec**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux rexec
```

grlogin

This command is an alias of GNU **rlogin**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux rlogin
```

grm

This command is an alias of GNU **rm**.

- View documentation for the original command:

  ```
  tldr rm
  ```

grmdir

This command is an alias of GNU **`rmdir`**.

- View documentation for the original command:

```
tldr rmdir
```

grsh

This command is an alias of GNU **rsh**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux rsh
```

gruncon

This command is an alias of GNU **runcon**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux runcon
```

gsed

This command is an alias of GNU **sed**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux sed
```

gseq

This command is an alias of GNU **seq**.

- View documentation for the original command:

```
tldr seq
```

gsha1sum

This command is an alias of GNU **`sha1sum`**.

- View documentation for the original command:

  ```
  tldr sha1sum
  ```

gsha224sum

This command is an alias of GNU **`sha224sum`**.

- View documentation for the original command:

```
tldr sha224sum
```

gsha256sum

This command is an alias of GNU **`sha256sum`**.

- View documentation for the original command:

```
tldr sha256sum
```

gsha384sum

This command is an alias of GNU **`sha384sum`**.

- View documentation for the original command:

```
tldr sha384sum
```

gsha512sum

This command is an alias of GNU **`sha512sum`**.

- View documentation for the original command:

```
tldr sha512sum
```

gshred

This command is an alias of GNU **`shred`**.

- View documentation for the original command:

```
tldr shred
```

gshuf

This command is an alias of GNU **`shuf`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} coomon shuf
```

gsleep

This command is an alias of GNU **`sleep`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux sleep
```

gsort

This command is an alias of GNU **`sort`**.

- View documentation for the original command:

  ```
  tldr sort
  ```

gsplit

This command is an alias of GNU **`split`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common split
```

gstat

This command is an alias of GNU **`stat`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common stat
```

gstdbuf

This command is an alias of GNU **`stdbuf`**.

- View documentation for the original command:

```
tldr stdbuf
```

gstty

This command is an alias of GNU **`stty`**.

- View documentation for the original command:

```
tldr stty
```

gsum

This command is an alias of GNU **`sum`**.

- View documentation for the original command:

```
tldr sum
```

gsync

This command is an alias of GNU **sync**.

- View documentation for the original command:

```
tldr sync
```

gtac

This command is an alias of GNU **tac**.

- View documentation for the original command:

```
tldr tac
```

gtail

This command is an alias of GNU **`tail`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common tail
```

gtalk

This command is an alias of GNU **`talk`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux talk
```

gtar

This command is an alias of GNU **tar**.

- View documentation for the original command:

```
tldr tar
```

gtee

This command is an alias of GNU **tee**.

- View documentation for the original command:

```
tldr tee
```

gtelnet

This command is an alias of GNU **`telnet`**.

- View documentation for the original command:

  ```
  tldr telnet
  ```

gtest

This command is an alias of GNU **`test`**.

- View documentation for the original command:

```
tldr test
```

gtftp

This command is an alias of GNU **`tftp`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux tftp
```

gtime

This command is an alias of GNU **`time`**.

- View documentation for the original command:

```
tldr time
```

gtimeout

This command is an alias of GNU **`timeout`**.

- View documentation for the original command:

```
tldr timeout
```

gtouch

This command is an alias of GNU **`touch`**.

- View documentation for the original command:

  ```
  tldr touch
  ```

gtr

This command is an alias of GNU **tr**.

- View documentation for the original command:

```
tldr tr
```

gtraceroute

This command is an alias of GNU **`traceroute`**.

- View documentation for the original command:

  ```
  tldr traceroute
  ```

gtrue

This command is an alias of GNU **`true`**.

- View documentation for the original command:

```
tldr true
```

gtruncate

This command is an alias of GNU **`truncate`**.

- View documentation for the original command:

```
tldr truncate
```

gtsort

This command is an alias of GNU **`tsort`**.

- View documentation for the original command:

```
tldr tsort
```

gtty

This command is an alias of GNU **`tty`**.

- View documentation for the original command:

```
tldr tty
```

guname

This command is an alias of GNU **uname**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common uname
```

gunexpand

This command is an alias of GNU **unexpand**.

- View documentation for the original command:

```
tldr unexpand
```

guniq

This command is an alias of GNU **`uniq`**.

- View documentation for the original command:

```
tldr uniq
```

gunits

This command is an alias of GNU **`units`**.

- View documentation for the original command:

```
tldr units
```

gunlink

This command is an alias of GNU **`unlink`**.

- View documentation for the original command:

  ```
  tldr unlink
  ```

gupdatedb

This command is an alias of GNU **updatedb**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} linux updatedb
```

guptime

This command is an alias of GNU **uptime**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common uptime
```

gusers

This command is an alias of GNU **`users`**.

- View documentation for the original command:

```
tldr users
```

gvdir

This command is an alias of GNU **vdir**.

- View documentation for the original command:

```
tldr vdir
```

gwc

This command is an alias of GNU **`wc`**.

- View documentation for the original command:

```
tldr {{[-p|--platform]}} common wc
```

gwhich

This command is an alias of GNU **`which`**.

- View documentation for the original command:

  ```
  tldr which
  ```

gwho

This command is an alias of GNU **who**.

- View documentation for the original command:

```
tldr who
```

gwhoami

This command is an alias of GNU **`whoami`**.

- View documentation for the original command:

  ```
  tldr whoami
  ```

gwhois

This command is an alias of GNU **`whois`**.

- View documentation for the original command:

```
tldr whois
```

gxargs

This command is an alias of GNU **`xargs`**.

- View documentation for the original command:

```
tldr xargs
```

gyes

This command is an alias of GNU **yes**.

- View documentation for the original command:

```
tldr yes
```

hdiutil

Utility to create and manage disk images.

More information: https://keith.github.io/xcode-man-pages/hdiutil.1.html.

- Mount an image:

```
hdiutil attach {{path/to/image_file}}
```

- Unmount an image:

```
hdiutil detach /Volumes/{{volume_file}}
```

- List mounted images:

```
hdiutil info
```

- Create an ISO image from the contents of a directory:

```
hdiutil makehybrid -o {{path/to/output_file}} {{path/to/
directory}}
```

head

Output the first part of files.

More information: https://keith.github.io/xcode-man-pages/head.1.html.

- Output the first 10 lines of a file:

```
head {{path/to/file}}
```

- Output the first 5 lines of multiple files:

```
head {{[-5|--lines 5]}} {{path/to/file1 path/to/file2 ...}}
```

- Output the first n lines of a file:

```
head {{[-n|--lines]}} {{n}} {{path/to/file}}
```

- Output the first n bytes of a file:

```
head {{[-c|--bytes]}} {{n}} {{path/to/file}}
```

herd list

List available commands in the Herd PHP platform.

See also: **herd**.

More information: https://herd.laravel.com/docs/macos/advanced-usage/herd-cli.

- List all available commands:

```
herd list
```

- List all available commands in a specific namespace:

```
herd list {{namespace}}
```

- List all commands in raw format (useful for embedding a command runner):

```
herd list --raw
```

- Display the list in a specific output format:

```
herd list --format {{txt|xml|json|md}}
```

- List all commands without describing their arguments:

```
herd list --short
```

herd

An official Laravel PHP development environment for macOS.

More information: https://herd.laravel.com/docs/macos/advanced-usage/herd-cli.

- Start the Herd services:

```
herd start
```

- Stop the Herd services:

```
herd stop
```

- Restart the Herd services:

```
herd restart
```

- Link the current working directory to Herd:

```
herd link
```

- Open the site for the current directory in the browser:

```
herd open
```

- List all available commands:

```
herd list
```

hidd

HID library userland daemon.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/hidd.8.html.

- Start the daemon:

```
hidd
```

icalBuddy

Command-line utility for printing events and tasks from the macOS calendar database.

More information: https://hasseg.org/icalBuddy/.

- Show events later today:

```
icalBuddy --includeOnlyEventsFromNowOn eventsToday
```

- Show uncompleted tasks:

```
icalBuddy uncompletedTasks
```

- Show a formatted list separated by calendar for all events today:

```
icalBuddy --formatOutput --separateByCalendar eventsToday
```

- Show tasks for a specified number of days:

```
icalBuddy --includeOnlyEventsFromNowOn "tasksDueBefore:today+{{number_of_days}}"
```

- Show events in a time range:

```
icalBuddy eventsFrom:{{start_date}} to:{{end_date}}
```

indent

Change the appearance of a C/C++ program by inserting or deleting whitespace.

More information: https://keith.github.io/xcode-man-pages/indent.1.html.

- Format C/C++ source according to the Berkeley style:

```
indent {{path/to/source_file.c}} {{path/to/indented_file.c}} -nbad -nbap -bc -br -c33 -cd33 -cdb -ce -ci4 -cli0 -di16 -fc1 -fcb -i4 -ip -l75 -lp -npcs -nprs -psl -sc -nsob -ts8
```

- Format C/C++ source according to the style of Kernighan & Ritchie (K&R):

```
indent {{path/to/source_file.c}} {{path/to/indented_file.c}} -nbad -bap -nbc -br -c33 -cd33 -ncdb -ce -ci4 -cli0 -cs -d0 -di1 -nfc1 -nfcb -i4 -nip -l75 -lp -npcs -nprs -npsl -nsc -nsob
```

InternetSharing

Set up Internet Sharing.

It should not be invoked manually.

More information: https://www.manpagez.com/man/8/InternetSharing/.

- Start the daemon:

```
InternetSharing
```

iostat

Report statistics for devices.

More information: https://ss64.com/mac/iostat.html.

- Display snapshot device statistics (kilobytes per transfer, transfers per second, megabytes per second), CPU statistics (percentages of time spent in user mode, system mode, and idle mode), and system load averages (for the past 1, 5, and 15 min):

```
iostat
```

- Display only device statistics:

```
iostat -d
```

- Display incremental reports of CPU and disk statistics every 2 seconds:

```
iostat 2
```

- Display statistics for the first disk every second indefinitely:

```
iostat -w 1 disk0
```

- Display statistics for the second disk every 3 seconds, 10 times:

```
iostat -w 3 -c 10 disk1
```

- Display using old-style `iostat` display. Shows sectors transferred per second, transfers per second, average milliseconds per transaction, and CPU statistics + load averages from default-style display:

```
iostat -o
```

- Display total device statistics (KB/t: kilobytes per transfer as before, xfrs: total number of transfers, MB: total number of megabytes transferred):

```
iostat -I
```

ipconfig

View and control IP configuration state.

More information: https://keith.github.io/xcode-man-pages/ipconfig.8.html.

- List all network interfaces:

```
ipconfig getiflist
```

- Show the IP address of an interface:

```
ipconfig getifaddr {{interface_name}}
```

istats

Show various statistics such as the CPU temperature, fan speeds and battery status.

More information: https://github.com/Chris911/iStats.

- Show all the stats:

```
istats
```

- Show all CPU stats:

```
istats cpu
```

- Show all fan stats:

```
istats fan
```

- Scan and print temperatures:

```
istats scan
```

java_home

Return a value for $JAVA_HOME or execute command using this variable.

More information: https://www.unix.com/man-page/osx/1/java_home.

- List JVMs based on a specific version:

```
java_home --version {{1.5+}}
```

- List JVMs based on a specific [arch]itecture:

```
java_home --arch {{i386}}
```

- List JVMs based on a specific tasks (defaults to `CommandLine`):

```
java_home --datamodel {{Applets|WebStart|BundledApp|JNI|
CommandLine}}
```

- List JVMs in a XML format:

```
java_home --xml
```

- Display help:

```
java_home --help
```

kmutil

Utility for managing kernel extensions (kexts) and kext collections on disk.

More information: https://keith.github.io/xcode-man-pages/kmutil.8.html.

- Find kexts available on the operating system:

```
kmutil find
```

- Display logging information about the Kernel Management sub-system:

```
kmutil log
```

- Inspect and display a kext collection's contents according to the options provided:

```
kmutil inspect {{options}}
```

- Check the consistency of kext collections against each other:

```
kmutil check
```

- Dump kernelmanagerd state for debugging:

```
sudo kmutil dumpstate
```

- Load one or more extensions based on the bundle specified at this path in the results:

```
kmutil load --bundle-path {{path/to/extension.kext}}
```

launchctl

Control Apple's **`launchd`** manager for launch daemons (system-wide services) and launch agents (per-user programs).

`launchd` loads XML-based **`*.plist`** files placed in the appropriate locations, and runs the corresponding commands according to their defined schedule.

More information: https://keith.github.io/xcode-man-pages/launchctl.1.html.

- Activate a user-specific agent to be loaded into `launchd` whenever the user logs in:

```
launchctl load ~/Library/LaunchAgents/{{my_script}}.plist
```

- Activate an agent which requires root privileges to run and/or should be loaded whenever any user logs in (note the absence of `~` in the path):

```
sudo launchctl load /Library/LaunchAgents/
{{root_script}}.plist
```

- Activate a system-wide daemon to be loaded whenever the system boots up (even if no user logs in):

```
sudo launchctl load /Library/LaunchDaemons/
{{system_daemon}}.plist
```

- Show all loaded agents/daemons, with the PID if the process they specify is currently running, and the exit code returned the last time they ran:

```
launchctl list
```

- Unload a currently loaded agent, e.g. to make changes (Note: The plist file is automatically loaded into `launchd` after a reboot and/or logging in):

```
launchctl unload ~/Library/LaunchAgents/{{my_script}}.plist
```

- Manually run a known (loaded) agent/daemon, even if it is not the right time (Note: This command uses the agent's label, rather than the filename):

```
launchctl start {{script_file}}
```

- Manually kill the process associated with a known agent/daemon, if it is running:

```
launchctl stop {{script_file}}
```

launchd

Manages processes, both for the system and users.

You cannot invoke launchd manually, use launchctl to interact with it.

More information: https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/Introduction.html.

- Run init:

```
/sbin/launchd
```

- View documentation for interacting with launchd using launchctl:

```
tldr launchctl
```

lipo

Handle Mach-O Universal Binaries.

More information: https://keith.github.io/xcode-man-pages/lipo.1.html.

- Create a universal file from two single-architecture files:

```
lipo {{path/to/binary_file.x86_64}} {{path/to/
binary_file.arm64e}} -create -output {{path/to/binary_file}}
```

- List all architectures contained in a universal file:

```
lipo {{path/to/binary_file}} -archs
```

- Display detailed information about a universal file:

```
lipo {{path/to/binary_file}} -detailed_info
```

- Extract a single-architecture file from a universal file:

```
lipo {{path/to/binary_file}} -thin {{arm64e}} -output {{path/
to/binary_file.arm64e}}
```

lldb

The LLVM Low-Level Debugger.

More information: https://lldb.llvm.org/man/lldb.html.

- Debug an executable:

```
lldb "{{executable}}"
```

- Attach `lldb` to a running process with a given PID:

```
lldb -p {{pid}}
```

- Wait for a new process to launch with a given name, and attach to it:

```
lldb -w -n "{{process_name}}"
```

llvm-lipo

This command is an alias of **`lipo`**.

- View documentation for the original command:

  ```
  tldr lipo
  ```

locate

Find filenames quickly.

More information: https://keith.github.io/xcode-man-pages/locate.1.html.

- Look for pattern in the database. Note: The database is recomputed periodically (usually weekly or daily):

```
locate "{{pattern}}"
```

- Look for a pattern case-insensitively:

```
locate -i "{{pattern}}"
```

- Look for a file by its exact filename (a pattern containing no globbing characters is interpreted as `*pattern*`):

```
locate "*/{{filename}}"
```

- Recompute the database. You need to do it if you want to find recently added files:

```
sudo /usr/libexec/locate.updatedb
```

log

View, export, and configure logging systems.

More information: https://keith.github.io/xcode-man-pages/log.1.html.

- Stream live system logs:

```
log stream
```

- Stream logs sent to `syslog` from the process with a specific PID:

```
log stream --process {{process_id}}
```

- Show logs sent to syslog from a process with a specific name:

```
log show --predicate "process == '{{process_name}}'"
```

- Export all logs to disk for the past hour:

```
sudo log collect --last {{1h}} --output {{path/to/
file.logarchive}}
```

look

Display lines beginning with a prefix in a sorted file.

See also: **`grep`**, **`sort`**.

More information: https://keith.github.io/xcode-man-pages/look.1.html.

- Search for lines beginning with a specific prefix in a specific file:

```
look {{prefix}} {{path/to/file}}
```

- Case-insensitively search only on alphanumeric characters:

```
look {{[-f|--ignore-case]}} {{[-d|--alphanum]}} {{prefix}} {{path/to/file}}
```

- Specify a string termination character (space by default):

```
look {{[-t|--terminate]}} {{,}}
```

- Search in `/usr/share/dict/words` (`--ignore-case` and `--alphanum` are assumed):

```
look {{prefix}}
```

lpstat

Display status information about the current classes, jobs, and printers.

More information: https://keith.github.io/xcode-man-pages/lpstat.1.html.

- Show a long listing of printers, classes, and jobs:

```
lpstat -l
```

- Force encryption when connecting to the CUPS server:

```
lpstat -E
```

- Show the ranking of print jobs:

```
lpstat -R
```

- Show whether or not the CUPS server is running:

```
lpstat -r
```

- Show all status information:

```
lpstat -t
```

lsappinfo

Control and query CoreApplicationServices about the app state on the system.

More information: https://keith.github.io/xcode-man-pages/lsappinfo.8.html.

- List all running applications with their details:

```
lsappinfo list
```

- Show the front application:

```
lsappinfo front
```

- Show the information for a specific application:

```
lsappinfo info {{com.apple.calculator}}
```

m

Swiss Army Knife for macOS.

More information: https://github.com/rgcr/m-cli.

- Get the battery status:

```
m battery status
```

- Turn off Bluetooth:

```
m bluetooth off
```

- List available filesystems for formatting:

```
m disk filesystems
```

- Enable Dock's auto hide feature:

```
m dock autohide YES
```

- Disable the firewall:

```
m firewall disable
```

mac-cleanup

A modern macOS cleanup tool to remove caches and junk.

More information: https://github.com/mac-cleanup/mac-cleanup-py.

- Start the cleanup process:

```
mac-cleanup
```

- Open the module configuration screen:

```
mac-cleanup {{[-c|--configure]}}
```

- Perform a dry-run, showing what will be removed without actually deleting it:

```
mac-cleanup {{[-n|--dry-run]}}
```

- Specify the directory with custom cleanup modules:

```
mac-cleanup {{[-p|--custom-path]}} {{path/to/directory}}
```

- Automatically acknowledge all warnings and continue with force:

```
mac-cleanup {{[-f|--force]}}
```

machine

Print machine type.

More information: https://keith.github.io/xcode-man-pages/machine.1.html.

- Print CPU architecture:

```
machine
```

mas

Command-line interface for the Mac App Store.

More information: https://github.com/mas-cli/mas.

- Sign into the Mac App Store for the first time:

```
mas signin "{{user@example.com}}"
```

- Show all installed applications and their product identifiers:

```
mas list
```

- Search for an application, displaying the price alongside the results:

```
mas search "{{application}}" --price
```

- Install or update an application using exact numeric id:

```
mas install {{numeric_product_id}}
```

- Install the first application that would be returned by the respective search:

```
mas lucky "{{search_term}}"
```

- List all outdated apps with pending updates:

```
mas outdated
```

- Install all pending updates:

```
mas upgrade
```

- Upgrade a specific application:

```
mas upgrade "{{numeric_product_id}}"
```

mate

General-purpose text editor for macOS.

More information: https://macromates.com/.

- Start TextMate:

```
mate
```

- Open specific files:

```
mate {{path/to/file1 path/to/file2 ...}}
```

- Specify the filetype of a file:

```
mate --type {{filetype}} {{path/to/file}}
```

- Open and wait until finished editing a specific file:

```
mate --wait {{path/to/file}}
```

- Open a file with the cursor at a specific line and column:

```
mate --line {{line_number}}:{{column_number}} {{path/to/
file}}
```

md5

Calculate MD5 cryptographic checksums.

More information: https://keith.github.io/xcode-man-pages/md5.1.html.

- Calculate the MD5 checksum for a file:

```
md5 {{path/to/file}}
```

- Calculate MD5 checksums for multiple files:

```
md5 {{path/to/file1 path/to/file2 ...}}
```

- Output only the md5 checksum (no filename):

```
md5 -q {{path/to/file}}
```

- Print a checksum of the given string:

```
md5 -s "{{string}}"
```

mdfind

List files matching a query.

More information: https://keith.github.io/xcode-man-pages/mdfind.1.html.

- Find a file by its name:

```
mdfind -name {{file}}
```

- Find a file by its content:

```
mdfind "{{query}}"
```

- Find a file containing a string, in a given directory:

```
mdfind -onlyin {{directory}} "{{query}}"
```

mdls

Display the metadata attributes for a file.

More information: https://keith.github.io/xcode-man-pages/mdls.1.html.

- Display the list of metadata attributes for file:

```
mdls {{path/to/file}}
```

- Display a specific metadata attribute:

```
mdls -name {{attribute}} {{path/to/file}}
```

mdutil

Manage the metadata stores used by Spotlight for indexing.

More information: https://keith.github.io/xcode-man-pages/mdutil.1.html.

- Show the indexing status of the startup volume:

```
mdutil -s {{/}}
```

- Turn on/off the Spotlight indexing for a given volume:

```
mdutil -i {{on|off}} {{path/to/volume}}
```

- Turn on/off indexing for all volumes:

```
mdutil -a -i {{on|off}}
```

- Erase the metadata stores and restart the indexing process:

```
mdutil -E {{path/to/volume}}
```

mist

MIST - macOS Installer Super Tool.

Automatically download macOS Firmwares/Installers.

More information: https://github.com/ninxsoft/mist-cli.

- List all available macOS Firmwares for Apple Silicon Macs:

```
mist list firmware
```

- List all available macOS Installers for Intel Macs, including Universal Installers for macOS Big Sur and later:

```
mist list installer
```

- List all macOS Installers that are compatible with this Mac, including Universal Installers for macOS Big Sur and later:

```
mist list installer --compatible
```

- List all available macOS Installers for Intel Macs, including betas, also including Universal Installers for macOS Big Sur and later:

```
mist list installer --include-betas
```

- List only the latest macOS Sonoma Installer for Intel Macs, including Universal Installers for macOS Big Sur and later:

```
mist list installer --latest "macOS Sonoma"
```

- List and export macOS Installers to a CSV file:

```
mist list installer --export "/{{path/to/export.csv}}"
```

- Download the latest macOS Sonoma Firmware for Apple Silicon Macs, with a custom name:

```
mist download firmware "macOS Sonoma" --firmware-name
"{{Install %NAME% %VERSION%-%BUILD%.ipsw}}"
```

- Download a specific macOS Installer version for Intel Macs, including Universal Installers for macOS Big Sur and later:

```
mist download installer "{{13.5.2}}" application
```

mktemp

Create a temporary file or directory.

More information: https://keith.github.io/xcode-man-pages/mktemp.1.html.

- Create an empty temporary file and print its absolute path:

```
mktemp
```

- Use a custom directory (defaults to the output of `getconf DARWIN_USER_TEMP_DIR`, or `/tmp`):

```
mktemp --tmpdir /{{path/to/temporary_directory}}
```

- Use a custom path template (Xs are replaced with random alphanumeric characters):

```
mktemp {{/tmp/example.XXXXXXXX}}
```

- Use a custom file name prefix:

```
mktemp -t {{example}}
```

- Create an empty temporary directory and print its absolute path:

```
mktemp --directory
```

mysides

Add, list and remove finder favorites.

More information: https://github.com/mosen/mysides.

- List sidebar favorites:

```
mysides list
```

- Add a new item to the end of the sidebar favorites:

```
mysides add {{example}} {{file:///Users/Shared/example}}
```

- Remove an item by name:

```
mysides remove {{example}}
```

- Add the current directory to the sidebar:

```
mysides add $(basename $(pwd)) file:///$(pwd)
```

- Remove the current directory from the sidebar:

```
mysides remove $(basename $(pwd))
```

netstat

Display network-related information such as open connections, open socket ports, etc.

See also: **`lsof`** for listing network connections, including listening ports.

More information: https://keith.github.io/xcode-man-pages/netstat.1.html.

- Display the PID and program name listening on a specific protocol:

```
netstat -p {{protocol}}
```

- Print the routing table and do not resolve IP addresses to hostnames:

```
netstat -nr
```

- Print the routing table of IPv4 addresses:

```
netstat -nr -f inet
```

nettop

Display updated information about the network.

More information: https://keith.github.io/xcode-man-pages/nettop.1.html.

- Monitor TCP and UDP sockets from all interfaces:

```
nettop
```

- Monitor TCP sockets from Loopback interfaces:

```
nettop -m {{tcp}} -t {{loopback}}
```

- Monitor a specific process:

```
nettop -p "{{process_id|process_name}}"
```

- Display a per-process summary:

```
nettop -P
```

- Print 10 samples of network information:

```
nettop -l {{10}}
```

- Monitor changes every 5 seconds:

```
nettop -d -s {{5}}
```

- While running nettop, list interactive commands:

```
<h>
```

- Display help:

```
nettop -h
```

networkQuality

Measure the network quality by connecting to the internet.

More information: https://support.apple.com/101942.

- Test the network quality for the default interface:

```
networkQuality
```

- Test the upload and download speeds sequentially instead of in parallel:

```
networkQuality -s
```

- Test a specified network interface:

```
networkQuality -I {{en0}}
```

- Test the network quality with verbose output:

```
networkQuality -v
```

networksetup

Configuration tool for Network System Preferences.

More information: https://support.apple.com/guide/remote-desktop/about-networksetup-apdd0c5a2d5/mac.

- List available network service providers (Ethernet, Wi-Fi, Bluetooth, etc):

```
networksetup -listallnetworkservices
```

- Show network settings for a particular networking device:

```
networksetup -getinfo "{{Wi-Fi}}"
```

- Get currently connected Wi-Fi network name (Wi-Fi device usually en0 or en1):

```
networksetup -getairportnetwork {{en0}}
```

- Connect to a particular Wi-Fi network:

```
networksetup -setairportnetwork {{en0}} {{Airport Network SSID}} {{password}}
```

nfcd

This daemon controls the NFC controller.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/nfcd.8.html.

- Start the daemon:

```
nfcd
```

notifyd

Notification server.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/notifyd.8.html.

- Start the daemon:

```
notifyd
```

- Log debug messages to the default log file (`/var/log/notifyd.log`):

```
notifyd -d
```

- Log debug messages to an alternate log file:

```
notifyd -d -log_file {{path/to/log_file}}
```

nvram

Manipulate firmware variables.

More information: https://keith.github.io/xcode-man-pages/nvram.8.html.

- [p]rint all the variables stored in the NVRAM:

```
nvram -p
```

- [p]rint all the variables stored in the NVRAM using [x]ML format:

```
nvram -xp
```

- Modify the value of a firmware variable:

```
sudo nvram {{name}}="{{value}}"
```

- [d]elete a firmware variable:

```
sudo nvram -d {{name}}
```

- [c]lear all the firmware variables:

```
sudo nvram -c
```

- Set a firmware variable from a specific [x]ML [f]ile:

```
sudo nvram -xf {{path/to/file.xml}}
```

ocspd

This retrieves and caches Certificate Revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP) responses for certificate verification.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/ocspd.1.html.

- Start the daemon:

```
ocspd
```

open

Open files, directories and applications.

More information: https://keith.github.io/xcode-man-pages/open.1.html.

- Open a file with the associated application:

```
open {{file.ext}}
```

- Run a graphical macOS [a]pplication:

```
open -a "{{Application}}"
```

- Run a graphical macOS app based on the [b]undle identifier (refer to `osascript` for an easy way to get this):

```
open -b {{com.domain.application}}
```

- Open the current directory in Finder:

```
open .
```

- [R]eveal a file in Finder:

```
open -R {{path/to/file}}
```

- Open all the files of a given extension in the current directory with the associated application:

```
open {{*.ext}}
```

- Open a [n]ew instance of an application specified via [b]undle identifier:

```
open -n -b {{com.domain.application}}
```

opensnoop

Track file opens on your system.

More information: https://keith.github.io/xcode-man-pages/opensnoop.1m.html.

- Print all file opens as they occur:

```
sudo opensnoop
```

- Track all file opens by a process by name:

```
sudo opensnoop -n "{{process_name}}"
```

- Track all file opens by a process by PID:

```
sudo opensnoop -p {{PID}}
```

- Track which processes open a specified file:

```
sudo opensnoop -f {{path/to/file}}
```

orb

Interface for OrbStack, a fast and lightweight container and virtual machine runtime for macOS.

Provides Docker-compatible commands and Linux VM management.

More information: https://docs.orbstack.dev/.

- List all containers and VMs:

```
orb list
```

- Create and start a Linux virtual machine:

```
orb create {{vm_name}}
```

- Create a VM with a specific Linux distribution:

```
orb create {{vm_name}} {{ubuntu|fedora|arch|debian}}
```

- Start or stop a virtual machine:

```
orb {{start|stop}} {{vm_name}}
```

- Connect to a VM via SSH:

```
orb ssh {{vm_name}}
```

- Execute a command in a VM:

```
orb exec {{vm_name}} {{command}}
```

- Delete a virtual machine:

```
orb delete {{vm_name}}
```

- Show system status and resource usage:

```
orb status
```

osascript

Run AppleScript or JavaScript for Automation (JXA).

More information: https://keith.github.io/xcode-man-pages/osascript.1.html.

- Run an AppleScript command:

```
osascript -e '{{say "Hello world"}}'
```

- Run multiple AppleScript commands:

```
osascript -e '{{say "Hello"}}' -e '{{say "world"}}'
```

- Run a compiled (`*.scpt`), bundled (`*.scptd`), or plaintext (`*.applescript`) AppleScript file:

```
osascript {{path/to/apple.scpt}}
```

- Get the bundle identifier of an application (useful for `open -b`):

```
osascript -e 'id of app "{{Application}}"'
```

- Run a JavaScript command:

```
osascript -l JavaScript -e "{{console.log('Hello world');}}"
```

- Run a JavaScript file:

```
osascript -l JavaScript {{path/to/script.js}}
```

pbcopy

Copy data from **`stdin`** to the clipboard.

Comparable to pressing **`<Cmd c>`** on the keyboard.

More information: https://keith.github.io/xcode-man-pages/pbcopy.1.html.

- Place the contents of a specific file in the clipboard:

```
pbcopy < {{path/to/file}}
```

- Place the results of a specific command in the clipboard:

```
find . -type t -name "*.png" | pbcopy
```

pbpaste

Send the contents of the clipboard to **`stdout`**.

Comparable to pressing **`<Cmd v>`** on the keyboard.

More information: https://keith.github.io/xcode-man-pages/pbcopy.1.

- Write the contents of the clipboard to a file:

```
pbpaste > {{path/to/file}}
```

- Use the contents of the clipboard as input to a command:

```
pbpaste | {{grep search_string}}
```

photoanalysisd

Analyze photo libraries for Memories, People, and scene or object based search.

`photoanalysisd` should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/photoanalysisd.8.html.

- Start the daemon:

```
photoanalysisd
```

photolibraryd

This handles all photo library requests.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/photolibraryd.8.html.

- Start the daemon:

```
photolibraryd
```

ping

Send ICMP ECHO_REQUEST packets to network hosts.

More information: https://keith.github.io/xcode-man-pages/ping.8.html.

- Ping the specified host:

```
ping "{{hostname}}"
```

- Ping a host a specific number of times:

```
ping -c {{count}} "{{host}}"
```

- Ping a host, specifying the interval in seconds between requests (default is 1 second):

```
ping -i {{seconds}} "{{host}}"
```

- Ping a host without trying to lookup symbolic names for addresses:

```
ping -n "{{host}}"
```

- Ping a host and ring the bell when a packet is received (if your terminal supports it):

```
ping -a "{{host}}"
```

- Ping a host and prints the time a packet was received (this option is an Apple addition):

```
ping --apple-time "{{host}}"
```

pkgutil

Query and manipulate Mac OS X Installer packages and receipts.

More information: https://keith.github.io/xcode-man-pages/pkgutil.1.html.

- List package IDs for all installed packages:

```
pkgutil --pkgs
```

- Verify cryptographic signatures of a package file:

```
pkgutil --check-signature {{path/to/file.pkg}}
```

- List all the files for an installed package given its ID:

```
pkgutil --files {{com.microsoft.Word}}
```

- Extract the contents of a package file into a directory:

```
pkgutil --expand-full {{path/to/file.pkg}} {{path/to/
directory}}
```

plutil

View, convert, validate, or edit property list ("plist") files.

More information: https://keith.github.io/xcode-man-pages/plutil.1.html.

- Display the contents of one or more plist files in human-readable format:

```
plutil -p {{file1.plist file2.plist ...}}
```

- Convert one or more plist files to XML format, overwriting the original files in-place:

```
plutil -convert xml1 {{file1.plist file2.plist ...}}
```

- Convert one or more plist files to binary format, overwriting the original files in-place:

```
plutil -convert binary1 {{file1.plist file2.plist ...}}
```

- Convert a plist file to a different format, writing to a new file:

```
plutil -convert {{xml1|binary1|json|swift|objc}} {{path/to/file.plist}} -o {{path/to/new_file.plist}}
```

- Convert a plist file to a different format, writing to `stdout`:

```
plutil -convert {{xml1|binary1|json|swift|objc}} {{path/to/file.plist}} -o -
```

pmset

Configure macOS power management settings, as one might do in System Preferences > Energy Saver.

Commands that modify settings must begin with **sudo**.

More information: https://keith.github.io/xcode-man-pages/pmset.1.html.

- Display the current power management settings:

```
pmset -g
```

- Display the current power source and battery levels:

```
pmset -g batt
```

- Put display to sleep immediately:

```
pmset displaysleepnow
```

- Set display to never sleep when on charger power:

```
sudo pmset -c displaysleep 0
```

- Set display to sleep after 15 minutes when on battery power:

```
sudo pmset -b displaysleep 15
```

- Schedule computer to automatically wake up every weekday at 9 AM:

```
sudo pmset repeat wake MTWRF 09:00:00
```

- Restore to system defaults:

```
sudo pmset -a displaysleep 10 disksleep 10 sleep 30 womp 1
```

pod

Dependency manager for Swift and Objective-C Cocoa projects.

More information: https://guides.cocoapods.org/terminal/commands.html.

- Create a Podfile for the current project with the default contents:

```
pod init
```

- Download and install all pods defined in the Podfile (that haven't been installed before):

```
pod install
```

- List all available pods:

```
pod list
```

- Show the outdated pods (of those currently installed):

```
pod outdated
```

- Update all currently installed pods to their newest version:

```
pod update
```

- Update a specific (previously installed) pod to its newest version:

```
pod update {{pod_name}}
```

- Remove CocoaPods from a Xcode project:

```
pod deintegrate {{xcode_project}}
```

port

Package manager for macOS.

More information: https://guide.macports.org/#using.port.

- Search for a package:

```
port search {{search_term}}
```

- Install a package:

```
sudo port install {{package}}
```

- List installed packages:

```
port installed
```

- Update port and fetch the latest list of available packages:

```
sudo port selfupdate
```

- Upgrade outdated packages:

```
sudo port upgrade outdated
```

- Remove old versions of installed packages:

```
sudo port uninstall inactive
```

ps

Information about running processes.

More information: https://keith.github.io/xcode-man-pages/ps.1.html.

- List all running processes:

```
ps aux
```

- List all running processes including the full command string:

```
ps auxww
```

- Search for a process that matches a string:

```
ps aux | grep {{string}}
```

- Get the parent PID of a process:

```
ps -o ppid= -p {{pid}}
```

- Sort processes by memory usage:

```
ps -m
```

- Sort processes by CPU usage:

```
ps -r
```

qlmanage

QuickLook server tool.

More information: https://keith.github.io/xcode-man-pages/qlmanage.1.html.

- Display QuickLook for one or multiple files:

```
qlmanage -p {{path/to/file1 path/to/file2 ...}}
```

- Compute 300px wide PNG thumbnails of all JPEGs in the current directory and put them in a directory:

```
qlmanage {{*.jpg}} -t -s {{300}} {{path/to/directory}}
```

- Reset QuickLook:

```
qlmanage -r
```

rargs

Execute a command for each line of **`stdin`**.

Like **`xargs`**, but with pattern matching support.

More information: https://github.com/lotabout/rargs.

- Execute a command for every line of input, just like `xargs` (`{0}` indicates where to substitute in the text):

```
{{command}} | rargs {{command}} {0}
```

- Do a dry run, which prints the commands that would be run instead of executing them:

```
{{command}} | rargs -e {{command}} {0}
```

- Remove the `.bak` extension from every file in a list:

```
{{command}} | rargs -p '(.*).bak mv {0} {1}
```

- Execute commands in parallel:

```
{{command}} | rargs -w {{max-procs}}
```

- Consider each line of input to be separated by a NUL character (`\0`) instead of a newline (`\n`):

```
{{command}} | rargs -0 {{command}} {0}
```

reboot

Reboot the system.

More information: https://keith.github.io/xcode-man-pages/reboot.8.html.

- Reboot immediately:

```
sudo reboot
```

- Reboot immediately without gracefully shutting down:

```
sudo reboot -q
```

rev

Reverse lines of text.

More information: https://keith.github.io/xcode-man-pages/rev.1.html.

- Reverse each line in a file to `stdout`:

```
rev {{path/to/file}}
```

- Reverse each line from `stdin` to `stdout`:

```
{{command}} | rev
```

route

Manually manipulate the routing tables.

Requires root privileges.

More information: https://keith.github.io/xcode-man-pages/route.8.html.

- Add a route to a destination through a gateway:

```
sudo route add "{{destination_ip_address}}"
"{{gateway_address}}"
```

- Add a route to a /24 subnet through a gateway:

```
sudo route add "{{subnet_ip_address}}/24"
"{{gateway_address}}"
```

- Run in test mode (does not do anything, just print):

```
sudo route -t add "{{destination_ip_address}}/24"
"{{gateway_address}}"
```

- Remove all routes:

```
sudo route flush
```

- Delete a specific route:

```
sudo route delete "{{destination_ip_address}}/24"
```

- Lookup and display the route for a destination (hostname or IP address):

```
sudo route get "{{destination}}"
```

SafeEjectGPU

Eject a GPU safely.

More information: https://keith.github.io/xcode-man-pages/SafeEjectGPU.8.html.

- Eject all GPUs:

```
SafeEjectGPU Eject
```

- List all GPUs attached:

```
SafeEjectGPU gpus
```

- List apps using a GPU:

```
SafeEjectGPU gpuid {{GPU_ID}} apps
```

- Get the status of a GPU:

```
SafeEjectGPU gpuid {{GPU_ID}} status
```

- Eject a GPU:

```
SafeEjectGPU gpuid {{GPU_ID}} Eject
```

- Launch an app on a GPU:

```
SafeEjectGPU gpuid {{GPU_ID}} LaunchOnGPU {{path/to/App.app}}
```

say

Convert text to speech.

More information: https://keith.github.io/xcode-man-pages/say.1.html.

- Say a phrase aloud:

```
say "{{I like to ride my bike.}}"
```

- Read a file aloud:

```
say --input-file {{filename.txt}}
```

- Say a phrase with a custom voice and speech rate:

```
say --voice {{voice}} --rate {{words_per_minute}} "{{I'm sorry Dave, I can't let you do that.}}"
```

- List the available voices (different voices speak in different languages):

```
say --voice "?"
```

- Say something in Polish:

```
say --voice {{Zosia}} "{{Litwo, ojczyzno moja!}}"
```

- Create an audio file of the spoken text:

```
say --output-file {{filename.aiff}} "{{Here's to the Crazy Ones.}}"
```

screencapture

Utility to take screenshots and screen recordings.

More information: https://keith.github.io/xcode-man-pages/screencapture.1.html.

- Take a screenshot and save it to a file:

```
screencapture {{path/to/file.png}}
```

- Take a screenshot including the mouse cursor:

```
screencapture -C {{path/to/file.png}}
```

- Take a screenshot and open it in Preview, instead of saving:

```
screencapture -P
```

- Take a screenshot of a selected rectangular area:

```
screencapture -i {{path/to/file.png}}
```

- Take a screenshot after a delay:

```
screencapture -T {{seconds}} {{path/to/file.png}}
```

- Make a screen recording and save it to a file:

```
screencapture -v {{path/to/file.mp4}}
```

scutil

Manage system configuration parameters.

Require root privileges when setting configuration.

More information: https://keith.github.io/xcode-man-pages/scutil.8.html.

- Display DNS Configuration:

```
scutil --dns
```

- Display proxy configuration:

```
scutil --proxy
```

- Get computer name:

```
scutil --get ComputerName
```

- Set computer name:

```
sudo scutil --set ComputerName {{computer_name}}
```

- Get hostname:

```
scutil --get HostName
```

- Set hostname:

```
scutil --set HostName {{hostname}}
```

sdef

Get or generate a scripting definitions (**sdef**) file from a scriptable application.

More information: https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ScriptableCocoaApplications/SApps_creating_sdef/SAppsCreateSdef.html.

- Print the scripting definitions of the given application:

```
sdef {{/Applications/XCode.app}}
```

secd

Controls access to and modification of keychain items.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/secd.8.html.

- Start the daemon:

```
secd
```

security

Administer keychains, keys, certificates and the Security framework.

More information: https://keith.github.io/xcode-man-pages/security.1.html.

- List all available keychains:

```
security list-keychains
```

- Delete a specific keychain:

```
security delete-keychain {{path/to/file.keychain}}
```

- Create a keychain:

```
security create-keychain -p {{password}} {{path/to/
file.keychain}}
```

- Set a certificate to use with a website or [s]ervice by its [c]ommon name (fails if several certificates with the same common name exist):

```
security set-identity-preference -s {{URL|hostname|service}}
-c "{{common_name}}" {{path/to/file.keychain}}
```

- Add a certificate from file to a [k]eychain (if -k isn't specified, the default keychain is used):

```
security add-certificates -k {{file.keychain}} {{path/to/
cert_file.pem}}
```

- Add a CA certificate to the per-user Trust Settings:

```
security add-trusted-cert -k {{path/to/user-
keychain.keychain-db}} {{path/to/ca-cert_file.pem}}
```

- Remove a CA certificate from the per-user Trust Settings:

```
security remove-trusted-cert {{path/to/ca-cert_file.pem}}
```

securityd

This manages security contexts and cryptographic operations.

Works with secd for keychain access.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/securityd.1.html.

- Start the daemon:

```
securityd
```

sed

Edit text in a scriptable manner.

See also: **awk**, **ed**.

More information: https://keith.github.io/xcode-man-pages/sed.1.html.

- Replace all `apple` (basic `regex`) occurrences with `mango` (basic `regex`) in all input lines and print the result to `stdout`:

```
{{command}} | sed 's/apple/mango/g'
```

- Execute a specific script [f]ile and print the result to `stdout`:

```
{{command}} | sed -f {{path/to/script_file.sed}}
```

- Replace all `apple` (extended `regex`) occurrences with `APPLE` (extended `regex`) in all input lines and print the result to `stdout`:

```
{{command}} | sed -E 's/(apple)/\U\1/g'
```

- Print just a first line to `stdout`:

```
{{command}} | sed -n '1p'
```

- Replace all `apple` (basic `regex`) occurrences with `mango` (basic `regex`) in a `file` and save a backup of the original to `file.bak`:

```
sed -i bak 's/apple/mango/g' {{path/to/file}}
```

setfile

Set file attributes on files in an HFS+ directory.

More information: https://ss64.com/osx/setfile.html.

- Set creation date for specific files:

```
setfile -d "{{MM/DD/YYYY HH:MM:SS}}" {{path/to/file1 path/to/file2 ...}}
```

- Set modification date for specific files:

```
setfile -m "{{MM/DD/YYYY HH:MM:SS}}" {{path/to/file1 path/to/file2 ...}}
```

- Set modification date for symlink file (not to linked file itself):

```
setfile -P -m "{{MM/DD/YYYY HH:MM:SS}}" {{path/to/file1 path/to/file2 ...}}
```

shortcuts

Manage shortcuts.

Note: You can also use the **`Shortcuts`** app.

More information: https://support.apple.com/guide/shortcuts-mac/run-shortcuts-from-the-command-line-apd455c82f02/mac.

- Run the specified shortcut (`Count holidays`):

  ```
  shortcuts run "{{Count holidays}}"
  ```

- Print all shortcuts:

  ```
  shortcuts list
  ```

- Print all shortcut folders:

  ```
  shortcuts list --folders
  ```

- Open the specified shortcut (`Count holidays`) in the Shortcuts editor:

  ```
  shortcuts view "{{Count holidays}}"
  ```

shuf

Generate random permutations.

More information: https://manpagez.com/man/1/shuf/.

- Randomize the order of lines in a file and output the result:

```
shuf {{path/to/file}}
```

- Only output the first 5 entries of the result:

```
shuf --head-count=5 {{path/to/file}}
```

- Write output to another file:

```
shuf {{path/to/input_file}} --output {{ath/to/output_file}}
```

- Generate random numbers in the range 1 to 10:

```
shuf --input-range=1-10
```

shutdown

Shutdown and reboot the system.

More information: https://keith.github.io/xcode-man-pages/shutdown.8.html.

- Power off (halt) immediately:

```
shutdown -h now
```

- Sleep immediately:

```
shutdown -s now
```

- Reboot immediately:

```
shutdown -r now
```

- Reboot in 5 minutes:

```
shutdown -r "+{{5}}"
```

- Power off (halt) at 1:00 pm (Uses 24h clock):

```
shutdown -h {{1300}}
```

- Reboot on May 10th 2042 at 11:30 am (Input format: YYMMDDHHMM):

```
shutdown -r {{4205101130}}
```

signal

Simplified software signal facilities.

More information: https://developer.apple.com/library/archive/documentation/System/Conceptual/ManPages_iPhoneOS/man3/signal.3.html.

- View documentation for signals in macOS:

```
man signal
```

sips

Apple Scriptable Image Processing System.

Raster/Query images and ColorSync ICC Profiles.

More information: https://keith.github.io/xcode-man-pages/sips.1.html.

- Specify an output directory so that originals do not get modified:

```
sips --out {{path/to/output_directory}}
```

- Resample image at specified size, Image aspect ratio may be altered:

```
sips --resampleHeightWidth {{1920}} {{300}}
{{image_file.ext}}
```

- Resample image so height and width aren't greater than specified size (notice the capital Z):

```
sips --resampleHeightWidthMax {{1920}} {{300}}
{{image_file.ext}}
```

- Resample all images in a directory to fit a width of 960px (honoring aspect ratio):

```
sips --resampleWidth {{960}} {{path/to/images}}
```

- Convert an image from CMYK to RGB:

```
sips --matchTo "/System/Library/ColorSync/Profiles/Generic
RGB Profile.icc" {{path/to/image.ext}} {{path/to/
output_directory}}
```

- Remove ColorSync ICC profile from an image:

```
sips --deleteProperty profile --
deleteColorManagementProperties {{path/to/image_file.ext}}
```

sntp

A very Simple Network Time Protocol client program.

More information: https://keith.github.io/xcode-man-pages/sntp.1.

- Query a specified SNTP server and display the time:

```
sntp {{pool.ntp.org}}
```

- Synchronize the system clock with a specified SNTP server:

```
sudo sntp -S {{pool.ntp.org}}
```

- Enable debug logging:

```
sntp -d {{pool.ntp.org}}
```

sntpd

An SNTP server.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/sntpd.8.html.

- Start the daemon:

```
sntpd
```

- Overwrite existing state with the local clock (stratum 1), for running a master/primary server, without synchronizing with another (higher stratum) server:

```
sntpd -L
```

- Use a custom file for the SNTP state:

```
sntpd -z {{path/to/state.bin}}
```

softwareupdate

Update macOS App Store apps.

More information: https://keith.github.io/xcode-man-pages/softwareupdate.8.html.

- List all available updates:

```
softwareupdate --list
```

- Download and install all updates:

```
softwareupdate --install --all
```

- Download and install all [r]ecommended updates:

```
softwareupdate --install --recommended
```

- Download and install a specific app:

```
softwareupdate --install {{update_name}}
```

spctl

Manage the security assessment policy subsystem.

Utility for managing Gatekeeper in macOS.

More information: https://keith.github.io/xcode-man-pages/spctl.8.html.

- Turn off Gatekeeper:

```
spctl --master-disable
```

- Add a rule to allow an application to run (labeling of rule is optional):

```
spctl --add --label {{rule_name}} {{path/to/file}}
```

- Turn on Gatekeeper:

```
spctl --master-enable
```

- List all rules on the system:

```
spctl --list
```

split

Split a file into pieces.

More information: https://keith.github.io/xcode-man-pages/split.1.html.

- Split a file, each split having 10 lines (except the last split):

```
split -l 10 {{path/to/file}}
```

- Split a file by a `regex`. The matching line will be the first line of the next output file:

```
split -p {{cat|^[dh]og}} {{path/to/file}}
```

- Split a file with 512 bytes in each split (except the last split; use 512k for kilobytes and 512m for megabytes):

```
split -b 512 {{path/to/file}}
```

- Split a file into 5 files. File is split such that each split has same size (except the last split):

```
split -n 5 {{path/to/file}}
```

spotify

A command-line interface to Spotify.

More information: https://github.com/hnarayanan/shpotify.

- Find a song by name and play it:

```
spotify play {{song_name}}
```

- Find a playlist by name and play it:

```
spotify play list {{playlist_name}}
```

- Pause (or resume) playback:

```
spotify pause
```

- Skip to the next song in a playlist:

```
spotify next
```

- Change volume:

```
spotify vol {{up|down|value}}
```

- Show the playback status and song details:

```
spotify status
```

stat

Display file status.

More information: https://keith.github.io/xcode-man-pages/stat.1.html.

- Show file properties such as size, permissions, creation and access dates among others:

```
stat {{path/to/file}}
```

- Same as above but verbose (more similar to Linux's `stat`):

```
stat -x {{path/to/file}}
```

- Show only octal file permissions:

```
stat -f %Mp%Lp {{path/to/file}}
```

- Show owner and group of the file:

```
stat -f "%Su %Sg" {{path/to/file}}
```

- Show the size of the file in bytes:

```
stat -f "%z %N" {{path/to/file}}
```

sw_vers

Print macOS operating system version information.

More information: https://keith.github.io/xcode-man-pages/sw_vers.1.html.

- Print all available information (OS name, version number, and build):

```
sw_vers
```

- Print only the version number of the operating system:

```
sw_vers -productVersion
```

- Print only the build identifier:

```
sw_vers -buildVersion
```

symptomsd

Provides services for **`Symptoms.framework`**.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/symptomsd.8.html.

- Start the daemon:

```
symptomsd
```

sysctl

Access kernel state information.

More information: https://keith.github.io/xcode-man-pages/sysctl.8.html.

- Show all available variables and their values:

```
sysctl -a
```

- Show Apple model identifier:

```
sysctl -n hw.model
```

- Show CPU model:

```
sysctl -n machdep.cpu.brand_string
```

- Show available CPU features (MMX, SSE, SSE2, SSE3, AES, etc):

```
sysctl -n machdep.cpu.features
```

- Set a changeable kernel state variable:

```
sysctl -w {{section.tunable}}={{value}}
```

system_profiler

Report system hardware and software configuration.

More information: https://keith.github.io/xcode-man-pages/system_profiler.8.html.

- Display a report with specific details level (mini [no personal information], basic or full):

```
system_profiler -detailLevel {{level}}
```

- Display a full system profiler report which can be opened by `System Profiler.app`:

```
system_profiler -xml > MyReport.spx
```

- Display a hardware overview (Model, CPU, Memory, Serial, etc) and software data (System, Kernel, Name, Uptime, etc):

```
system_profiler SPHardwareDataType SPSoftwareDataType
```

- Print the system serial number:

```
system_profiler SPHardwareDataType|grep "Serial Number (system)" | awk '{ print $4 }'
```

systemsetup

Configure System Preferences machine settings.

More information: https://support.apple.com/guide/remote-desktop/about-systemsetup-apd95406b8d/mac.

- Enable remote login (SSH):

```
systemsetup -setremotelogin on
```

- Specify timezone, NTP Server and enable network time:

```
systemsetup -settimezone "{{US/Pacific}}" -
setnetworktimeserver {{us.pool.ntp.org}} -setusingnetworktime
on
```

- Make the machine never sleep and automatically restart on power failure or kernel panic:

```
systemsetup -setsleep off -setrestartpowerfailure on -
setrestartfreeze on
```

- List valid startup disks:

```
systemsetup -liststartupdisks
```

- Specify a new startup disk:

```
systemsetup -setstartupdisk {{path/to/directory}}
```

systemsoundserverd

Core Audio related daemon.

It should not be invoked manually.

- Start the daemon:

  ```
  systemsoundserverd
  ```

tag

Edit tags on Mac OS X files (10.9 Mavericks and above).

More information: https://github.com/jdberry/tag/.

- Add tags to a file:

```
tag --add {{tag_name1,tag_name2,...}} {{path/to/file}}
```

- Remove a tag:

```
tag --remove {{tag_name}} {{path/to/file}}
```

- Remove all tags from a file:

```
tag --remove \* {{path/to/file}}
```

- Show all files with a given tag:

```
tag --match {{tag_name}}
```

tail

Display the last part of a file.

See also: **head**.

More information: https://keith.github.io/xcode-man-pages/tail.1.html.

- Show last 8 lines in file:

```
tail -n 8 {{path/to/file}}
```

- Print a file from a specific line number:

```
tail -n +{{8}} {{path/to/file}}
```

- Print a specific count of bytes from the end of a given file:

```
tail -c {{8}} {{path/to/file}}
```

- Print the last lines of a given file and keep reading it until `<Ctrl c>`:

```
tail -f {{path/to/file}}
```

- Keep reading file until `<Ctrl c>`, even if the file is inaccessible:

```
tail -F {{path/to/file}}
```

- Show last `count` lines in a file and refresh every `seconds` seconds:

```
tail -n {{count}} -s {{seconds}} -f {{path/to/file}}
```

tart

Build, run and manage macOS and Linux virtual machines (VMs) on Apple Silicon.

More information: https://github.com/cirruslabs/tart.

- Pull a remote VM image:

```
tart pull {{acme.io/org/name:tag}}
```

- Clone a VM from a local or remote image source:

```
tart clone {{source-vm}} {{vm-name}}
```

- Create a new Mac VM from a specific ipsw file:

```
tart create --from-ipsw {{latest|path/to/file.ipsw}} {{vm-name}}
```

- Run an existing VM:

```
tart run {{vm-name}}
```

- Run an existing VM with a specific mounted directory:

```
tart run --dir {{path/to/directory}}:/{{path/to/local_directory}} {{vm-name}}
```

- List VMs:

```
tart list
```

- Get IP address of a running VM:

```
tart ip {{vm-name}}
```

- Change a VM's display resolution:

```
tart set {{vm-name}} --display {{640}}x{{400}}
```

terminal-notifier

Send macOS User Notifications.

More information: https://github.com/julienXX/terminal-notifier.

- Send a notification (only the message is required):

```
terminal-notifier -group {{tldr-info}} -title {{TLDR}} -
message '{{TLDR rocks}}'
```

- Display piped data with a sound:

```
echo '{{Piped Message Data!}}' | terminal-notifier -sound
{{default}}
```

- Open a URL when the notification is clicked:

```
terminal-notifier -message '{{Check your Apple stock!}}' -
open '{{http://finance.yahoo.com/q?s=AAPL}}'
```

- Open an app when the notification is clicked:

```
terminal-notifier -message '{{Imported 42 contacts.}}' -
activate {{com.apple.AddressBook}}
```

textutil

Manipulate text files of various formats.

More information: https://keith.github.io/xcode-man-pages/textutil.1.html.

- Display information about `file.rtf`:

```
textutil -info {{path/to/file.rtf}}
```

- Convert `file.rtf` into `file.html`:

```
textutil -convert html {{path/to/file.rtf}}
```

- Convert rich text to normal text:

```
textutil {{path/to/file.rtf}} -convert txt
```

- Convert `file.txt` into `file.rtf`, using Times 10 for the font:

```
textutil -convert rtf -font Times -fontsize 10 {{path/to/
file.txt}}
```

- Load all RTF files in the current directory, concatenates their contents, and writes the result out as `index.html` with the HTML title set to "Several Files":

```
textutil -cat {{html}} -title "Several Files" -output {{path/
to/index.html}} *.rtf
```

timed

Service that synchronizes system time (e.g. using NTP).

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/timed.8.html.

- Start the daemon:

```
timed
```

tmutil

Utility for managing Time Machine backups. Most verbs require root privileges.

More information: https://keith.github.io/xcode-man-pages/tmutil.8.html.

- Set an HFS+ drive as the backup destination:

```
sudo tmutil setdestination {{path/to/disk_mount_point}}
```

- Set an APF share or SMB share as the backup destination:

```
sudo tmutil setdestination "{{protocol://
user[:password]@host/share}}"
```

- Append the given destination to the list of destinations:

```
sudo tmutil setdestination -a {{destination}}
```

- Enable automatic backups:

```
sudo tmutil enable
```

- Disable automatic backups:

```
sudo tmutil disable
```

- Start a backup, if one is not running already, and release control of the shell:

```
sudo tmutil startbackup
```

- Start a backup and block until the backup is finished:

```
sudo tmutil startbackup -b
```

- Stop a backup:

```
sudo tmutil stopbackup
```

top

Display dynamic real-time information about running processes.

More information: https://keith.github.io/xcode-man-pages/top.1.html.

- Start `top`, all options are available in the interface:

```
top
```

- Start `top` sorting processes by internal memory size (default order - process ID):

```
top -o mem
```

- Start `top` sorting processes first by CPU, then by running time:

```
top -o cpu -O time
```

- Start `top` displaying only processes owned by given user:

```
top -user {{user_name}}
```

- Display help about interactive commands:

```
<?>
```

translationd

Enables Translation features.

It should not be invoked manually.

- Start the daemon:

  ```
  translationd
  ```

uname

Print details about the current machine and the operating system running on it.

Note: For additional information about the operating system, try the **`sw_vers`** command.

More information: https://keith.github.io/xcode-man-pages/uname.1.html.

- Print kernel name:

```
uname
```

- Print system architecture and processor information:

```
uname -mp
```

- Print kernel name, kernel release and kernel version:

```
uname -srv
```

- Print system hostname:

```
uname -n
```

- Print all available system information:

```
uname -a
```

universalaccessd

Get universal access services.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/universalaccessd.8.html.

- Start the daemon:

```
universalaccessd
```

uptime

Tell how long the system has been running and other information.

More information: https://keith.github.io/xcode-man-pages/uptime.1.html.

- Print current time, uptime, number of logged-in users and other information:

```
uptime
```

usernoted

Provides notification services.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/usernoted.8.html.

- Start the daemon:

```
usernoted
```

uuidgen

Generate new UUID (Universally Unique IDentifier) strings.

More information: https://keith.github.io/xcode-man-pages/uuidgen.1.html.

- Generate a UUID string:

```
uuidgen
```

valet

A Laravel development environment that allows hosting sites via local tunnels on **`http://example.test`**.

More information: https://laravel.com/docs/valet.

- Start the valet daemon:

  ```
  valet start
  ```

- Register the current working directory as a path that Valet should search for sites:

  ```
  valet park
  ```

- View "parked" paths:

  ```
  valet paths
  ```

- Serve a single site instead of an entire directory:

  ```
  valet link {{application_name}}
  ```

- Share a project via an Ngrok tunnel:

  ```
  valet share
  ```

vm_stat

Show virtual memory statistics.

More information: https://keith.github.io/xcode-man-pages/vm_stat.1.html.

- Display virtual memory statistics:

```
vm_stat
```

- Display reports every 2 seconds for 5 times:

```
vm_stat -c {{5}} {{2}}
```

vpnd

Listens for incoming VPN connections.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/vpnd.8.html.

- Start the daemon:

```
vpnd
```

- Run the daemon in the foreground:

```
vpnd -x
```

- Run the daemon in the foreground and print logs to the terminal:

```
vpnd -d
```

- Run the daemon in the foreground, print logs to the terminal, and quit after validating arguments:

```
vpnd -n
```

- Run the daemon for a specific server configuration:

```
vpnd -i {{server_id}}
```

- Display help:

```
vpnd -h
```

w

Show who is logged on and what they are doing.

Print user login, TTY, remote host, login time, idle time, current process.

More information: https://keith.github.io/xcode-man-pages/w.1.html.

- Show logged-in users information:

```
w
```

- Show logged-in users information without a header:

```
w -h
```

- Show information about logged-in users, sorted by their idle time:

```
w -i
```

wacaw

Capture both still pictures and video from an attached camera.

More information: https://webcam-tools.sourceforge.net.

- Take a picture from webcam:

```
wacaw {{filename}}
```

- Record a video:

```
wacaw --video {{filename}} --duration {{10}}
```

- Take a picture with custom resolution:

```
wacaw --width {{width}} --height {{100}} {{filename}}
```

- Copy image just taken to clipboard:

```
wacaw --to-clipboard
```

- List the devices available:

```
wacaw --list-devices
```

warmd

Controls caches used during startup and login.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/warmd.8.html.

- Start the daemon:

```
warmd
```

watchdogd

Works with the Watchdog KEXT to ensure that the system is healthy and running.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/watchdogd.8.html.

- Start the daemon:

```
watchdogd
```

watchlistd

Manage the Apple TV app's watch list.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/watchlistd.8.html.

- Start the daemon:

```
watchlistd
```

wc

Count lines, words, or bytes.

More information: https://keith.github.io/xcode-man-pages/wc.1.html.

- Count lines in file:

```
wc -l {{path/to/file}}
```

- Count words in file:

```
wc -w {{path/to/file}}
```

- Count characters (bytes) in file:

```
wc -c {{path/to/file}}
```

- Count characters in file (taking multi-byte character sets into account):

```
wc -m {{path/to/file}}
```

- Use `stdin` to count lines, words and characters (bytes) in that order:

```
{{find .}} | wc
```

webinspectord

Relays commands between Web Inspector and remote targets like WKWebView.

It should not be invoked manually.

More information: https://www.manpagez.com/man/8/webinspectord/.

- Start the daemon:

```
webinspectord
```

whatis

Search a set of database files for short descriptions of system commands for keywords.

More information: https://www.linfo.org/whatis.html.

- Search for information about keyword:

```
whatis {{keyword}}
```

- Search for information about multiple keywords:

```
whatis {{keyword1}} {{keyword2}}
```

wifi-password

Get the password of the Wi-Fi.

More information: https://github.com/rauchg/wifi-password.

- Get the password for the Wi-Fi you are currently logged onto:

```
wifi-password
```

- Get the password for the Wi-Fi with a specific SSID:

```
wifi-password {{ssid}}
```

- Print only the password as output:

```
wifi-password -q
```

wifivelocityd

XPC helper for performing system context actions for the WiFiVelocity framework.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/wifivelocityd.8.html.

- Start the daemon:

```
wifivelocityd
```

wps

Assists AirPort in connecting to a network using Wireless Protected Setup.

It should not be invoked manually.

More information: https://www.manpagez.com/man/8/wps/.

- Start the daemon:

```
wps
```

wwand

USB WWAN device configuration daemon.

It should not be invoked manually.

More information: https://web.archive.org/web/20230331164459/https://keith.github.io/xcode-man-pages/wwand.8.html.

- Start the daemon:

```
wwand
```

xartstorageremoted

The xART Remote Storage Daemon. Receives save/fetch requests from the CoProcessor.

It should not be invoked manually.

More information: https://keith.github.io/xcode-man-pages/xartstorageremoted.8.html.

- Start the daemon:

```
xartstorageremoted
```

xattr

Utility to work with extended filesystem attributes.

More information: https://keith.github.io/xcode-man-pages/xattr.1.html.

- List key:value extended attributes for a given file:

```
xattr -l {{file}}
```

- Write an attribute for a given file:

```
xattr -w {{attribute_key}} {{attribute_value}} {{file}}
```

- Delete an attribute from a given file:

```
xattr -d {{com.apple.quarantine}} {{file}}
```

- Delete all extended attributes from a given file:

```
xattr -c {{file}}
```

- Recursively delete an attribute in a given directory:

```
xattr -rd {{attribute_key}} {{directory}}
```

xcode-select

Switch between different versions of Xcode and the included developer tools.

Also used to update the path to Xcode if it is moved after installation.

More information: https://developer.apple.com/library/archive/technotes/tn2339/_index.html.

- Install Xcode's command-line tools:

```
xcode-select --install
```

- Select a given path as the active developer directory:

```
xcode-select --switch {{path/to/Xcode.app/Contents/Developer}}
```

- Select a given Xcode instance and use its developer directory as the active one:

```
xcode-select --switch {{path/to/Xcode_file.app}}
```

- Print the currently selected developer directory:

```
xcode-select --print-path
```

- Discard any user-specified developer directory so that it will be found via the default search mechanism:

```
sudo xcode-select --reset
```

xcodebuild

Build Xcode projects.

More information: https://developer.apple.com/library/archive/technotes/tn2339/_index.html.

- Build workspace:

```
xcodebuild -workspace {{workspace_name.workspace}} -scheme {{scheme_name}} -configuration {{configuration_name}} clean build SYMROOT={{SYMROOT_path}}
```

- Build project:

```
xcodebuild -target {{target_name}} -configuration {{configuration_name}} clean build SYMROOT={{SYMROOT_path}}
```

- Show SDKs:

```
xcodebuild -showsdks
```

xcodes runtimes

Manage Xcode Simulator runtimes.

More information: https://github.com/xcodesorg/xcodes.

- Display all available Simulator runtimes:

```
xcodes runtimes --include-betas
```

- Download a Simulator runtime:

```
xcodes runtimes download {{runtime_name}}
```

- Download and install a Simulator runtime:

```
xcodes runtimes install {{runtime_name}}
```

- Download/install a Simulator runtime for specific iOS/watchOS/tvOS/visionOS version (must be written as case-sensitive):

```
xcodes runtimes {{download|install}} "{{iOS|watchOS|tvOS|visionOS}} {{runtime_version}}"
```

- Set a specific location where the runtime archive will be first downloaded (defaults to `~/Downloads`):

```
xcodes runtimes {{download|install}} {{runtime_name}} --directory {{path/to/directory}}
```

- Do not delete the downloaded archive when the Simulator is successfully installed:

```
xcodes runtimes install {{runtime_name}} --keep-archive
```

xcodes

Download, install and manage multiple Xcode versions.

See also: **`xcodes runtimes`**.

More information: https://github.com/xcodesorg/xcodes.

- List all installed Xcode versions:

```
xcodes installed
```

- List all available Xcode versions:

```
xcodes list
```

- Select an Xcode version by specifying a version number or a path:

```
xcodes select {{xcode_version|path/to/Xcode.app}}
```

- Download and install a specific Xcode version:

```
xcodes install {{xcode_version}}
```

- Install the latest Xcode release and select it:

```
xcodes install --latest --select
```

- Download a specific Xcode version archive to a given directory without installing it:

```
xcodes download {{xcode_version}} --directory {{path/to/directory}}
```

xcrun

Run or locate development tools and properties.

More information: https://keith.github.io/xcode-man-pages/xcrun.1.html.

- Find and run a tool from the active developer directory:

```
xcrun {{tool}} {{arguments}}
```

- Show verbose output:

```
xcrun {{tool}} {{arguments}} --verbose
```

- Find a tool for a given SDK:

```
xcrun --sdk {{sdk_name}}
```

- Find a tool for a given toolchain:

```
xcrun --toolchain {{name}}
```

- Display help:

```
xcrun --help
```

- Display version:

```
xcrun --version
```

xctool

Build Xcode projects.

More information: https://github.com/facebookarchive/xctool.

- Build a single project without any workspace:

```
xctool -project {{YourProject.xcodeproj}} -scheme
{{YourScheme}} build
```

- Build a project that is part of a workspace:

```
xctool -workspace {{YourWorkspace.xcworkspace}} -scheme
{{YourScheme}} build
```

- Clean, build and execute all the tests:

```
xctool -workspace {{YourWorkspace.xcworkspace}} -scheme
{{YourScheme}} clean build test
```

xed

Open files for editing in Xcode.

More information: https://keith.github.io/xcode-man-pages/xed.1.html.

- Open file in Xcode:

```
xed {{path/to/file1 path/to/file2 ...}}
```

- Open file(s) in Xcode, create if it doesn't exist:

```
xed --create {{path/to/file1 path/to/file2 ...}}
```

- Open a file in Xcode and jump to line number 75:

```
xed --line 75 {{path/to/file}}
```

xip

Create or expand compressed files in a secure xip archive.

Only archives signed by Apple are trusted, so this tool should not be used to create archives.

More information: https://keith.github.io/xcode-man-pages/xip.1.html.

- Expand the archive into the current working directory:

```
xip --expand {{path/to/file.xip}}
```

xml2man

Compile MPGL to mdoc.

More information: https://developer.apple.com/library/archive/documentation/DeveloperTools/Conceptual/HeaderDoc/mpgl/mpgl.html.

- Compile an MPGL file to a viewable man page:

```
xml2man {{path/to/command_file.mxml}}
```

- Compile an MPGL file to a specific output file:

```
xml2man {{path/to/service_file.mxml}} {{path/to/service_file.7}}
```

- Compile an MPGL file to a specific output file, overwriting if it already exists:

```
xml2man -f {{path/to/function_file.mxml}} {{path/to/function_file.3}}
```

xsand

Xsan file system management daemon. Provides services for the Xsan file system.

It should not be invoked manually.

More information: https://developer.apple.com/support/downloads/Xsan-Management-Guide.pdf.

- Start the daemon:

```
xsand
```

xsltproc

Transform XML with XSLT to produce output (usually HTML or XML).

More information: https://manned.org/xsltproc.

- Transform an XML file with a specific XSLT stylesheet:

```
xsltproc --output {{path/to/output_file.html}} {{path/to/
stylesheet_file.xslt}} {{path/to/file.xml}}
```

- Pass a value to a parameter in the stylesheet:

```
xsltproc --output {{path/to/output_file.html}} --stringparam
"{{name}}" "{{value}}" {{path/to/stylesheet_file.xslt}}
{{path/to/xml_file.xml}}
```

yaa

Create and manipulate YAA archives.

More information: https://www.manpagez.com/man/1/yaa/.

- Create an archive from a directory:

```
yaa archive -d {{path/to/directory}} -o {{path/to/
output_file.yaa}}
```

- Create an archive from a file:

```
yaa archive -i {{path/to/file}} -o {{path/to/
output_file.yaa}}
```

- Extract an archive to the current directory:

```
yaa extract -i {{path/to/archive_file.yaa}}
```

- List the contents of an archive:

```
yaa list -i {{path/to/archive_file.yaa}}
```

- Create an archive with a specific compression algorithm:

```
yaa archive -a {{algorithm}} -d {{path/to/directory}} -o
{{path/to/output_file.yaa}}
```

- Create an archive with an 8 MB block size:

```
yaa archive -b 8m -d {{path/to/directory}} -o {{path/to/
output_file.yaa}}
```

yabai

A tiling window manager for macOS based on binary space partitioning.

More information: https://github.com/koekeishiya/yabai/wiki.

- Send a config [m]essage for setting the layout:

```
yabai -m config layout {{bsp|stack|float}}
```

- Set the window gap in pt:

```
yabai -m config window_gap {{10}}
```

- Enable opacity:

```
yabai -m config window_opacity on
```

- Disable window shadow:

```
yabai -m config window_shadow off
```

- Enable status bar:

```
yabai -m config status_bar on
```

Sunos

devfsadm

Administration command for **/dev**. Maintains the **/dev** namespace.

More information: https://www.unix.com/man-page/sunos/1m/devfsadm.

- Scan for new disks:

```
devfsadm -c disk
```

- Cleanup any dangling /dev links and scan for new device:

```
devfsadm -C -v
```

- Dry-run - output what would be changed but make no modifications:

```
devfsadm -C -v -n
```

dmesg

Write the kernel messages to **`stdout`**.

More information: https://www.unix.com/man-page/sunos/1m/dmesg.

- Show kernel messages:

```
dmesg
```

- Show how much physical memory is available on this system:

```
dmesg | grep -i memory
```

- Show kernel messages 1 page at a time:

```
dmesg | less
```

prctl

Get or set the resource controls of running processes, tasks, and projects.

More information: https://www.unix.com/man-page/sunos/1/prctl.

- Examine process limits and permissions:

```
prctl {{pid}}
```

- Examine process limits and permissions in machine parsable format:

```
prctl -P {{pid}}
```

- Get specific limit for a running process:

```
prctl -n process.max-file-descriptor {{pid}}
```

prstat

Report active process statistics.

More information: https://www.unix.com/man-page/sunos/1m/prstat.

- Examine all processes and reports statistics sorted by CPU usage:

```
prstat
```

- Examine all processes and reports statistics sorted by memory usage:

```
prstat -s rss
```

- Report total usage summary for each user:

```
prstat -t
```

- Report microstate process accounting information:

```
prstat -m
```

- Print out a list of top 5 CPU using processes every second:

```
prstat -c -n 5 -s cpu 1
```

share

Make local resource/filesystem available for mounting by remote systems.

More information: https://docs.oracle.com/cd/E36784_01/html/E36825/gntjt.html.

- List all currently shared file systems:

```
share
```

- Share a directory with read/write access:

```
share -F nfs -o rw /{{path/to/directory}}
```

- Share a directory with read-only access:

```
share -F nfs -o ro /{{path/to/directory}}
```

- Share a directory with specific options (e.g., allow root access from a specific host):

```
share -F nfs -o rw,root={{hostname}} /{{path/to/directory}}
```

- Make sharing persistent by adding entries to `/etc/dfs/dfstab`:

```
echo "share -F nfs -o rw /{{path/to/directory}}" >> /etc/dfs/dfstab
```

snoop

Network packet sniffer.

SunOS equivalent of tcpdump.

More information: https://www.unix.com/man-page/sunos/1m/snoop.

- Capture packets on a specific network interface:

```
snoop -d {{e1000g0}}
```

- Save captured packets in a file instead of displaying them:

```
snoop -o {{path/to/file}}
```

- Display verbose protocol layer summary of packets from a file:

```
snoop -V -i {{path/to/file}}
```

- Capture network packets that come from a hostname and go to a given port:

```
snoop to port {{port}} from host {{hostname}}
```

- Capture and show a hex-dump of network packets exchanged between two IP addresses:

```
snoop -x0 -p4 {{ip1}} {{ip2}}
```

svcadm

Manipulate service instances.

More information: https://www.unix.com/man-page/linux/1m/svcadm.

- Enable a service in the service database:

```
svcadm enable {{service_name}}
```

- Disable service:

```
svcadm disable {{service_name}}
```

- Restart a running service:

```
svcadm restart {{service_name}}
```

- Command service to re-read configuration files:

```
svcadm refresh {{service_name}}
```

- Clear a service from maintenance state and command it to start:

```
svcadm clear {{service_name}}
```

svccfg

Import, export, and modify service configurations.

More information: https://www.unix.com/man-page/linux/1m/svccfg.

- Validate configuration file:

```
svccfg validate {{path/to/smf_file.xml}}
```

- Export service configurations to file:

```
svccfg export {{servicename}} > {{path/to/smf_file.xml}}
```

- Import/update service configurations from file:

```
svccfg import {{path/to/smf_file.xml}}
```

svcs

List information about running services.

More information: https://www.unix.com/man-page/linux/1/svcs.

- List all running services:

```
svcs
```

- List services that are not running:

```
svcs -vx
```

- List information about a service:

```
svcs apache
```

- Show location of service log file:

```
svcs -L apache
```

- Display end of a service log file:

```
tail $(svcs -L apache)
```

truss

Troubleshooting tool for tracing system calls.

SunOS equivalent of strace.

More information: https://www.unix.com/man-page/linux/1/truss.

- Start tracing a program by executing it, following all child processes:

```
truss -f {{program}}
```

- Start tracing a specific process by its PID:

```
truss -p {{pid}}
```

- Start tracing a program by executing it, showing arguments and environment variables:

```
truss -a -e {{program}}
```

- Count time, calls, and errors for each system call and report a summary on program exit:

```
truss -c -p {{pid}}
```

- Trace a process filtering output by system call:

```
truss -p {{pid}} -t {{system_call_name}}
```

Windows

Add-AppxPackage

A PowerShell utility to add a signed app package (**`.appx`**, **`.msix`**, **`.appxbundle`** and **`.msixbundle`**) to a user account.

More information: https://learn.microsoft.com/powershell/module/appx/Add-AppxPackage.

- Add an app package:

```
Add-AppxPackage -Path {{path\to\package.msix}}
```

- Add an app package with dependencies:

```
Add-AppxPackage -Path {{path\to\package.msix}} -
DependencyPath {{path\to\dependencies.msix}}
```

- Install an app using the app installer file:

```
Add-AppxPackage -AppInstallerFile
{{path\to\app.appinstaller}}
```

- Add an unsigned package:

```
Add-AppxPackage -Path {{path\to\package.msix}} -
DependencyPath {{path\to\dependencies.msix}} -AllowUnsigned
```

assoc

Display or change associations between file extensions and file types.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/assoc.

- List all associations between file extensions and file types:

```
assoc
```

- Display the associated file type for a specific extension:

```
assoc {{.txt}}
```

- Set the associated file type for a specific extension:

```
assoc .{{txt}}={{txtfile}}
```

- View the output of `assoc` one screen at a time:

```
assoc | {{more}}
```

attrib

Display or change attributes of files or directories.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/attrib.

- Display all set attributes of files in the current directory:

```
attrib
```

- Display all set attributes of files in a specific directory:

```
attrib {{path\to\directory}}
```

- Display all set attributes of files and [d]irectories in the current directory:

```
attrib /d
```

- Display all set attributes of files in the current directory and [s]ub-directories:

```
attrib /s
```

- Add the `[r]ead-only` or `[a]rchive` or `[s]ystem` or `[h]idden` or `not content [i]ndexed` attribute to files or directories:

```
attrib +{{r|a|s|h|i}} {{path\to\file_or_directory1 path\to\file_or_directory2 ...}}
```

- Remove a specific attribute of files or directories:

```
attrib -{{r|a|s|h|i}} {{path\to\file_or_directory1 path\to\file_or_directory2 ...}}
```

azcopy

A file transfer tool for uploading to Azure Cloud Storage Accounts.

More information: https://learn.microsoft.com/azure/storage/common/storage-use-azcopy-v10.

- Log in to an Azure Tenant:

```
azcopy login
```

- Upload a local file:

```
azcopy copy '{{path\to\source_file}}' 'https://
{{storage_account_name}}.blob.core.windows.net/
{{container_name}}/{{blob_name}}'
```

- Upload files with `.txt` and `.jpg` extensions:

```
azcopy copy '{{path\to\source_directory}}' 'https://
{{storage_account_name}}.blob.core.windows.net/
{{container_name}}' --include-pattern '{{*.txt;*.jpg}}'
```

- Copy a container directly between two Azure storage accounts:

```
azcopy copy 'https://
{{source_storage_account_name}}.blob.core.windows.net/
{{container_name}}' 'https://
{{destination_storage_account_name}}.blob.core.windows.net/
{{container_name}}'
```

- Synchronize a local directory and delete files in the destination if they no longer exist in the source:

```
azcopy sync '{{path\to\source_directory}}' 'https://
{{storage_account_name}}.blob.core.windows.net/
{{container_name}}' --recursive --delete-destination=true
```

- Display help:

```
azcopy --help
```

bcdboot

Configure or repair boot files.

More information: https://learn.microsoft.com/windows-hardware/manufacture/desktop/bcdboot-command-line-options-techref-di.

- Initialize the system partition by using BCD files from the source Windows folder:

```
bcdboot {{C:\Windows}}
```

- Enable [v]erbose mode:

```
bcdboot {{C:\Windows}} /v
```

- Specify the volume letter of the [s]ystem partition:

```
bcdboot {{C:\Windows}} /s {{S:}}
```

- Specify a [l]ocale:

```
bcdboot {{C:\Windows}} /l {{en-us}}
```

- Specify a [f]irmware type while copying the boot files to a specified volume:

```
bcdboot {{C:\Windows}} /s {{S:}} /f {{UEFI|BIOS|ALL}}
```

bleachbit

This command is an alias of **`bleachbit_console`**.

- View documentation for the original command:

```
tldr bleachbit_console
```

bleachbit_console

Clean junk files on the filesystem.

More information: https://docs.bleachbit.org/doc/command-line-interface.html.

- Start the graphical user interface (GUI) version of Bleachbit:

```
bleachbit_console.exe --gui
```

- Shred a file:

```
bleachbit_console.exe --shred {{path/to/file}}
```

- List available cleaner options:

```
bleachbit_console.exe --list-cleaners
```

- Preview the files that will be deleted and other changes that will be made before actually performing the clean-up operation:

```
bleachbit_console.exe --preview {{--preset|cleaner1.option1 cleaner2.* ...}}
```

- Perform the clean-up operation and delete files:

```
bleachbit_console.exe --clean {{--preset|cleaner1.option1 cleaner2.* ...}}
```

cd

Display the current working directory or move to a different directory.

In PowerShell, this command is an alias of **`Set-Location`**. This documentation is based on the Command Prompt (**`cmd`**) version of **`cd`**.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/cd.

- View documentation of the equivalent PowerShell command:

```
tldr set-location
```

- Display the path of the current directory:

```
cd
```

- Go to a specific directory on the same drive:

```
cd {{path\to\directory}}
```

- Go to a specific directory on a different [d]rive:

```
cd /d {{C}}:{{path\to\directory}}
```

- Go up to the parent of the current directory:

```
cd ..
```

- Go to the home directory of the current user:

```
cd %userprofile%
```

- Go to root of current drive:

```
cd \
```

certutil

A tool to manage and configure certificate information.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/certutil.

- Dump the configuration information or files:

```
certutil {{filename}}
```

- Encode a file in hexadecimal:

```
certutil -encodehex {{path\to\input_file}}
{{path\to\output_file}}
```

- Encode a file to Base64:

```
certutil -encode {{path\to\input_file}}
{{path\to\output_file}}
```

- Decode a Base64-encoded file:

```
certutil -decode {{path\to\input_file}}
{{path\to\output_file}}
```

- Generate and display a cryptographic hash over a file:

```
certutil -hashfile {{path\to\input_file}} {{md2|md4|md5|sha1|
sha256|sha384|sha512}}
```

chdir

This command is an alias of **`cd`** in Command Prompt, and subsequently **`Set-Location`** in PowerShell.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/chdir.

- View documentation for the original Command Prompt command:

```
tldr cd
```

- View documentation for the original PowerShell command:

```
tldr set-location
```

chkdsk

Check file system and volume metadata for errors.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/chkdsk.

- Specify the drive letter (followed by a colon), mount point, or volume name to check:

```
chkdsk {{volume}}
```

- Fix errors on a specific volume:

```
chkdsk {{volume}} /f
```

- Dismount a specific volume before checking:

```
chkdsk {{volume}} /x
```

- Change the log file size to the specified size (only for NTFS):

```
chkdsk /l{{size}}
```

choco apikey

Manage API keys for Chocolatey sources.

More information: https://chocolatey.org/docs/commands-apikey.

- Display a list of sources and their API keys:

```
choco apikey
```

- Display a specific source and its API key:

```
choco apikey {{[-s|--source]}} "{{source_url}}"
```

- Set an API key for a source:

```
choco apikey {{[-s|--source]}} "{{source_url}}" {{[-k|--api-key]}} "{{api_key}}"
```

- Remove an API key for a source:

```
choco apikey {{[-s|--source]}} "{{source_url}}" --remove
```

choco feature

Interact with features with Chocolatey.

More information: https://chocolatey.org/docs/commands-feature.

- Display a list of available features:

  ```
  choco feature list
  ```

- Enable a feature:

  ```
  choco feature enable --name {{name}}
  ```

- Disable a feature:

  ```
  choco feature disable --name {{name}}
  ```

choco info

Display detailed information about a package with Chocolatey.

More information: https://chocolatey.org/docs/commands-info.

- Display information on a specific package:

```
choco info {{package}}
```

- Display information for a local package only:

```
choco info {{package}} {{[-l|--local-only]}}
```

- Specify a custom source to receive packages information from:

```
choco info {{package}} {{[-s|--source]}} {{source_url|alias}}
```

- Provide a username and password for authentication:

```
choco info {{package}} {{[-u|--user]}} {{username}} {{[-p|--password]}} {{password}}
```

choco install

Install one or more packages with Chocolatey.

More information: https://chocolatey.org/docs/commands-install.

- Install one or more packages:

```
choco install {{package1 package2 ...}}
```

- Install packages from a custom configuration file:

```
choco install {{path\to\packages_file.config}}
```

- Install a specific `nuspec` or `nupkg` file:

```
choco install {{path\to\file}}
```

- Install a specific version of a package:

```
choco install {{package}} --version {{version}}
```

- Allow installing multiple versions of a package:

```
choco install {{package}} --allow-multiple
```

- Confirm all prompts automatically:

```
choco install {{package}} --yes
```

- Specify a custom source to receive packages from:

```
choco install {{package}} --source {{source_url|alias}}
```

- Provide a username and password for authentication:

```
choco install {{package}} --user {{username}} --password
{{password}}
```

choco list

Display a list of packages with Chocolatey.

More information: https://chocolatey.org/docs/commands-list.

- Display all available packages:

```
choco list
```

- Display all locally installed packages:

```
choco list --local-only
```

- Display a list including local programs:

```
choco list {{[-i|--include-programs]}}
```

- Display only approved packages:

```
choco list --approved-only
```

- Specify a custom source to display packages from:

```
choco list {{[-s|--source]}} {{source_url|alias}}
```

- Provide a username and password for authentication:

```
choco list --user {{username}} --password {{password}}
```

choco new

Generate new package specification files with Chocolatey.

More information: https://chocolatey.org/docs/commands-new.

- Create a new package skeleton:

```
choco new {{package}}
```

- Create a new package with a specific version:

```
choco new {{package}} --version {{version}}
```

- Create a new package with a specific maintainer name:

```
choco new {{package}} --maintainer {{maintainer_name}}
```

- Create a new package in a custom output directory:

```
choco new {{package}} {{[--out|--output-directory]}} {{path/to/directory}}
```

- Create a new package with specific 32-bit and 64-bit installer URLs:

```
choco new {{package}} url="{{url}}" url64="{{url}}"
```

choco outdated

Check for outdated packages with Chocolatey.

More information: https://chocolatey.org/docs/commands-outdated.

- Display a list of outdated packages in table format:

  ```
  choco outdated
  ```

- Ignore pinned packages in the output:

  ```
  choco outdated --ignore-pinned
  ```

- Specify a custom source to check packages from:

  ```
  choco outdated --source {{source_url|alias}}
  ```

- Provide a username and password for authentication:

  ```
  choco outdated --user {{username}} --password {{password}}
  ```

choco pack

Package a NuGet specification into a **nupkg** file.

More information: https://chocolatey.org/docs/commands-pack.

- Package a NuGet specification to a nupkg file:

```
choco pack {{path\to\specification_file}}
```

- Package a NuGet specification specifying the version of the resulting file:

```
choco pack {{path\to\specification_file}} --version
{{version}}
```

- Package a NuGet specification to a specific directory:

```
choco pack {{path\to\specification_file}} {{[--out|--output-
directory]}} {{path\to\output_directory}}
```

choco pin

Pin a package at a version with Chocolatey.

Pinned packages are skipped automatically when upgrading.

More information: https://chocolatey.org/docs/commands-pin.

- Display a list of pinned packages and their versions:

```
choco pin list
```

- Pin a package at its current version:

```
choco pin add {{[-n|--name]}} {{package}}
```

- Pin a package at a specific version:

```
choco pin add {{[-n|--name]}} {{package}} --version
{{version}}
```

- Remove a pin for a specific package:

```
choco pin remove {{[-n|--name]}} {{package}}
```

choco push

Push a compiled NuGet package (**nupkg**) to a package feed.

More information: https://docs.chocolatey.org/en-us/create/commands/push.

- Push a compiled `nupkg` to the specified feed:

```
choco push {{[-s|--source]}} {{https://push.chocolatey.org/}}
```

- Push a compiled `nupkg` to the specified feed with a timeout in seconds (default is 2700):

```
choco push {{[-s|--source]}} {{https://push.chocolatey.org/}}
{{[--timeout|--execution-timeout]}} {{500}}
```

choco search

Search for a local or remote package with Chocolatey.

More information: https://chocolatey.org/docs/commands-search.

- Search for a package:

```
choco search {{query}}
```

- Search for a package locally:

```
choco search {{query}} --local-only
```

- Only include exact matches in the results:

```
choco search {{query}} {{[-e|--exact]}}
```

- Confirm all prompts automatically:

```
choco search {{query}} {{[-y|--yes]}}
```

- Specify a custom source to search for packages in:

```
choco search {{query}} {{[-s|--source]}} {{source_url|alias}}
```

- Provide a username and password for authentication:

```
choco search {{query}} {{[-u|--user]}} {{username}} {{[-p|--password]}} {{password}}
```

choco source

Manage sources for packages with Chocolatey.

More information: https://chocolatey.org/docs/commands-source.

- List currently available sources:

```
choco source list
```

- Add a new package source:

```
choco source add {{[-n|--name]}} {{name}} {{[-s|--source]}} {{url}}
```

- Add a new package source with credentials:

```
choco source add {{[-n|--name]}} {{name}} {{[-s|--source]}} {{url}} {{[-u|--user]}} {{username}} {{[-p|--password]}} {{password}}
```

- Add a new package source with a client certificate:

```
choco source add {{[-n|--name]}} {{name}} {{[-s|--source]}} {{url}} --cert {{path\to\certificate_file}}
```

- Enable a package source:

```
choco source enable {{[-n|--name]}} {{name}}
```

- Disable a package source:

```
choco source disable {{[-n|--name]}} {{name}}
```

- Remove a package source:

```
choco source remove {{[-n|--name]}} {{name}}
```

choco uninstall

Uninstall packages with Chocolatey.

More information: https://chocolatey.org/docs/commands-uninstall.

- Uninstall one or more packages:

```
choco uninstall {{package1 package2 ...}}
```

- Uninstall a specific version of a package:

```
choco uninstall {{package}} --version {{version}}
```

- Confirm all prompts automatically:

```
choco uninstall {{package}} --yes
```

- Remove all dependencies when uninstalling:

```
choco uninstall {{package}} --remove-dependencies
```

- Uninstall all packages:

```
choco uninstall all
```

choco upgrade

Upgrade one or more packages with Chocolatey.

More information: https://chocolatey.org/docs/commands-upgrade.

- Upgrade one or more packages:

```
choco upgrade {{package1 package2 ...}}
```

- Upgrade to a specific version of a package:

```
choco upgrade {{package}} --version {{version}}
```

- Upgrade all packages:

```
choco upgrade all
```

- Upgrade all except specified comma-separated packages:

```
choco upgrade all --except "{{package1,package2,...}}"
```

- Confirm all prompts automatically:

```
choco upgrade {{package}} --yes
```

- Specify a custom source to receive packages from:

```
choco upgrade {{package}} --source {{source_url|alias}}
```

- Provide a username and password for authentication:

```
choco upgrade {{package}} --user {{username}} --password
{{password}}
```

choco

The Chocolatey package manager.

Some subcommands such as **`install`**, **`upgrade`**, **`pin`** have their own usage documentation.

More information: https://docs.chocolatey.org/en-us/choco/commands/.

- Install a package:

```
choco install {{package_name}}
```

- Upgrade a specific installed package:

```
choco upgrade {{package_name}}
```

- Upgrade all outdated packages and automatically confirm all prompts:

```
choco upgrade all {{[-y|--yes]}}
```

- Uninstall a package and automatically confirm all prompts:

```
choco uninstall {{package_name}} {{[-y|--yes]}}
```

- Search for packages by name or keyword:

```
choco search {{query}}
```

- List all packages installed on the machine:

```
choco list
```

- Show packages that have newer versions available:

```
choco outdated
```

- Install a package from a specific source:

```
choco install {{package_name}} {{[-s|--source]}} {{source}}
```

choice

Prompt user to select a choice and return the selected choice index.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/choice.

- Prompt the current user to select a `Y` or `N` choice:

```
choice
```

- Prompt the current user to select a [c]hoice from a specific set:

```
choice /c {{AB}}
```

- Prompt the current user to select a choice with a specific [m]essage:

```
choice /m "{{message}}"
```

- Prompt the current user to select a [c]ase-[s]ensitive [c]hoice from a specific set:

```
choice /cs /c {{Ab}}
```

- Prompt the current user to select a choice and prefer the [d]efault choice in a specific [t]ime:

```
choice /t {{5}} /d {{default_choice}}
```

- Display help:

```
choice /?
```

chrome

This command is an alias of **`chromium`**.

More information: https://chrome.google.com.

- View documentation for the original command:

```
tldr chromium
```

chromium

Open-source web browser principally developed and maintained by Google.

Note: You may need to replace the **`chromium`** command with your desired web browser, such as **`brave`**, **`google-chrome`**, **`microsoft-edge`**/**`msedge`**, **`opera`**, or **`vivaldi`**.

More information: https://www.chromium.org/developers/how-tos/run-chromium-with-flags/.

- Open a specific URL or file:

```
chromium {{https://example.com|path\to\file.html}}
```

- Open in incognito mode (use `--inprivate` for Microsoft Edge):

```
{{chromium --incognito|msedge --inprivate}} {{example.com}}
```

- Open in a new window:

```
chromium --new-window {{example.com}}
```

- Open in application mode (without toolbars, URL bar, buttons, etc.):

```
chromium --app {{https://example.com}}
```

- Use a proxy server:

```
chromium --proxy-server "{{socks5://hostname:
66}}" {{example.com}}
```

- Open with a custom profile directory:

```
chromium --user-data-dir {{path\to\directory}}
```

- Open without CORS validation (useful to test an API):

```
chromium --user-data-dir {{path\to\directory}} --disable-web-
security
```

- Open with a DevTools window for each tab opened:

```
chromium --auto-open-devtools-for-tabs
```

cinst

This command is an alias of **`choco install`**.

- View documentation for the original command:

```
tldr choco install
```

cipher

Display or alter the encryption of directories and files on NTFS volumes.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/cipher.

- Display information about a specific encrypted file or directory:

```
cipher /c:{{path\to\file_or_directory}}
```

- [e]ncrypt a file or directory (files added later to the directory are also encrypted as the directory is marked):

```
cipher /e:{{path\to\file_or_directory}}
```

- [d]ecrypt a file or directory:

```
cipher /d:{{path\to\file_or_directory}}
```

- Securely remove a file or directory:

```
cipher /w:{{path\to\file_or_directory}}
```

cl

The Microsoft C/C++ compiler for compiling and linking source code files.

More information: https://learn.microsoft.com/cpp/build/reference/compiler-command-line-syntax.

- Compile a source file:

```
cl {{path/to/source.c}}
```

- Compile and create an executable with a custom name:

```
cl /Fe {{path/to/output_executable}} {{path/to/source.c}}
```

- Compile a source file with optimization enabled:

```
cl /O2 {{path/to/source.c}}
```

- Compile a source file and create a debug executable:

```
cl /Zi {{path/to/source.c}}
```

- Compile multiple source files:

```
cl {{path/to/source1.c path/to/source2.c ...}}
```

- Specify the output directory for compiled files:

```
cl /Fo {{path/to/output_directory}}/ {{path/to/source.c}}
```

- Compile with warnings as errors:

```
cl /WX {{path/to/source.c}}
```

cleanmgr

Clear unnecessary files from the computer's hard disk.

Deprecated in newer Windows versions and replaced by "Storage Sense".

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/cleanmgr.

- Open Disk Cleanup for a specific drive:

```
cleanmgr /d {{C}}
```

- Open Disk Cleanup with all options selected by default:

```
cleanmgr /d {{C}} /lowdisk
```

- Clean up all files automatically without user prompts:

```
cleanmgr /d {{C}} /verylowdisk
```

- Configure which files to clean and save the settings to a specific profile:

```
cleanmgr /sageset:{{profile_number}}
```

- Run Disk Cleanup using a previously configured profile:

```
cleanmgr /sagerun:{{profile_number}}
```

- Clean up files left after a Windows upgrade:

```
cleanmgr /autoclean
```

- Clean up files from a previous Windows installation:

```
cleanmgr /setup
```

- Display help:

```
cleanmgr /?
```

Clear-History

Delete entries from the PowerShell session command history.

Note: **`clhy`** can be used as an alias for **`Clear-History`**.

More information: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/clear-history.

- Clear all command history from current session:

```
Clear-History
```

- Clear command by specific name:

```
Clear-History -CommandLine "{{command}}"
```

- Clear multiple commands by name:

```
Clear-History -CommandLine {{"command1", "command2", ...}}
```

- Clear a specific history entry by ID:

```
Clear-History -Id {{id_number}}
```

- Clear multiple IDs:

```
Clear-History -Id {{id1, id2, ...}}
```

- Clear commands within a range of IDs:

```
Clear-History -Id ({{start_id}}..{{end_id}})
```

- Show what would be deleted:

```
Clear-History -WhatIf
```

- Ask for confirmation before clearing:

```
Clear-History -Confirm
```

Clear-Host

Clears the screen.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/clear-host.

- Clear the screen:

```
cls
```

Clear-RecycleBin

Clear items from the Recycle Bin.

This command can only be used through PowerShell versions 5.1 and below, or 7.1 and above.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/clear-recyclebin.

- Clear and delete all items inside the Recycle Bin:

  ```
  Clear-RecycleBin
  ```

- Clear the Recycle Bin for a specific drive:

  ```
  Clear-RecycleBin -DriveLetter {{C}}
  ```

- Clear the Recycle Bin without further confirmation:

  ```
  Clear-RecycleBin -Force
  ```

clear

In PowerShell, this command is an alias of **`Clear-Host`**.

- View documentation for the original command:

```
tldr clear-host
```

clhy

This command is an alias of **`Clear-History`**.

- View documentation for the original command:

```
tldr Clear-History
```

clip

Copy input content to the Windows clipboard.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/clip.

- Pipe command-line output to the Windows clipboard:

```
{{dir}} | clip
```

- Copy the contents of a file to the Windows clipboard:

```
clip < {{path\to\file.ext}}
```

- Copy text with a trailing newline to the Windows clipboard:

```
echo {{some text}} | clip
```

- Copy text without a trailing newline to the Windows clipboard:

```
echo | set /p="some text" | clip
```

clist

This command is an alias of **`choco list`**.

- View documentation for the original command:

```
tldr choco list
```

cls

Clears the screen.

In PowerShell, this command is an alias of **`Clear-Host`**. This documentation is based on the Command Prompt (**`cmd`**) version of **`cls`**.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/cls.

- View the documentation of the equivalent PowerShell command:

```
tldr clear-host
```

- Clear the screen:

```
cls
```

cmd

The Windows command interpreter.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/cmd.

- Start an interactive shell session:

```
cmd
```

- Execute specific [c]ommands:

```
cmd /c {{echo Hello world}}
```

- Execute a specific script:

```
cmd {{path\to\script.bat}}
```

- Execute specific commands and then enter an interactive shell:

```
cmd /k {{echo Hello world}}
```

- Start an interactive shell session where `echo` is disabled in command output:

```
cmd /q
```

- Start an interactive shell session with delayed [v]ariable expansion enabled or disabled:

```
cmd /v:{{on|off}}
```

- Start an interactive shell session with command [e]xtensions enabled or disabled:

```
cmd /e:{{on|off}}
```

- Start an interactive shell session with used [u]nicode encoding:

```
cmd /u
```

cmdkey

Create, show, and delete stored user names and passwords.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/cmdkey.

- List all user credentials:

```
cmdkey /list
```

- Store credentials for a user that accesses a server:

```
cmdkey /add:{{server_name}} /user:{{user_name}}
```

- Delete credentials for a specific target:

```
cmdkey /delete {{target_name}}
```

cmstp

Manage connection service profiles.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/cmstp.

- Install a specific profile:

```
cmstp "{{path\to\profile_file}}"
```

- Install without creating a desktop shortcut:

```
cmstp /ns "{{path\to\profile_file}}"
```

- Install without checking for dependencies:

```
cmstp /nf "{{path\to\profile_file}}"
```

- Only install for the current user:

```
cmstp /su "{{path\to\profile_file}}"
```

- Install for all users (requires administrator privileges):

```
cmstp /au "{{path\to\profile_file}}"
```

- Install silently without any prompts:

```
cmstp /s "{{path\to\profile_file}}"
```

- Uninstall a specific profile:

```
cmstp /u "{{path\to\profile_file}}"
```

- Uninstall silently without a confirmation prompt:

```
cmstp /u /s "{{path\to\profile_file}}"
```

color

Set the console foreground and background colors.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/color.

- Set the console colors to the default values:

  ```
  color
  ```

- List available color values and detailed information:

  ```
  color /?
  ```

- Set the console foreground and background to a specific color using hexadecimal numbers (`1-9,a-f`):

  ```
  color {{foreground_code}}{{background_code}}
  ```

comp

Compare the contents of two files or sets of files.

Use wildcards (*) to compare sets of files.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/comp.

- Compare files interactively:

```
comp
```

- Compare two specified files:

```
comp {{path\to\file1}} {{path\to\file2}}
```

- Compare two sets of files:

```
comp {{path\to\directory1}}\* {{path\to\directory2}}\*
```

- Display differences in [d]ecimal format:

```
comp /d {{path\to\file1}} {{path\to\file2}}
```

- Display differences in [a]SCII format:

```
comp /a {{path\to\file1}} {{path\to\file2}}
```

- Display [l]ine numbers for differences:

```
comp /l {{path\to\file1}} {{path\to\file2}}
```

- Compare files [c]ase-insensitively:

```
comp /c {{path\to\file1}} {{path\to\file2}}
```

- Compare only the first 5 lines of each file:

```
comp /n=5 {{path\to\file1}} {{path\to\file2}}
```

Compress-Archive

A cmdlet in PowerShell is used to create compressed (zipped) archive files.

More information: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.archive/compress-archive.

- Compress a single file:

```
Compress-Archive -Path {{path\to\file.txt}} -DestinationPath
{{path\to\file.zip}}
```

- Compress multiple files:

```
Compress-Archive -Path {{path\to\file1, path\to\file2, ...}}
-DestinationPath {{path\to\files.zip}}
```

- Compress a directory:

```
Compress-Archive -Path {{path\to\directory}} -DestinationPath
{{path\to\directory.zip}}
```

- Update an existing archive:

```
Compress-Archive -Path {{path\to\file}} -DestinationPath
{{path\to\directory.zip}} -Update
```

- Set compression level:

```
Compress-Archive -Path {{path\to\directory}} -DestinationPath
{{path\to\directory.zip}} -CompressionLevel {{Optimal|
Fastest|NoCompression}}
```

cpush

This command is an alias of **`choco push`**.

- View documentation for the original command:

  ```
  tldr choco push
  ```

cuninst

This command is an alias of **`choco uninstall`**.

- View documentation for the original command:

```
tldr choco uninstall
```

curl

In PowerShell, this command may be an alias of **`Invoke-WebRequest`** when the original **`curl`** program (https://curl.se) is not properly installed.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/invoke-webrequest.

- View documentation for the original `curl` command:

```
tldr curl {{[-p|--platform]}} common
```

- View documentation for PowerShell's `Invoke-WebRequest` command:

```
tldr invoke-webrequest
```

- Check whether `curl` is properly installed by printing its version number. If this command evaluates into an error, PowerShell may have substituted this command with `Invoke-WebRequest`:

```
curl --version
```

date

Display or set the system date.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/date.

- Display the current system date and prompt to enter a new date (leave empty to keep unchanged):

```
date
```

- Display the current system date without prompting for a new date:

```
date /t
```

- Change the current system date to a specific date:

```
date {{month}}-{{day}}-{{year}}
```

del

Delete one or more files.

In PowerShell, this command is an alias of **`Remove-Item`**. This documentation is based on the Command Prompt (**`cmd`**) version of **`del`**.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/del.

- View the documentation of the equivalent PowerShell command:

```
tldr remove-item
```

- Delete one or more files or patterns:

```
del {{file_pattern1 file_pattern2 ...}}
```

- Prompt for confirmation before deleting each file:

```
del {{file_pattern}} /p
```

- Force the deletion of read-only files:

```
del {{file_pattern}} /f
```

- Recursively delete file(s) from all subdirectories:

```
del {{file_pattern}} /s
```

- Do not prompt when deleting files based on a global wildcard:

```
del {{file_pattern}} /q
```

- Delete files based on specified attributes:

```
del {{file_pattern}} /a {{attribute}}
```

- Display help and list available attributes:

```
del /?
```

dir

List directory contents.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/dir.

- Show the contents of the current directory:

```
dir
```

- Show the contents of a given directory:

```
dir {{path\to\directory}}
```

- Show the contents of the current directory, including hidden ones:

```
dir /a
```

- Show the contents of a given directory, including hidden ones:

```
dir {{path\to\directory}} /a
```

- Show a bare list of directories and files, with no additional information:

```
dir /b
```

- Sort results by date/time, oldest first:

```
dir /o:d
```

diskpart

Disk, volume and partition manager.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/diskpart.

- Run diskpart by itself in an administrative command prompt to enter its command-line:

```
diskpart
```

- List all disks:

```
list disk
```

- Select a volume:

```
select volume {{volume}}
```

- Assign a drive letter to the selected volume:

```
assign letter {{letter}}
```

- Create a new partition:

```
create partition primary
```

- Activate the selected volume:

```
active
```

- Exit diskpart:

```
exit
```

doskey

Manage macros, windows commands and command-lines.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/doskey.

- List available macros:

```
doskey /macros
```

- Create a new macro:

```
doskey {{name}} = "{{command}}"
```

- Create a new macro for a specific executable:

```
doskey /exename={{executable}} {{name}} = "{{command}}"
```

- Remove a macro:

```
doskey {{name}} =
```

- Display all commands that are stored in memory:

```
doskey /history
```

- Save macros to a file for portability:

```
doskey /macros > {{path\to\macinit_file}}
```

- Load macros from a file:

```
doskey /macrofile = {{path\to\macinit_file}}
```

driverquery

Display information about installed device drivers.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/driverquery.

- Display a list of all installed device drivers:

```
driverquery
```

- Display a list of drivers in the specified format:

```
driverquery /fo {{table|list|csv}}
```

- Display a list of drivers with a column to indicate if they are signed:

```
driverquery /si
```

- Exclude the header in the output list:

```
driverquery /nh
```

- Display a list of drivers for a remote machine:

```
driverquery /s {{hostname}} /u {{username}} /p {{password}}
```

- Display a list of drivers with verbose information:

```
driverquery /v
```

- Display help:

```
driverquery /?
```

Enable-PnpDevice

The Enable-PnpDevice cmdlet enables a Plug and Play (PnP) device. You must use an Administrator account to enable a device.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/pnpdevice/enable-pnpdevice.

- Enable a device:

```
Enable-PnpDevice -InstanceId 'RETRIEVED USING Get-PnpDevice COMMAND'
```

- Enable all disabled PnP devices:

```
Get-PnpDevice | Where-Object {$_.Problem -eq 22} | Enable-PnpDevice
```

- Enable a device without confirmation:

```
Enable-PnpDevice -InstanceId 'RETRIEVED USING Get-PnpDevice COMMAND' -Confirm:$False
```

- Dry run of what would happen if the cmdlet runs:

```
Enable-PnpDevice -InstanceId 'USB\VID_5986&;PID_0266&;MI_00\7&;1E5D3568&;0&;0000' -WhatIf:$True
```

es

Command-line interface for Everything, a fast file and folder search tool for Windows.

Requires Everything to be installed and running in the background.

More information: https://www.voidtools.com/support/everything/command_line_interface/.

- Search for a file or folder by name:

```
es {{search_term}}
```

- Search using a `regex`:

```
es -r {{regex}}
```

- Match whole words:

```
es -w {{search_term}}
```

- Limit the number of results shown:

```
es -n {{10}} {{search_term}}
```

- Search within a specific folder:

```
es -path {{folder_path}} {{search_term}}
```

- List folders only:

```
es /ad
```

- List files only:

```
es /a-d
```

- Sort results (e.g., by name):

```
es -sort {{name-ascending}}
```

eventcreate

Create custom entries in the event log.

Event IDs can be any number between 1 and 1000.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/eventcreate.

- Create a new event with a given ID (1-1000) in the log:

```
eventcreate /t {{success|error|warning|information}} /id {{id}} /d "{{message}}"
```

- Create an event in a specific event log:

```
eventcreate /l {{log_name}} /t {{type}} /id {{id}} /d "{{message}}"
```

- Create an event with a specific source:

```
eventcreate /so {{source_name}} /t {{type}} /id {{id}} /d "{{message}}"
```

- Create an event in a remote machine's event log:

```
eventcreate /s {{hostname}} /u {{username}} /p {{password}} /t {{type}} /id {{id}} /d "{{message}}"
```

exit

Quit the current CMD instance or the current batch file.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/exit.

- Quit the current CMD instance:

```
exit
```

- Quit the current [b]atch script:

```
exit /b
```

- Quit using a specific exit code:

```
exit {{2}}
```

Expand-Archive

A cmdlet in PowerShell is used to extract files from a compressed archive.

More information: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.archive/expand-archive.

- Extract a ZIP file to a folder:

```
Expand-Archive -Path {{path\to\example.zip}} -DestinationPath {{path\to\extracted_files}}
```

- Overwrite existing files:

```
Expand-Archive -Path {{path\to\example.zip}} -DestinationPath {{path\to\extracted_files}} -Force
```

- Preview without extracting:

```
Expand-Archive -Path {{path\to\example.zip}} -DestinationPath {{path\to\extracted_files}} -WhatIf
```

expand

Uncompress Windows Cabinet files.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/expand.

- Uncompress a single-file Cabinet file to the specified directory:

```
expand {{path\to\file.cab}} {{path\to\directory}}
```

- Display the list of files in a source Cabinet file:

```
expand {{path\to\file.cab}} {{path\to\directory}} -d
```

- Uncompress all files from the Cabinet file:

```
expand {{path\to\file.cab}} {{path\to\directory}} -f:*
```

- Uncompress a specific file from a Cabinet file:

```
expand {{path\to\file.cab}} {{path\to\directory}} -f:
{{path\to\file}}
```

- Ignore the directory structure when uncompressing, and add them to a single directory:

```
expand {{path\to\file.cab}} {{path\to\directory}} -i
```

explorer

The Windows File Explorer.

More information: https://ss64.com/nt/explorer.html.

- Open Windows Explorer:

```
explorer
```

- Open Windows Explorer in the current directory:

```
explorer .
```

- Open Windows Explorer in a specific directory:

```
explorer {{path\to\directory}}
```

fc

Compare the differences between two files or sets of files.

Use wildcards (*) to compare sets of files.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/fc.

- Compare 2 specified files:

```
fc {{path\to\file1}} {{path\to\file2}}
```

- Perform a case-insensitive comparison:

```
fc /c {{path\to\file1}} {{path\to\file2}}
```

- Compare files as Unicode text:

```
fc /u {{path\to\file1}} {{path\to\file2}}
```

- Compare files as ASCII text:

```
fc /l {{path\to\file1}} {{path\to\file2}}
```

- Compare files as binary:

```
fc /b {{path\to\file1}} {{path\to\file2}}
```

- Disable tab-to-space expansion:

```
fc /t {{path\to\file1}} {{path\to\file2}}
```

- Compress whitespace (tabs and spaces) for comparisons:

```
fc /w {{path\to\file1}} {{path\to\file2}}
```

find

Find a specified string in files.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/find.

- Find lines that contain a specified string:

```
find "{{string}}" {{path\to\file_or_directory}}
```

- Display lines that do not contain the specified string:

```
find "{{string}}" {{path\to\file_or_directory}} /v
```

- Display the count of lines that contain the specified string:

```
find "{{string}}" {{path\to\file_or_directory}} /c
```

- Display line numbers with the list of lines:

```
find "{{string}}" {{path\to\file_or_directory}} /n
```

findstr

Find specified text within one or more files.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/findstr.

- Find one or more strings in all files:

```
findstr "{{string1 string2 ...}}" *
```

- Find one or more strings in a piped command's output:

```
{{dir}} | findstr "{{string1 string2 ...}}"
```

- Find one or more strings in all files recur[s]ively:

```
findstr /s "{{string1 string2 ...}}" *
```

- Find strings using a case-[i]nsensitive search:

```
findstr /i "{{string1 string2 ...}}" *
```

- Find strings in all files using `regex`:

```
findstr /r "{{regex}}" *
```

- Find a literal string (containing spaces) in all text files:

```
findstr /c:"{{string1 string2 ...}}" *.txt
```

- Display the line [n]umber before each matching line:

```
findstr /n "{{string1 string2 ...}}" *
```

- Display only the filenames that contain a [m]atch:

```
findstr /m "{{string1 string2 ...}}" *
```

finger

Return information about users on a specified system.

The remote system must be running the Finger service.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/finger.

- Display information about a specific user:

```
finger {{user}}@{{host}}
```

- Display information about all users on the specified host:

```
finger @{{host}}
```

- Display information in a longer format:

```
finger {{user}}@{{host}} -l
```

- Display help:

```
finger /?
```

fondue

Install optional Windows features.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/fondue.

- Enable a specific Windows feature:

```
fondue /enable-feature:{{feature}}
```

- Hide all output messages to the user:

```
fondue /enable-feature:{{feature}} /hide-ux:all
```

- Specify a caller process name for error reporting:

```
fondue /enable-feature:{{feature}} /caller-name:{{name}}
```

for

Conditionally execute a command several times.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/for.

- Execute given commands for the specified set:

```
for %{{variable}} in ({{item_a item_b item_c}}) do ({{echo Loop is executed}})
```

- Iterate over a given range of numbers:

```
for /l %{{variable}} in ({{from}}, {{step}}, {{to}}) do ({{echo Loop is executed}})
```

- Iterate over a given list of files:

```
for %{{variable}} in ({{path\to\file1.ext path\to\file2.ext ...}}) do ({{echo Loop is executed}})
```

- Iterate over a given list of directories:

```
for /d %{{variable}} in ({{path\to\directory1.ext path\to\directory2.ext ...}}) do ({{echo Loop is executed}})
```

- Perform a given command in every directory:

```
for /d %{{variable}} in (*) do (if exist %{{variable}} {{echo Loop is executed}})
```

forfiles

Select files to execute a specified command on.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/forfiles.

- Search for files in the current directory:

```
forfiles
```

- Search for files in a specific directory:

```
forfiles /p {{path\to\directory}}
```

- Run the specified command for each file:

```
forfiles /c "{{command}}"
```

- Search for files using a specific glob mask:

```
forfiles /m {{glob_pattern}}
```

- Search for files recursively:

```
forfiles /s
```

- Search for files older than 5 days:

```
forfiles /d +{{5}}
```

fsutil

Display information about file system volumes.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/fsutil.

- Display a list of volumes:

```
fsutil volume list
```

- Display information about a volume's file system:

```
fsutil fsInfo volumeInfo {{drive_letter|volume_path}}
```

- Display the current state of the file system auto-repair for all volumes:

```
fsutil repair state
```

- Display the dirty bit state of all volumes:

```
fsutil dirty query
```

- Set the dirty bit state of a volume:

```
fsutil dirty set {{drive_letter|volume_path}}
```

ftp

Interactively transfer files between a local and remote FTP server.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/ftp.

- Connect to a remote FTP server interactively:

```
ftp {{host}}
```

- Log in as an anonymous user:

```
ftp -A {{host}}
```

- Disable automatic login upon initial connection:

```
ftp -n {{host}}
```

- Run a file containing a list of FTP commands:

```
ftp -s:{{path\to\file}} {{host}}
```

- Download multiple files (glob expression):

```
mget {{*.png}}
```

- Upload multiple files (glob expression):

```
mput {{*.zip}}
```

- Delete multiple files on the remote server:

```
mdelete {{*.txt}}
```

- Display help:

```
ftp --help
```

ftype

Display or modify file types used for file extension association.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/ftype.

- Display a list of all file types:

```
ftype
```

- Display the associated program for a specific file type:

```
ftype {{file_type}}
```

- Set the associated program for a specific file type:

```
ftype {{file_type}}="{{path/to/executable_file}}"
```

gal

In PowerShell, this command is an alias of **`Get-Alias`**.

- View documentation for the original command:

  ```
  tldr get-alias
  ```

gcb

This command is an alias of **`Get-Clipboard`**.

- View documentation for the original command:

  ```
  tldr get-clipboard
  ```

gcrane completion

Generate the autocompletion script for gcrane for the specified shell.

The available shells are Bash, fish, PowerShell, and Zsh.

More information: https://github.com/google/go-containerregistry/blob/main/cmd/gcrane/README.md.

- Generate the autocompletion script for your shell:

```
gcrane completion {{shell_name}}
```

- Disable completion descriptions:

```
gcrane completion {{shell_name}} --no-descriptions
```

- Load completions in your current shell session (PowerShell):

```
gcrane completion powershell | Out-String | Invoke-Expression
```

- Load completions for every new session (PowerShell):

```
gcrane completion powershell | Out-String | Invoke-Expression
```

- Display help:

```
gcrane completion {{shell_name}} {{[-h|--help]}}
```

Get-Acl

Get the security descriptor for a resource, such as a file or registry key.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.security/get-acl.

- Display the ACL for a specific directory:

```
Get-Acl {{path\to\directory}}
```

- Get an ACL for a registry key:

```
Get-Acl -Path {{HKLM:\System\CurrentControlSet\Control}} | Format-List
```

Get-Alias

List and get command aliases in the current PowerShell session.

This command can only be run under PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/get-alias.

- List all aliases in the current session:

  ```
  Get-Alias
  ```

- Get the aliased command name:

  ```
  Get-Alias {{command_alias}}
  ```

- List all aliases assigned to a specific command:

  ```
  Get-Alias -Definition {{command}}
  ```

- List aliases that begins with `abc`, excluding those which ends at `def`:

  ```
  Get-Alias {{abc}}* -Exclude *{{def}}
  ```

Get-ChildItem

List items in a directory.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/get-childitem.

- List all non-hidden items in the current directory:

```
Get-ChildItem
```

- List only directories in the current directory:

```
Get-ChildItem -Directory
```

- List only files in the current directory:

```
Get-ChildItem -File
```

- List items in the current directory, including hidden items:

```
Get-ChildItem -Hidden
```

- List items in a directory other than the current one:

```
Get-ChildItem -Path {{path\to\directory}}
```

Get-Clipboard

A PowerShell command to get content from clipboard.

Note: **`gcb`** can be used as an alias for **`Get-Clipboard`**.

More information: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-clipboard.

- Get clipboard text:

  ```
  Get-Clipboard
  ```

- Get clipboard content as specific text format:

  ```
  Get-Clipboard -TextFormatType {{Text|Html|Rtf}}
  ```

- Get raw clipboard content:

  ```
  Get-Clipboard -Raw
  ```

- Retrieve an Image:

  ```
  Get-Clipboard -Format Image
  ```

- Get file paths copied in explorer:

  ```
  Get-Clipboard -Format FileDropList
  ```

Get-Command

List and get available commands in the current PowerShell session.

This command can only be run through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/get-command.

- List all available PowerShell commands (aliases, cmdlets, functions) in the current computer:

  ```
  Get-Command
  ```

- List all available PowerShell commands in the current session:

  ```
  Get-Command -ListImported
  ```

- List only PowerShell aliases/cmdlets/functions available in the computer:

  ```
  Get-Command -Type {{Alias|Cmdlet|Function}}
  ```

- List only programs or commands available on PATH in the current session:

  ```
  Get-Command -Type Application
  ```

- List only PowerShell commands by the module name, e.g. `Microsoft.PowerShell.Utility` for utility-related commands:

  ```
  Get-Command -Module {{module}}
  ```

- Get the command information (e.g. version number or module name) by its name:

  ```
  Get-Command {{command}}
  ```

Get-Content

Get the content of the item at the specified location.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/get-content.

- Display the content of a file:

```
Get-Content -Path {{path\to\file}}
```

- Display the first few lines of a file:

```
Get-Content -Path {{path\to\file}} -TotalCount {{10}}
```

- Display the content of the file and keep reading from it until `<Ctrl c>` is pressed:

```
Get-Content -Path {{path\to\file}} -Wait
```

Get-Date

Get the current date and time.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/get-date.

- Display the current date and time:

  ```
  Get-Date
  ```

- Display the current date and time with a .NET format specifier:

  ```
  Get-Date -Format "{{yyyy-MM-dd HH:mm:ss}}"
  ```

- Display the current date and time in UTC and ISO 8601 format:

  ```
  (Get-Date).ToUniversalTime()
  ```

- Convert a Unix timestamp:

  ```
  Get-Date -UnixTimeSeconds {{1577836800}}
  ```

Get-DedupProperties

Get Data Deduplication information.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/storage/get-deduppproperties.

- Get Data Deduplication information of the drive:

```
Get-DedupProperties -DriveLetter 'C'
```

- Get Data Deduplication information of the drive using the drive label:

```
Get-DedupProperties -FileSystemLabel 'Label'
```

- Get Data Dedpulication information of the drive using the input object:

```
Get-DedupProperties -InputObject $(Get-Volume -DriveLetter
'E')
```

Get-FileHash

Calculate a hash for a file.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/get-filehash.

- Calculate a hash for a specified file using the SHA256 algorithm:

  ```
  Get-FileHash {{path\to\file}}
  ```

- Calculate a hash for a specified file using a specified algorithm:

  ```
  Get-FileHash {{path\to\file}} -Algorithm {{SHA1|SHA384|SHA256|SHA512|MD5}}
  ```

Get-Help

Display help information and documentation for PowerShell commands (aliases, cmdlets, and functions).

This command can only be run through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/get-help.

- Display general help information for a specific PowerShell command:

  ```
  Get-Help {{command}}
  ```

- Display a more detailed documentation for a specific PowerShell command:

  ```
  Get-Help {{command}} -Detailed
  ```

- Display the full technical documentation for a specific PowerShell command:

  ```
  Get-Help {{command}} -Full
  ```

- Print only the documentation for a specific parameter of the PowerShell command (use `*` to show all parameters), if available:

  ```
  Get-Help {{command}} -Parameter {{parameter}}
  ```

- Print only the examples of the cmdlet, if available:

  ```
  Get-Help {{command}} -Examples
  ```

- List all available cmdlet help pages:

  ```
  Get-Help *
  ```

- Update the current help and documentation knowledge base using `Update-Help`:

  ```
  Update-Help
  ```

- View an online version of PowerShell command documentation in the default web browser:

  ```
  Get-Help {{command}} -Online
  ```

Get-History

Display PowerShell command history.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/get-history.

- Display the commands history list with ID:

```
Get-History
```

- Get PowerShell history item by ID:

```
Get-History -Id {{id}}
```

- Display the last n commands:

```
Get-History -Count {{n}}
```

Get-Location

Print name of current/working directory.

This command can only be run through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/get-location.

- Print the current directory:

```
Get-Location
```

Get-WUApiVersion

Get the Windows Update Agent version. Part of external **PSWindowsUpdate** module.

This command can only be run under PowerShell.

More information: https://github.com/mgajda83/PSWindowsUpdate.

- Get the currently-installed Windows Update Agent version:

```
Get-WUApiVersion
```

- Send the current configuration data via email (SMTP):

```
Get-WUApiVersion -SendReport -PSWUSettings
@{SmtpServer="{{smtp_server}}"; Port={{smtp_port}}
From="{{sender_email}}" To="{{receiver_email}}"}
```

Get-WUHistory

Get the history of installed updates from Windows Update. Part of external **`PSWindowsUpdate`** module.

This command can only be run under PowerShell.

More information: https://github.com/mgajda83/PSWindowsUpdate.

- Get list of update history:

```
Get-WUHistory
```

- List the last 10 installed updates:

```
Get-WUHistory -Last {{10}}
```

- List all updates installed from a specific date to today:

```
Get-WUHistory -MaxDate {{date}}
```

- List all updates installed in the past 24 hours:

```
Get-WUHistory -MaxDate (Get-Date).AddDays(-1)
```

- Send the results via email (SMTP):

```
Get-WUHistory -SendReport -PSWUSettings
@{SmtpServer="{{smtp_server}}"; Port={{smtp_port}}
From="{{sender_email}}" To="{{receiver_email}}"}
```

Get-WUSettings

Get the current Windows Update Agent configuration. Part of external **PSWindowsUpdate** module.

This command can only be run under PowerShell.

More information: https://github.com/mgajda83/PSWindowsUpdate.

- Get the current Windows Update Agent configuration:

```
Get-WUSettings
```

- Send the current configuration data via email (SMTP):

```
Get-WUSettings -SendReport -PSWUSettings
@{SmtpServer="{{smtp_server}}"; Port={{smtp_port}}
From="{{sender_email}}" To="{{receiver_email}}"}
```

getmac

Display the MAC addresses of a system.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/getmac.

- Display the MAC addresses for the current system:

```
getmac
```

- Display the details in a specific format:

```
getmac /fo {{table|list|csv}}
```

- Exclude the header in the output list:

```
getmac /nh
```

- Display the MAC addresses for a remote machine:

```
getmac /s {{hostname}} /u {{username}} /p {{password}}
```

- Display the MAC addresses with verbose information:

```
getmac /v
```

- Display help:

```
getmac /?
```

ghy

This command is an alias of **`Get-History`**.

- View documentation for the original command:

  ```
  tldr Get-History
  ```

gl

In PowerShell, this command is an alias of **`Get-Location`**.

- View documentation for the original command:

```
tldr get-location
```

gpupdate

Check and apply Windows Group Policy settings.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/gpupdate.

- Check and apply updated Group Policy settings:

```
gpupdate
```

- Specify the target Group Policy settings to check for update:

```
gpupdate /target:{{computer|user}}
```

- Force all Group Policy settings to be reapplied:

```
gpupdate /force
```

- Display help:

```
gpupdate /?
```

h

This command is an alias of **`Get-History`**.

- View documentation for the original command:

```
tldr Get-History
```

history

This command is an alias of **`Get-History`**.

- View documentation for the original command:

```
tldr Get-History
```

if

Performs conditional processing in batch scripts.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/if.

- Execute the specified commands if the condition is true:

```
if {{condition}} ({{echo Condition is true}})
```

- Execute the specified commands if the condition is false:

```
if not {{condition}} ({{echo Condition is true}})
```

- Execute the first specified commands if the condition is true otherwise execute the second specified commands:

```
if {{condition}} ({{echo Condition is true}}) else ({{echo Condition is false}})
```

- Check whether `%errorlevel%` is greater than or equal to the specified exit code:

```
if errorlevel {{2}} ({{echo Condition is true}})
```

- Check whether two strings are equal:

```
if %{{variable}}% == {{string}} ({{echo Condition is true}})
```

- Check whether two strings are equal without respecting letter case:

```
if /i %{{variable}}% == {{string}} ({{echo Condition is true}})
```

- Check whether a file exist:

```
if exist {{path\to\file}} ({{echo Condition is true}})
```

Install-Module

Install PowerShell modules from PowerShell Gallery, NuGet, and other repositories.

More information: https://learn.microsoft.com/powershell/module/powershellget/install-module.

- Install a module, or update it to the latest available version:

```
Install-Module {{module}}
```

- Install a module with a specific version:

```
Install-Module {{module}} -RequiredVersion {{version}}
```

- Install a module no earlier than a specific version:

```
Install-Module {{module}} -MinimumVersion {{version}}
```

- Specify a range of supported versions (inclusive) of the required module:

```
Install-Module {{module}} -MinimumVersion {{minimum_version}} -MaximumVersion {{maximum_version}}
```

- Install module from a specific repository:

```
Install-Module {{module}} -Repository {{repository}}
```

- Install module from specific repositories:

```
Install-Module {{module}} -Repository {{repository1 , repository2 , ...}}
```

- Install the module for all/current user:

```
Install-Module {{module}} -Scope {{AllUsers|CurrentUser}}
```

- Perform a dry run to determine which modules will be installed, upgraded, or removed through `Install-Module`:

```
Install-Module {{module}} -WhatIf
```

Invoke-Item

Open files in their respective default programs.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/invoke-item.

- Open a file in its default program:

```
Invoke-Item -Path {{path\to\file}}
```

- Open all files inside a directory:

```
Invoke-Item -Path {{path\to\directory}}\*
```

- Open all PNGs inside a directory:

```
Invoke-Item -Path {{path\to\directory}}\*.png
```

- Open all files inside a directory containing a specific keyword:

```
Invoke-Item -Path {{path\to\directory}}\* -Include {{*keyword*}}
```

- Open all files inside a directory except those containing a specific keyword:

```
Invoke-Item -Path {{path\to\directory}}\* -Exclude {{*keyword*}}
```

- Perform a dry run to determine which files will be opened inside a directory through `Invoke-Item`:

```
Invoke-Item -Path {{path\to\directory}}\* -WhatIf
```

Invoke-WebRequest

Performs a HTTP/HTTPS request to the Web.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/invoke-webrequest.

- Download the contents of a URL to a file:

```
Invoke-WebRequest {{http://example.com}} -OutFile {{path\to\file}}
```

- Send form-encoded data (POST request of type `application/x-www-form-urlencoded`):

```
Invoke-WebRequest -Method Post -Body @{ name='bob' } {{http://example.com/form}}
```

- Send a request with an extra header, using a custom HTTP method:

```
Invoke-WebRequest -Headers {{@{ X-My-Header = '123' }}} -Method {{PUT}} {{http://example.com}}
```

- Send data in JSON format, specifying the appropriate content-type header:

```
Invoke-WebRequest -Body '{{{"name":"bob"}}}' -ContentType 'application/json' {{http://example.com/users/1234}}
```

- Pass a username and password for server authentication:

```
Invoke-WebRequest -Headers @{ Authorization = "Basic "+ [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetByte
{{http://example.com}}
```

ipconfig

Display and manage the network configuration of Windows.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/ipconfig.

- List all network adapters:

```
ipconfig
```

- Show a detailed list of network adapters:

```
ipconfig /all
```

- Renew the IP addresses for a network adapter:

```
ipconfig /renew {{adapter}}
```

- Free up the IP addresses for a network adapter:

```
ipconfig /release {{adapter}}
```

- Show the local DNS cache:

```
ipconfig /displaydns
```

- Remove all data from the local DNS cache:

```
ipconfig /flushdns
```

iscc

Compiler for Inno Setup installers.

It compiles an Inno Setup scripts into an Windows installer executable.

More information: https://jrsoftware.org/isinfo.php.

- Compile an Inno Setup script:

```
iscc {{path\to\file.iss}}
```

- Quietly compile an Inno Setup installer:

```
iscc /Q {{path\to\file.iss}}
```

- Compile a signed Inno Setup installer:

```
iscc /S={{name}}={{command}} {{path\to\file.iss}}
```

iwr

In PowerShell, this command is an alias of **`Invoke-WebRequest`**.

- View documentation for the original command:

```
tldr invoke-webrequest
```

lib

The Microsoft Library Manager for creating and managing static libraries of object files.

More information: https://learn.microsoft.com/cpp/build/reference/lib-reference.

- Create a static library from object files:

```
lib /OUT :{{path/to/library.lib}} {{path/to/file1.obj path/to/file2.obj ...}}
```

- List the contents of a library:

```
lib /LIST {{path/to/library.lib}}
```

- Add an object file to an existing library:

```
lib {{path/to/library.lib}} {{path/to/file.obj}}
```

- Remove an object file from a library:

```
lib /REMOVE :{{path/to/file.obj}} {{path/to/library.lib}}
```

- Extract an object file from a library:

```
lib /EXTRACT :{{path/to/file.obj}} {{path/to/library.lib}}
```

- Create an import library from a DLL:

```
lib /DEF :{{path/to/definition.def}} /OUT:{{path/to/import.lib}}
```

logoff

Terminate a login session.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/logoff.

- Terminate the current session:

```
logoff
```

- Terminate a session by its name or ID:

```
logoff {{session_name|session_id}}
```

- Terminate a session on a specific server connected through RDP:

```
logoff {{session_name|session_id}} /server:{{servername}}
```

Measure-Command

Measures the time it takes to run script blocks and cmdlets.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/measure-command.

- Measure the time it takes to run a command:

  ```
  Measure-Command { {{command}} }
  ```

- Pipe input to Measure-Command (objects that are piped to `Measure-Command` are available to the script block that is passed to the Expression parameter):

  ```
  10, 20, 50 | Measure-Command -Expression { for ($i=0; $i -lt $_; $i++) {$i} }
  ```

Measure-Object

Calculates the numeric properties of objects, and the characters, words, and lines in string objects, such as files of text.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/measure-object.

- Count the files and folders in a directory:

```
Get-ChildItem | Measure-Object
```

- Pipe input to Measure-Command (objects that are piped to `Measure-Command` are available to the script block that is passed to the Expression parameter):

```
"One", "Two", "Three", "Four" | Set-Content -Path
"{{path\to\file}}"; Get-Content "{{path\to\file}}"; |
Measure-Object -Character -Line -Word
```

mi

In PowerShell, this command is an alias of **`Move-Item`**.

- View documentation for the original command:

```
tldr move-item
```

microsoft-edge

The Microsoft Edge command-line utility is available as **`msedge`** on Windows and **`microsoft-edge`** for other platforms.

More information: https://microsoft.com/edge.

- View the documentation for Microsoft Edge for Windows:

```
tldr {{[-p|--platform]}} windows msedge
```

- View the documentation for Microsoft Edge for other platforms:

```
tldr {{[-p|--platform]}} common microsoft-edge
```

mimikatz crypto

Manipulate Windows cryptographic services and certificates.

More information: https://github.com/gentilkiwi/mimikatz.

- List cryptographic providers:

```
mimikatz "crypto::providers"
```

- List keys in a cryptographic provider:

```
mimikatz "crypto::capi"
```

- Export certificates and keys:

```
mimikatz "crypto::certificates /export"
```

mimikatz dpapi

Interact with the Windows Data Protection API (DPAPI).

More information: https://github.com/gentilkiwi/mimikatz.

- List master keys:

```
mimikatz "dpapi::masterkey /list"
```

- Decrypt a DPAPI blob:

```
mimikatz "dpapi::blob /in:blob_file.bin"
```

- Retrieve Chrome credentials using DPAPI:

```
mimikatz "dpapi::chrome /in:Login Data"
```

mimikatz event

Manage Windows Event Log records.

More information: https://github.com/gentilkiwi/mimikatz.

- Clear event logs:

```
mimikatz "event::clear"
```

- Display event log sources:

```
mimikatz "event::providers"
```

mimikatz kerberos

Interact with Kerberos tickets.

More information: https://github.com/gentilkiwi/mimikatz.

- List current Kerberos tickets:

```
mimikatz "kerberos::list"
```

- Purge all Kerberos tickets:

```
mimikatz "kerberos::purge"
```

- Inject a ticket from a `.kirbi` file:

```
mimikatz "kerberos::ptt ticket.kirbi"
```

mimikatz lsadump

Dump secrets from the Windows Local Security Authority (LSA).

Requires SYSTEM privileges.

More information: https://github.com/gentilkiwi/mimikatz.

- Dump SAM hashes:

```
mimikatz "lsadump::sam"
```

- Dump secrets from the SECURITY hive:

```
mimikatz "lsadump::secrets"
```

- Dump cached domain credentials:

```
mimikatz "lsadump::cache"
```

mimikatz misc

Miscellaneous system and utility commands.

More information: https://github.com/gentilkiwi/mimikatz.

- Trigger Blue Screen of Death (for testing):

```
mimikatz "misc::bsod"
```

- List loaded kernel drivers:

```
mimikatz "misc::minidump"
```

- Change system time:

```
mimikatz "misc::systemtime"
```

mimikatz net

Perform network and domain operations.

More information: https://github.com/gentilkiwi/mimikatz.

- List domain users:

```
mimikatz "net::users"
```

- List domain computers:

```
mimikatz "net::computers"
```

- Retrieve domain controller information:

```
mimikatz "net::domaincontrollers"
```

mimikatz privilege

Manage privileges for mimikatz operations.

More information: https://github.com/gentilkiwi/mimikatz.

- Enable debug privilege (required for many modules):

```
mimikatz "privilege::debug"
```

- Check current privilege state:

```
mimikatz "privilege::whoami"
```

mimikatz process

Manage process privileges and tokens.

More information: https://github.com/gentilkiwi/mimikatz.

- List processes with their tokens:

```
mimikatz "process::list"
```

- Elevate mimikatz to a SYSTEM process:

```
mimikatz "process::token /user:NT AUTHORITY\SYSTEM"
```

mimikatz sekurlsa

Extract credentials and secrets from memory.

Requires debug privileges.

More information: https://github.com/gentilkiwi/mimikatz.

- Extract plaintext passwords:

```
mimikatz "sekurlsa::logonpasswords"
```

- List Kerberos tickets in memory:

```
mimikatz "sekurlsa::tickets"
```

- Dump LSA secrets:

```
mimikatz "sekurlsa::secrets"
```

mimikatz service

Manage Windows services through mimikatz.

More information: https://github.com/gentilkiwi/mimikatz.

- Start a service:

```
mimikatz "service::start service_name"
```

- Stop a service:

```
mimikatz "service::stop service_name"
```

- Delete a service:

```
mimikatz "service::delete service_name"
```

mimikatz standard

Basic commands and mimikatz environment management.

More information: https://github.com/gentilkiwi/mimikatz.

- Display system information:

```
mimikatz "standard::info"
```

- Clear the mimikatz command history:

```
mimikatz "standard::clearev"
```

- Show command history:

```
mimikatz "standard::history"
```

mimikatz token

List and manipulate security tokens.

More information: https://github.com/gentilkiwi/mimikatz.

- List tokens:

```
mimikatz "token::list"
```

- Elevate privileges by impersonating a token:

```
mimikatz "token::elevate"
```

- Revert to original token:

```
mimikatz "token::revert"
```

mimikatz vault

Extract credentials stored in the Windows Credential Vault.

More information: https://github.com/gentilkiwi/mimikatz.

- List vault credentials:

```
mimikatz "vault::list"
```

- Dump all vault credentials:

```
mimikatz "vault::cred"
```

mimikatz

Interact with Windows credentials, perform credential dumping, token manipulation, and more.

Requires administrator privileges and typically runs on Windows.

More information: https://github.com/gentilkiwi/mimikatz.

- Run mimikatz in interactive mode:

```
mimikatz
```

- Enable debug privileges (needed for most operations):

```
mimikatz "privilege::debug"
```

- List available logon sessions:

```
mimikatz "sekurlsa::logonpasswords"
```

- Dump plaintext passwords, NTLM hashes, and Kerberos tickets from memory:

```
mimikatz "sekurlsa::logonpasswords"
```

- Pass-the-Hash with a specific NTLM hash and launch a command:

```
mimikatz "sekurlsa::pth /user:{{username}} /domain:
{{domain}} /ntlm:{{hash}} /run:{{cmd}}"
```

- Dump local SAM database hashes:

```
mimikatz "lsadump::sam"
```

- Extract Kerberos tickets and export to a file:

```
mimikatz "kerberos::list /export"
```

- Exit mimikatz:

```
exit
```

mkdir

Create a directory.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/mkdir.

- Create a directory:

```
mkdir {{path\to\directory}}
```

- Create a nested directory tree recursively:

```
mkdir {{path\to\sub_directory}}
```

mklink

Create symbolic links.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/mklink.

- Create a symbolic link to a file:

```
mklink {{path\to\link_file}} {{path\to\source_file}}
```

- Create a symbolic link to a directory:

```
mklink /d {{path\to\link_file}} {{path\to\source_directory}}
```

- Create a hard link to a file:

```
mklink /h {{path\to\link_file}} {{path\to\source_file}}
```

- Create a directory junction:

```
mklink /j {{path\to\link_file}} {{path\to\source_file}}
```

more

Display paginated output from **`stdin`** or a file.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/more.

- Display paginated output from `stdin`:

```
{{echo test}} | more
```

- Display paginated output from one or more files:

```
more {{path\to\file}}
```

- Convert tabs to the specified number of spaces:

```
more {{path\to\file}} /t{{spaces}}
```

- Clear the screen before displaying the page:

```
more {{path\to\file}} /c
```

- Display the output starting at line 5:

```
more {{path\to\file}} +{{5}}
```

- Enable extended interactive mode (see help for usage):

```
more {{path\to\file}} /e
```

- Display help:

```
more /?
```

mount

Mount Network File System (NFS) network shares.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/mount.

- Mount a share to the "Z" drive letter:

```
mount \\{{computer_name}}\{{share_name}} {{Z:}}
```

- Mount a share to the next available drive letter:

```
mount \\{{computer_name}}\{{share_name}} *
```

- Mount a share with a read timeout in seconds (defaults to 0.8, can be 0.9 or 1 to 60):

```
mount -o timeout={{seconds}} \\{{computer_name}}\
{{share_name}} {{Z:}}
```

- Mount a share and retry up to 10 times if it fails:

```
mount -o retry=10 \\{{computer_name}}\{{share_name}} {{Z:}}
```

- Mount a share with forced case sensitivity:

```
mount -o casesensitive \\{{computer_name}}\{{share_name}}
{{Z:}}
```

- Mount a share as an anonymous user:

```
mount -o anon \\{{computer_name}}\{{share_name}} {{Z:}}
```

- Mount a share using a specific mount type:

```
mount -o mtype={{soft|hard}} \\{{computer_name}}\
{{share_name}} {{Z:}}
```

Move-Item

Move or rename files, directories, registry keys, and other PowerShell data items.

This command can only be run through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/move-item.

- Rename a file or directory when the target is not an existing directory:

```
Move-Item {{path\to\source}} {{path\to\target}}
```

- Move a file or directory into an existing directory:

```
Move-Item {{path\to\source}} {{path\to\existing_directory}}
```

- Rename or move file(s) with specific name (do not treat special characters inside strings):

```
Move-Item -LiteralPath
"{{path\to\source}}" {{path\to\file_or_directory}}
```

- Move multiple files into an existing directory, keeping the filenames unchanged:

```
Move-Item {{path\to\source1 , path\to\source2 ...}}
{{path\to\existing_directory}}
```

- Move or rename registry key(s):

```
Move-Item {{path\to\source_key1 , path\to\source_key2 ...}}
{{path\to\new_or_existing_key}}
```

- Do not prompt for confirmation before overwriting existing files or registry keys:

```
mv -Force {{path\to\source}} {{path\to\target}}
```

- Prompt for confirmation before overwriting existing files, regardless of file permissions:

```
mv -Confirm {{path\to\source}} {{path\to\target}}
```

- Move files in dry-run mode, showing files and directories which could be moved without executing them:

```
mv -WhatIf {{path\to\source}} {{path\to\target}}
```

move

Move or rename files and directories.

In PowerShell, this command is an alias of **`Move-Item`**. This documentation is based on the Command Prompt (**`cmd`**) version of **`move`**.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/move.

- View documentation of the equivalent PowerShell command:

```
tldr move-item
```

- Rename a file or directory when the target is not an existing directory:

```
move {{path\to\source}} {{path\to\target}}
```

- Move a file or directory into an existing directory:

```
move {{path\to\source}} {{path\to\existing_directory}}
```

- Move a file or directory across drives:

```
move {{C:\path\to\source}} {{D:\path\to\target}}
```

- Do not prompt for confirmation before overwriting existing files:

```
move /Y {{path\to\source}} {{path\to\existing_directory}}
```

- Prompt for confirmation before overwriting existing files, regardless of file permissions:

```
move /-Y {{path\to\source}} {{path\to\existing_directory}}
```

msedge

Modern web browser developed by Microsoft based on the Chromium web browser developed by Google.

This command is available instead as **`microsoft-edge`** for other platforms.

Note: Additional command arguments from **`chromium`** may also be usable to control Microsoft Edge.

More information: https://microsoft.com/edge.

- Open a specific URL or file:

```
msedge {{https://example.com|path\to\file.html}}
```

- Open in InPrivate mode:

```
msedge --inprivate {{example.com}}
```

- Open in a new window:

```
msedge --new-window {{example.com}}
```

- Open in application mode (without toolbars, URL bar, buttons, etc.):

```
msedge --app {{https://example.com}}
```

- Use a proxy server:

```
msedge --proxy-server "{{socks5://hostname:
66}}" {{example.com}}
```

- Open with a custom profile directory:

```
msedge --user-data-dir {{path\to\directory}}
```

- Open without CORS validation (useful to test an API):

```
msedge --user-data-dir {{path\to\directory}} --disable-web-
security
```

- Open with a DevTools window for each tab opened:

```
msedge --auto-open-devtools-for-tabs
```

msg

Send a message to a user or session.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/msg.

- Send a message to a specified user or session:

```
msg {{username|session_name|session_id}} {{message}}
```

- Send a message from `stdin`:

```
echo "{{message}}" | msg {{username|session_name|session_id}}
```

- Send a message to a specific server:

```
msg /server:{{server_name}} {{username|session_name|
session_id}}
```

- Send a message to all users of the current machine:

```
msg *
```

- Set a delay in seconds for a message:

```
msg /time:{{10}}
```

msiexec

Install, update, repair, or uninstall Windows programs using MSI and MSP package files.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/msiexec.

- Install a program from its MSI package:

```
msiexec /package {{path\to\file.msi}}
```

- Install a MSI package from a website:

```
msiexec /package {{https://example.com/installer.msi}}
```

- Install a MSP patch file:

```
msiexec /update {{path\to\file.msp}}
```

- Uninstall a program or patch using their respective MSI or MSP file:

```
msiexec /uninstall {{path\to\file}}
```

mv

In PowerShell, this command is an alias of **`Move-Item`**.

However, this command is not available on the Command Prompt (**`cmd`**). Use **`move`** instead for similar functionality.

- View documentation for the equivalent Command Prompt command:

  ```
  tldr move
  ```

- View documentation for the original PowerShell command:

  ```
  tldr move-item
  ```

net

System utility to view and modify network-related settings.

More information: https://learn.microsoft.com/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/gg651155(v=ws.11).

- Start or stop a Windows service synchronously:

```
net {{start|stop}} {{service}}
```

- Make sure an SMB share is available in the current console:

```
net use {{\\smb_shared_folder}} /USER:{{username}}
```

- Show the folders currently shared over SMB:

```
net share
```

- Show who is using your SMB shares (run in elevated console):

```
net session
```

- Show users in a local security group:

```
net localgroup "{{Administrators}}"
```

- Add a user to the local security group (run in elevated console):

```
net localgroup "{{Administrators}}" {{username}} /add
```

- Display help for a subcommand:

```
net help {{subcommand}}
```

- Display help:

```
net help
```

netsh interface portproxy

Configure and display the status of various network components.

More information: https://learn.microsoft.com/windows-server/networking/technologies/netsh/netsh-interface-portproxy.

- Display the current port forwarding setup:

```
netsh interface portproxy show all
```

- Set up IPv4 port forwarding (run in elevated console):

```
netsh interface portproxy add v4tov4
listenaddress={{192.168.0.1}} listenport={{8080}}
connectaddress={{10.0.0.1}} connectport={{80}}
```

- Remove IPv4 port forwarding (run in elevated console):

```
netsh interface portproxy delete v4tov4
listenaddress={{192.168.0.1}} listenport={{8080}}
```

- Display help:

```
netsh interface portproxy
```

netsh wlan

Manage wireless networks.

More information: https://www.serverwatch.com/guides/netsh-commands/.

- Show all available wireless networks:

```
netsh wlan show networks
```

- Connect to a wireless network with a specific SSID:

```
netsh wlan connect name={{SSID}}
```

- Disconnect from the current wireless network:

```
netsh wlan disconnect
```

- Show current wireless network interfaces and status:

```
netsh wlan show interfaces
```

- Export a wireless network profile to an XML file:

```
netsh wlan export profile name={{SSID}} folder={{C:\path\to\folder}} key=clear
```

- Delete a saved wireless network profile:

```
netsh wlan delete profile name={{SSID}}
```

- Enable hosted network (turn PC into Wi-Fi hotspot):

```
netsh wlan set hostednetwork mode=allow ssid={{SSID}} key={{password}}
```

- Start the hosted network:

```
netsh wlan start hostednetwork
```

netsh

Manage Windows network settings.

Some subcommands such as **`wlan`** have their own usage documentation.

More information: https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/netsh.

- Add a helper Dynamic Link Library (DLL):

```
netsh add helper {{path\to\file.dll}}
```

- Show all loaded helper DLLs:

```
netsh show helper
```

- Delete a helper DLL:

```
netsh delete helper {{path\to\file.dll}}
```

- Export your network configuration settings to a file:

```
netsh dump > {{path\to\output_file.txt}}
```

- Show available network interfaces for tracing:

```
netsh trace show interfaces
```

- Exit the shell:

```
exit
```

- Display help:

```
netsh help
```

netstat

Display active TCP connections, ports on which the computer is listening, network adapter statistics, the IP routing table, IPv4 statistics and IPv6 statistics.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/netstat.

- Display active TCP connections:

```
netstat
```

- Display all active TCP connections and the TCP and UDP ports on which the computer is listening:

```
netstat -a
```

- Display network adapter statistics, such as the number of bytes and packets sent and received:

```
netstat -e
```

- Display active TCP connections and express addresses and port numbers numerically:

```
netstat -n
```

- Display active TCP connections and include the process ID (PID) for each connection:

```
netstat -o
```

- Display the contents of the IP routing table:

```
netstat -r
```

- Display statistics by protocol:

```
netstat -s
```

- Display a list of currently open ports and related IP addresses:

```
netstat -an
```

New-Item

Create a new file, directory, symbolic link, or a registry entry.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/new-item.

- Create a new blank file (equivalent to `touch`):

```
New-Item {{path\to\file}}
```

- Create a new directory:

```
New-Item -ItemType Directory {{path\to\directory}}
```

- Write a new text file with specified content:

```
New-Item {{path\to\file}} -Value {{content}}
```

- Write the same text file in multiple locations:

```
New-Item {{path\to\file1 , path\to\file2 , ...}} -Value
{{content}}
```

- Create a symbolic link\hard link\junction to a file or directory:

```
New-Item -ItemType {{SymbolicLink|HardLink|Junction}} -Path
{{path\to\link_file}} -Target
{{path\to\source_file_or_directory}}
```

- Create a new blank registry entry (in REG_SZ, use `New-ItemProperty` or `Set-ItemProperty` to fine-tune the value type):

```
New-Item {{path\to\registry_key}}
```

- Create a new blank registry entry with specified value:

```
New-Item {{path\to\registry_key}} -Value {{value}}
```

nfsstat

Display or reset the number of calls made to the NFS server.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/nfsstat.

- Display the recorded number of calls made to the NFS server:

```
nfsstat
```

- Reset the recorded number of calls made to the NFS server:

```
nfsstat -z
```

ni

In PowerShell, this command is an alias of **`New-Item`**.

- View documentation for the original command:

```
tldr new-item
```

nmake

The Microsoft Program Maintenance Utility for building projects based on commands in a makefile.

More information: https://learn.microsoft.com/cpp/build/reference/nmake-reference.

- Build targets using the default makefile in the current directory:

```
nmake
```

- Build targets using a specific makefile:

```
nmake /F {{path/to/makefile}}
```

- Build a specific target:

```
nmake {{target}}
```

- Display commands without executing them:

```
nmake /N
```

- Display all macro definitions and target descriptions:

```
nmake /P
```

- Continue building unrelated targets on error:

```
nmake /K
```

- Build and ignore timestamp checks (force rebuild):

```
nmake /A
```

- Suppress copyright message:

```
nmake /NOLOGO
```

nvm

Install, uninstall, or switch between Node.js versions.

Supports version numbers like "12.8" or "v16.13.1", and labels like "stable", "system", etc.

More information: https://github.com/coreybutler/nvm-windows.

- Install a specific version of Node.js:

```
nvm install {{node_version}}
```

- Set the default version of Node.js (must be run as Administrator):

```
nvm use {{node_version}}
```

- List all available Node.js versions and highlight the default one:

```
nvm list
```

- List all remote Node.js versions:

```
nvm ls-remote
```

- Uninstall a given Node.js version:

```
nvm uninstall {{node_version}}
```

octo

Command-line tools for Octopus Deploy.

More information: https://octopus.com/docs/octopus-rest-api/octo.exe-command-line.

- Create a package:

```
octo pack --id {{package}}
```

- Push a package to a repository on the Octopus server:

```
octo push --package {{package}}
```

- Create a release:

```
octo create-release --project {{project_name}} --
packageversion {{version}}
```

- Deploy a release:

```
octo deploy-release --project {{project_name}} --
packageversion {{version}} --deployto {{environment_name}} --
tenant {{deployment_target}}
```

ospp.vbs

Install, activate, and manage volume licensed versions of Microsoft Office products.

Note: This command may override, deactivate, and/or remove your current volume of licensed Office product versions, so please proceed cautiously.

More information: https://learn.microsoft.com/deployoffice/vlactivation/tools-to-manage-volume-activation-of-office.

- Install a product key (Note: It replaces the existing key):

  ```
  cscript ospp.vbs /inpkey:{{product_key}}
  ```

- Uninstall an installed product key with the last five digits of the product key:

  ```
  cscript ospp.vbs /unpkey:{{product_key_digits}}
  ```

- Set a KMS host name:

  ```
  cscript ospp.vbs /sethst:{{ip|hostname}}
  ```

- Set a KMS port:

  ```
  cscript ospp.vbs /setprt:{{port}}
  ```

- Activate installed Office product keys:

  ```
  cscript ospp.vbs /act
  ```

- Display license information for installed product keys:

  ```
  cscript ospp.vbs /dstatus
  ```

Out-String

Outputs input objects as a string.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/out-string.

- Print host information as string:

```
Get-Alias | Out-String
```

- Convert each object to a string rather than concatenating all the objects into a single string:

```
Get-Alias | Out-String -Stream
```

- Use the `Width` parameter to prevent truncation:

```
@{TestKey = ('x' * 200)} | Out-String -Width {{250}}
```

pabcnetcclear

Preprocess and compile PascalABC.NET source files.

More information: https://pascalabc.net.

- Compile the specified source file into an executable with the same name:

```
pabcnetcclear {{path\to\source_file.pas}}
```

- Compile the specified source file into an executable with the specified name:

```
pabcnetcclear /Output:{{path\to\_file.exe}}
{{path\to\source_file.pas}}
```

- Compile the specified source file into an executable with the same name along with/without debug information:

```
pabcnetcclear /Debug:{{0|1}} {{path\to\source_file.pas}}
```

- Allow units to be searched in the specified path while compiling the source file into an executable with the same name:

```
pabcnetcclear /SearchDir:{{path\to\directory}}
{{path\to\source_file.pas}}
```

- Compile the specified source file into an executable, defining a symbol:

```
pabcnetcclear /Define:{{symbol}} {{path\to\source_file.pas}}
```

path

Display or set the search path for executable files.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/path.

- Display the current path:

```
path
```

- Set the path to one or more semicolon-separated directories:

```
path {{path\to\directory1 path\to\directory2 ...}}
```

- Append a new directory to the original path:

```
path {{path\to\directory}};%path%
```

- Set command prompt to only search the current directory for executables:

```
path ;
```

pathping

A trace route tool combining features of **`ping`** and **`tracert`**.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/pathping.

- Ping and trace the route to a host:

```
pathping {{hostname}}
```

- Do not perform reverse lookup of IP address to hostname:

```
pathping {{hostname}} -n
```

- Specify the maximum number of hops to search for the target (the default is 30):

```
pathping {{hostname}} -h {{max_hops}}
```

- Specify the milliseconds to wait between pings (the default is 240):

```
pathping {{hostname}} -p {{time}}
```

- Specify the number of queries per hop (the default is 100):

```
pathping {{hostname}} -q {{queries}}
```

- Force IPV4 usage:

```
pathping {{hostname}} -4
```

- Force IPV6 usage:

```
pathping {{hostname}} -6
```

- Display help:

```
pathping /?
```

pipwin

A tool to install unofficial Python package binaries on Windows.

More information: https://github.com/lepisma/pipwin.

- List all available packages for download:

```
pipwin list
```

- Search packages:

```
pipwin search {{partial_name|name}}
```

- Install a package:

```
pipwin install {{package}}
```

- Uninstall a package:

```
pipwin uninstall {{package}}
```

- Download a package to a specific directory:

```
pipwin download --dest {{path\to\directory}} {{package}}
```

- Install packages according to `requirements.txt`:

```
pipwin install --file {{path\to\requirements.txt}}
```

pnputil

Manage the driver store and driver packages on Windows.

Requires an elevated Command Prompt.

More information: https://learn.microsoft.com/windows-hardware/drivers/devtest/pnputil-command-syntax.

- List all installed driver packages:

```
pnputil /enum-drivers
```

- Install a driver package from an INF file:

```
pnputil /add-driver {{path\to\driver.inf}}
```

- Add and install all INF files in a folder (including subdirectories):

```
pnputil /add-driver {{path\to\folder\*.inf}} /subdirs
```

- Delete a driver package using its published name:

```
pnputil /delete-driver {{oemXX.inf}}
```

- Install a driver package using its published name:

```
pnputil /add-driver {{oemXX.inf}}
```

- Export all driver packages to a folder:

```
pnputil /export-driver * {{path\to\folder}}
```

- List all devices:

```
pnputil /enum-devices
```

popd

Changes the current directory to the directory stored by the **pushd** command.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/popd.

- Switch to directory at the top of the stack:

```
popd
```

powercfg

Configure power settings and manage power schemes.

More information: https://learn.microsoft.com/windows-hardware/design/device-experiences/powercfg-command-line-options.

- Display the current power scheme:

```
powercfg /getactivescheme
```

- List all available power schemes:

```
powercfg {{[/L|/list]}}
```

- Set the active power scheme by its GUID:

```
powercfg /setactive {{GUID}}
```

- Turn off the display after a specific number of minutes on AC power:

```
powercfg {{[/X|/change]}} monitor-timeout-ac {{minutes}}
```

- Turn off the display after a specific number of minutes on battery power:

```
powercfg {{[/X|/change]}} monitor-timeout-dc {{minutes}}
```

- Put the system to sleep after a specific number of minutes on AC power:

```
powercfg {{[/X|/change]}} standby-timeout-ac {{minutes}}
```

- Generate a system sleep diagnostics report and save it to a file:

```
powercfg /sleepstudy /output {{path\to\report.html}}
```

- Generate a full power efficiency report and save it to a file:

```
powercfg /energy /output {{path\to\report.html}}
```

powershell

Command-line shell and scripting language designed especially for system administration.

This command refers to PowerShell version 5.1 and below (also known as the legacy Windows PowerShell). To use the newer, cross-platform version of PowerShell (also known as PowerShell Core), use **`pwsh`** instead of **`powershell`**.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/powershell.

- Start an interactive shell session:

```
powershell
```

- Start an interactive shell session without loading startup configs:

```
powershell -NoProfile
```

- Execute specific commands:

```
powershell -Command "{{echo 'powershell is executed'}}"
```

- Execute a specific script:

```
powershell -File {{path/to/script.ps1}}
```

- Start a session with a specific version of PowerShell:

```
powershell -Version {{version}}
```

- Prevent a shell from exit after running startup commands:

```
powershell -NoExit
```

- Describe the format of data sent to PowerShell:

```
powershell -InputFormat {{Text|XML}}
```

- Determine how an output from PowerShell is formatted:

```
powershell -OutputFormat {{Text|XML}}
```

print

Print a text file to a printer.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/print.

- Print a text file to the default printer:

```
print {{path\to\file}}
```

- Print a text file to a specific printer:

```
print /d:{{printer}} {{path\to\file}}
```

prompt

Change the default DOS style prompt in a command window.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/prompt.

- Reset the prompt to the default setting:

```
prompt
```

- Set a specific prompt:

```
prompt {{prompt}}
```

- Change the prompt to show the current date first:

```
prompt $D $P$G
```

- Change the prompt to show the current time first:

```
prompt $T $P$G
```

- Change the prompt by adding a specific text first:

```
prompt {{text}} $P$G
```

psexec

Execute a command-line process on a remote machine.

This is an advanced command and it might potentially be dangerous.

More information: https://learn.microsoft.com/sysinternals/downloads/psexec.

- Execute a command using `cmd` in a remote shell:

```
psexec \\{{remote_host}} cmd
```

- Execute a command on a remote host (pre-authenticated):

```
psexec \\{{remote_host}} -u {{user_name}} -p {{password}}
```

- Execute a command remotely and output the result to a file:

```
psexec \\{{remote_host}} cmd /c {{command}} -an
^>{{path\to\file.txt}}
```

- Execute a program to interact with users:

```
psexec \\{{remote_host}} -d -i {{program_name}}
```

- Display the IP configuration of the remote host:

```
psexec \\{{remote_host}} ipconfig /all
```

psping

A ping tool that includes TCP ping, latency and bandwidth measurement.

More information: https://learn.microsoft.com/sysinternals/downloads/psping.

- Ping a host using ICMP:

```
psping {{hostname}}
```

- Ping a host over a TCP port:

```
psping {{hostname}}:{{port}}
```

- Specify the number of pings and perform it quietly:

```
psping {{hostname}} -n {{pings}} -q
```

- Ping the target over TCP 50 times and produce a histogram of the results:

```
psping {{hostname}}:{{port}} -q -n {{50}} -h
```

- Display help:

```
psping /?
```

PSVersionTable

A read-only variable (as **`$PSVersionTable`**) to get the current PowerShell version.

This command can only be run under PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/about/about_automatic_variables#psversiontable.

- Print a summary of the currently installed PowerShell version and edition:

  ```
  $PSVersionTable
  ```

- Get the detailed (major, minor, build, and revision) version number of PowerShell:

  ```
  $PSVersionTable.PSVersion
  ```

- List all supported PowerShell script versions that this PowerShell version supports:

  ```
  $PSVersionTable.PSCompatibleVersions
  ```

- Get the latest Git commit ID where the currently-installed PowerShell version is based on (works on PowerShell 6.0 and later):

  ```
  $PSVersionTable.GitCommitId
  ```

- Check whether the user is running PowerShell Core (6.0 or later) or the original "Windows PowerShell" (version 5.1 or below):

  ```
  $PSVersionTable.PSEdition
  ```

PSWindowsUpdate

A PowerShell external module to manage Windows Update.

This tool provides multiple commands that all can only be run through PowerShell.

More information: https://github.com/mgajda83/PSWindowsUpdate.

- Install the module using `Install-Module`:

  ```
  Install-Module PSWindowsUpdate
  ```

- List all commands available under the module:

  ```
  Get-Command -Module PSWindowsUpdate
  ```

pushd

Place a directory on a stack so it can be accessed later.

See also: **popd** to switch back to original directory.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/pushd.

- Switch to directory and push it on the stack:

```
pushd {{path\to\directory}}
```

pwd

In PowerShell, this command is an alias of **`Get-Location`**.

However, this command is not available on the Command Prompt (**`cmd`**). Use **`cd`** instead for similar functionality.

- View documentation for the equivalent Command Prompt command:

  ```
  tldr cd
  ```

- View documentation for the original PowerShell command:

  ```
  tldr get-location
  ```

pwlauncher

Manage the Windows To Go startup options.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/pwlauncher.

- Display the current Windows To Go status:

  ```
  pwlauncher
  ```

- Enable or disable the Windows To Go startup options:

  ```
  pwlauncher /{{enable|disable}}
  ```

pwsh where

This command is an alias of **`Where-Object`**.

- View documentation for the original command:

```
tldr Where-Object
```

py

Python Launcher for Windows that runs with specified Python version.

See also: **`python`**.

More information: https://docs.python.org/using/windows.html#python-launcher-for-windows.

- Start a REPL (interactive shell), optionally with arguments supported by `python` (like `-c`, `-m`, etc.):

```
py {{python_arguments}}
```

- Execute a specific Python file:

```
py {{path/to/file.py}}
```

- Run specific Python version. If the version is missing, and `PYLAUNCHER_ALLOW_INSTALL` environment variable is set, auto-install via Microsoft Store or Winget:

```
py {{-2|-3.7|...}}
```

- List installed Python versions:

```
py --list
```

query

Display information about user sessions and process.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/query.

- Display all user sessions:

```
query session
```

- Display the current user sessions on a remote computer:

```
query session /server:{{hostname}}
```

- Display logged in users:

```
query user
```

- Display all user sessions on a remote computer:

```
query session /server:{{hostname}}
```

- Display all running processes:

```
query process
```

- Display running processes by session or user name:

```
query process {{session_name|user_name}}
```

rd

This command is an alias of **`rmdir`** on the Command Prompt and **`Remove-Item`** in PowerShell.

- View documentation for the original Command Prompt command:

  ```
  tldr rmdir
  ```

- View documentation for the original PowerShell command:

  ```
  tldr remove-item
  ```

rdpsign

A tool for signing Remote Desktop Protocol (RDP) files.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/rdpsign.

- Sign an RDP file:

```
rdpsign {{path\to\file.rdp}}
```

- Sign an RDP file using a specific sha256 hash:

```
rdpsign {{path\to\file.rdp}} /sha265 {{hash}}
```

- Enable quiet output:

```
rdpsign {{path\to\file.rdp}} /q
```

- Display verbose warnings, messages and statuses:

```
rdpsign {{path\to\file.rdp}} /v
```

- Test the signing by displaying the output to `stdout` without updating the file:

```
rdpsign {{path\to\file.rdp}} /l
```

reg add

Add new keys and their values to the registry.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/reg-add.

- Add a new registry key:

```
reg add {{key_name}}
```

- Add a new [v]alue under a specific key:

```
reg add {{key_name}} /v {{value}}
```

- Add a new value with specific [d]ata:

```
reg add {{key_name}} /d {{data}}
```

- Add a new value to a key with a specific data [t]ype:

```
reg add {{key_name}} /t REG_{{SZ|MULTI_SZ|DWORD_BIG_ENDIAN|
DWORD|BINARY|DWORD_LITTLE_ENDIAN|LINK|
FULL_RESOURCE_DESCRIPTOR|EXPAND_SZ}}
```

- [f]orcefully (without a prompt) overwrite the existing registry value:

```
reg add {{key_name}} /f
```

reg compare

Compare keys and their values in the registry.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/reg-compare.

- Compare all values under a specific key with another key:

```
reg compare {{key_name1}} {{key_name2}}
```

- Compare a specific [v]alue under two keys:

```
reg compare {{key_name1}} {{key_name2}} /v {{value}}
```

- Compare all [s]ubkeys and values for two keys:

```
reg compare {{key_name1}} {{key_name2}} /s
```

- Only [o]utput the matches ([s]ame) between the specified keys:

```
reg compare {{key_name1}} {{key_name2}} /os
```

- [o]utput the differences and matches ([a]ll) between the specified keys:

```
reg compare {{key_name1}} {{key_name2}} /oa
```

- Compare two keys, [o]utputting [n]othing:

```
reg compare {{key_name1}} {{key_name2}} /on
```

reg copy

Copy keys and their values in the registry.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/reg-copy.

- Copy a registry key to a new registry location:

```
reg copy {{old_key_name}} {{new_key_name}}
```

- Copy a registry key recursively (with all [s]ubkeys) to a new registry location:

```
reg copy {{old_key_name}} {{new_key_name}} /s
```

- [f]orcefully (without a prompt) copy a registry key:

```
reg copy {{old_key_name}} {{new_key_name}} /f
```

reg delete

Delete keys or their values from the registry.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/reg-delete.

- Delete a specific registry key:

```
reg delete {{key_name}}
```

- Delete a [v]alue under a specific key:

```
reg delete {{key_name}} /v {{value}}
```

- Delete [a]ll [v]alues recursively under the specified key:

```
reg delete {{key_name}} /va
```

- [f]orcefully (without a prompt) delete [a]ll [v]alues recursively under a key:

```
reg delete {{key_name}} /f /va
```

reg export

Export the specified subkeys and values to a `.reg` file.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/reg-export.

- Export all subkeys and values of a specific key:

```
reg export {{key_name}} {{path\to\file.reg}}
```

- Forcefully (assuming [y]es) overwrite of an existing file:

```
reg export {{key_name}} {{path\to\file.reg}} /y
```

reg import

Import all available keys, subkeys, and values from a `.reg` file.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/reg-import.

- Import all keys, subkeys and values from a file:

```
reg import {{path\to\file.reg}}
```

reg load

Load saved subkeys into a different subkey in the registry.

Note: This is intended for troubleshooting and temporary keys.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/reg-load.

- Load a backup file into the specified key:

```
reg load {{key_name}} {{path\to\file.hiv}}
```

reg query

Display the values of keys and subkeys in the registry.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/reg-query.

- Display all values of a key:

```
reg query {{key_name}}
```

- Display a specific [v]alue of a key:

```
reg query {{key_name}} /v {{value}}
```

- Display all values of a key and its [s]ubkeys:

```
reg query {{key_name}} /s
```

- Search [f]or keys and values matching a specific pattern:

```
reg query {{key_name}} /f "{{query_pattern}}"
```

- Display a value of a key matching a specified data [t]ype:

```
reg query {{key_name}} /t REG_{{SZ|MULTI_SZ|EXPAND_SZ|DWORD|BINARY|NONE}}
```

- Only search in [d]ata:

```
reg query {{key_name}} /d
```

- Only search in [k]ey names:

```
reg query {{key_name}} /f "{{query_pattern}}" /k
```

- [c]ase-sensitively search for an [e]xact match:

```
reg query {{key_name}} /c /e
```

reg restore

Restore a key and its values from a native **`.hiv`** file.

See **`reg-save`** for more information.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/reg-restore.

- Overwrite a specified key with data from a backup file:

```
reg restore {{key_name}} {{path\to\file.hiv}}
```

reg save

Save a registry key, its subkeys and values to a native **`.hiv`** file.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/reg-save.

- Save a registry key, its subkeys and values to a specific file:

```
reg save {{key_name}} {{path\to\file.hiv}}
```

- Forcefully (assuming [y]es) overwrite an existing file:

```
reg save {{key_name}} {{path\to\file.hiv}} /y
```

reg unload

Remove data from the registry that was loaded using the **`reg load`** command.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/reg-unload.

- Remove data from the registry for a specified key:

```
reg unload {{key_name}}
```

reg

Manage keys and their values in the Windows registry.

Some subcommands such as **add** have their own usage documentation.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/reg.

- Execute a registry command:

```
reg {{command}}
```

- View documentation for adding and copying subkeys:

```
tldr reg {{add|copy}}
```

- View documentation for deleting keys and subkeys:

```
tldr reg {{delete|unload}}
```

- View documentation for searching, viewing, and comparing keys:

```
tldr reg {{compare|query}}
```

- View documentation for exporting and importing registry keys not preserving the key ownerships and ACLs:

```
tldr reg {{export|import}}
```

- View documentation for saving, restoring registry and unloading keys preserving the key ownerships and ACLs:

```
tldr reg {{save|restore|load|unload}}
```

- Display help:

```
reg /?
```

- Display help for a specific command:

```
reg {{command}} /?
```

Remove-AppxPackage

A PowerShell utility to remove an app package from user accounts.

More information: https://learn.microsoft.com/powershell/module/appx/Remove-AppxPackage.

- Remove an app package:

```
Remove-AppxPackage {{package}}
```

- Remove an app package for a specific user:

```
Remove-AppxPackage {{package}} -User {{username}}
```

- Remove an app package for all users:

```
Remove-AppxPackage {{package}} -AllUsers
```

- Remove an app package but preserve it's app data:

```
Remove-AppxPackage {{package}} -PreserveApplicationData
```

Remove-Item

Delete files, folders, as well as registry keys and subkeys.

This command can only be run through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/remove-item.

- Remove specific files or registry keys (without subkeys):

```
Remove-Item {{path\to\file_or_key1 ,
path\to\file_or_key2 ...}}
```

- Remove hidden or read-only files:

```
Remove-Item -Force {{path\to\file1 , path\to\file2 ...}}
```

- Remove specific files or registry keys interactively prompting before each removal:

```
Remove-Item -Confirm {{path\to\file_or_key1 ,
path\to\file_or_key2 ...}}
```

- Remove specific files and directories recursively (Windows 10 version 1909 or later):

```
Remove-Item -Recurse {{path\to\file_or_directory1 ,
path\to\file_or_directory2 ...}}
```

- Remove specific Windows registry keys and all its subkeys:

```
Remove-Item -Recurse {{path\to\key1 , path\to\key2 ...}}
```

- Perform a dry run of the deletion process:

```
Remove-Item -WhatIf {{path\to\file1 , path\to\file2 ...}}
```

ren

This command is an alias of **`Rename-Item`**.

- View documentation for original command:

  ```
  tldr Rename-Item
  ```

Rename-Item

Powershell command to rename an item.

Note: **`ren`** and **`rni`** can both be used as an alias for **`Rename-Item`**.

More information: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/rename-item.

- Rename a file:

```
Rename-Item -Path "{{path/to/file}}" -NewName
"{{new_file_name}}"
```

- Rename a directory:

```
Rename-Item -Path "{{path/to/directory}}" -NewName
"{{new_directory_name}}"
```

- Rename and move a file:

```
Rename-Item -Path "{{path/to/file}}" -NewName "{{path/to/
new_file_name}}"
```

- Rename a file by force:

```
Rename-Item -Path "{{path/to/file}}" -NewName
"{{new_file_name}}" -Force
```

- Prompt confirmation before renaming a file:

```
Rename-Item -Path "{{path/to/file}}" -NewName
"{{new_file_name}}" {{[-Confirm|-cf]}}
```

repair-bde

Attempt to repair or decrypt a damaged BitLocker-encrypted volume.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/repair-bde.

- Attempt to repair a specified volume:

```
repair-bde {{C:}}
```

- Attempt to repair a specified volume and output to another volume:

```
repair-bde {{C:}} {{D:}}
```

- Attempt to repair a specified volume using the provided recovery key file:

```
repair-bde {{C:}} -RecoveryKey {{path\to\file.bek}}
```

- Attempt to repair a specified volume using the provided numerical recovery password:

```
repair-bde {{C:}} -RecoveryPassword {{password}}
```

- Attempt to repair a specified volume using the provided password:

```
repair-bde {{C:}} -Password {{password}}
```

- Attempt to repair a specified volume using the provided key package:

```
repair-bde {{C:}} -KeyPackage {{path\to\directory}}
```

- Log all output to a specific file:

```
repair-bde {{C:}} -LogFile {{path\to\file}}
```

- Display help:

```
repair-bde /?
```

replace

Replace files.

See also: **robocopy**, **move**, **copy**, **del**.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/replace.

- Replace the destination file with the one from the source directory:

```
replace {{path\to\file_or_directory}}
{{path\to\destination_directory}}
```

- Add files to the destination directory instead of replacing existing files:

```
replace {{path\to\file_or_directory}}
{{path\to\destination_directory}} /a
```

- Interactively copy multiple files, with a prompt before replacing or adding a destination file:

```
replace {{path\to\file_or_directory}}
{{path\to\destination_directory}} /p
```

- Replace even read only files:

```
replace {{path\to\file_or_directory}}
{{path\to\destination_directory}} /r
```

- Wait for you to insert a disk before it replaces files (originally to allow inserting a floppy disk):

```
replace {{path\to\file_or_directory}}
{{path\to\destination_directory}} /w
```

- Replace all files in subdirectories of the destination:

```
replace {{path\to\file_or_directory}}
{{path\to\destination_directory}} /s
```

- Replace only files in the destination directory which are older than the files in the source directory:

```
replace {{path\to\file_or_directory}}
{{path\to\destination_directory}} /u
```

- Display help:

```
    replace /?
```

Resolve-Path

Resolve the wildcard characters in a path, and display the path contents.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/resolve-path.

- Resolve the home folder path:

```
Resolve-Path {{~}}
```

- Resolve a UNC path:

```
Resolve-Path -Path "\\{{hostname}}\{{path\to\file}}"
```

- Get relative paths:

```
Resolve-Path -Path {{path\to\file_or_directory}} -Relative
```

ri

In PowerShell, this command is an alias of **`Remove-Item`**.

- View documentation for the original command:

```
tldr remove-item
```

rm

In PowerShell, this command is an alias of **`Remove-Item`**.

- View documentation for the original command:

```
tldr remove-item
```

rmdir

Remove a directory and its contents.

In PowerShell, this command is an alias of **`Remove-Item`**. This documentation is based on the Command Prompt (**`cmd`**) version of **`rmdir`**.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/rmdir.

- View the documentation of the equivalent PowerShell command:

```
tldr remove-item
```

- Remove an empty directory:

```
rmdir {{path\to\directory}}
```

- Remove a directory and its contents recursively:

```
rmdir {{path\to\directory}} /s
```

- Remove a directory and its contents recursively without prompting:

```
rmdir {{path\to\directory}} /s /q
```

rni

This command is an alias of **`Rename-Item`**.

- View documentation for original command:

```
tldr Rename-Item
```

robocopy

Robust File and Folder Copy.

By default files will only be copied if the source and destination have different time stamps or different file sizes.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/robocopy.

- Copy all `.jpg` and `.bmp` files from one directory to another:

```
robocopy {{path\to\source_directory}}
{{path\to\destination_directory}} {{*.jpg}} {{*.bmp}}
```

- Copy all files and subdirectories, including empty ones:

```
robocopy {{path\to\source_directory}}
{{path\to\destination_directory}} /E
```

- Mirror/Sync a directory, deleting anything not in source and include all attributes and permissions:

```
robocopy {{path\to\source_directory}}
{{path\to\destination_directory}} /MIR /COPYALL
```

- Copy all files and subdirectories, excluding source files that are older than destination files:

```
robocopy {{path\to\source_directory}}
{{path\to\destination_directory}} /E /XO
```

- List all files 50 MB or larger instead of copying them:

```
robocopy {{path\to\source_directory}}
{{path\to\destination_directory}} /MIN:{{52428800}} /L
```

- Allow resuming if network connection is lost and limit retries to 5 and wait time to 15 sec:

```
robocopy {{path\to\source_directory}}
{{path\to\destination_directory}} /Z /R:5 /W:15
```

- Display help:

```
robocopy /?
```

rpcinfo

List programs via RPC on remote computers.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/rpcinfo.

- List all programs registered on the local computer:

```
rpcinfo
```

- List all programs registered on a remote computer:

```
rpcinfo /p {{computer_name}}
```

- Call a specific program on a remote computer using TCP:

```
rpcinfo /t {{computer_name}} {{program_name}}
```

- Call a specific program on a remote computer using UDP:

```
rpcinfo /u {{computer_name}} {{program_name}}
```

runas

Run a program as another user.

More information: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc771525(v=ws.11).

- Run a program as the Administrator local user:

```
runas /user:.\Administrator "{{command}}"
```

- Run a program as a specific user:

```
runas /user:{{domain\username}} "{{command}}"
```

- Run a program without loading the user's profile:

```
runas /noprofile /user:{{domain\username}} "{{command}}"
```

- Open Command Prompt as another user:

```
runas /user:{{domain\username}} cmd
```

- Run Notepad as a specific user with the current user's environment variables, opening a file with escaped quotes:

```
runas /env /user:{{domain\username}} "notepad \"{{C:\path\to\file.txt}}\""
```

- Run Active Directory Users and Computers as a specific user:

```
runas /env /user:{{domain\username}} "mmc %windir%\system32\dsa.msc"
```

sal

This command is an alias of **`Set-Alias`**.

- View documentation for the original command:

```
tldr set-alias
```

SC

Communicate with the Service Control Manager and services.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/sc-query.

- Show the status of a service (no service name will list all services):

```
sc query {{service_name}}
```

- Start a service asynchronously:

```
sc start {{service_name}}
```

- Stop a service asynchronously:

```
sc stop {{service_name}}
```

- Create a service:

```
sc create {{service_name}} binpath=
{{path\to\service_binary_file}}
```

- Delete a service:

```
sc delete {{service_name}}
```

- Set the type of a service:

```
sc config {{service_name}} type= {{service_type}}
```

scb

This command is an alias of **`Set-Clipboard`**.

- View documentation:

```
tldr Set-Clipboard
```

scoop bucket

Manage buckets: Git repositories containing files which describe how scoop installs applications.

If Scoop doesn't know where the bucket is located its repository location must be specified.

More information: https://github.com/lukesampson/scoop/wiki/Buckets.

- List all buckets currently in use:

```
scoop bucket list
```

- List all known buckets:

```
scoop bucket known
```

- Add a known bucket by its name:

```
scoop bucket add {{name}}
```

- Add an unknown bucket by its name and Git repository URL:

```
scoop bucket add {{name}} {{https://example.com/repository.git}}
```

- Remove a bucket by its name:

```
scoop bucket rm {{name}}
```

scoop

The Scoop package manager.

Some subcommands such as **`bucket`** have their own usage documentation.

More information: https://scoop.sh.

- Install a package:

```
scoop install {{package}}
```

- Remove a package:

```
scoop uninstall {{package}}
```

- Update all installed packages:

```
scoop update --all
```

- List installed packages:

```
scoop list
```

- Display information about a package:

```
scoop info {{package}}
```

- Search for a package:

```
scoop search {{package}}
```

- Remove old versions of all packages and clear the download cache:

```
scoop cleanup --cache --all
```

sdelete

Securely delete file/directory from disk, or clean the free space on a volume/physical disk.

More information: https://learn.microsoft.com/en-us/sysinternals/downloads/sdelete.

- Delete files with 3 [p]asses:

```
sdelete -p 3 {{path\to\file1 path\to\file2 ...}}
```

- Delete folders and its [s]ubdirectories with 1 pass (default):

```
sdelete -s {{path\to\directory1 path\to\directory2 ...}}
```

- Clean the free space of volume D: with 3 [p]asses:

```
sdelete -p 3 D:
```

- Clean the free space with [z]eros of physical disk 2, which should not contain any volumes to be cleaned:

```
sdelete -z 2
```

sdelete64

This command is the 64 bit version of **`sdelete`**.

- View documentation for the original command:

  ```
  tldr sdelete
  ```

Select-String

Finds text in strings and files in PowerShell.

Note: This command can only be used through PowerShell.

You can use **`Select-String`** similar to **`grep`** in UNIX or **`findstr.exe`** in Windows.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/select-string.

- Search for a pattern within a file:

```
Select-String -Path "{{path\to\file}}" -Pattern
'{{search_pattern}}'
```

- Search for an exact string (disables `regex`):

```
Select-String -SimpleMatch
"{{exact_string}}" {{path\to\file}}
```

- Search for pattern in all `.ext` files in current dir:

```
Select-String -Path "{{*.ext}}" -Pattern '{{search_pattern}}'
```

- Capture the specified number of lines before and after the line that matches the pattern:

```
Select-String --Context {{2,3}}
"{{search_pattern}}" {{path\to\file}}
```

- Search `stdin` for lines that do not match a pattern:

```
Get-Content {{path\to\file}} | Select-String --NotMatch
"{{search_pattern}}"
```

Set-Acl

Changes the security descriptor of a specified item, such as a file or a registry key.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.security/set-acl.

- Copy a security descriptor from one file to another:

```
$OriginAcl = Get-Acl -Path {{path\to\file}}; Set-Acl -Path {{path\to\file}} -AclObject $OriginAcl
```

- Use the pipeline operator to pass a descriptor:

```
Get-Acl -Path {{path\to\file}} | Set-Acl -Path {{path\to\file}}
```

Set-Alias

A PowerShell command to set or modify alias.

Note: **`sal`** can be used as an alias for **`Set-Alias`**.

More information: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/set-alias.

- Create or reassign new alias:

```
Set-Alias -Name {{text}} -Value {{command}}
```

- Add a description to alias:

```
Set-Alias -Name {{text}} -Value {{command}} -Description "{{description}}"
```

Set-Clipboard

PowerShell command to set content to clipboard.

Note: **`scb`** can be used as an alias for **`Set-Clipboard`**.

More information: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/set-clipboard.

- Copy text to the clipboard:

```
Set-Clipboard -Value "{{text}}"
```

- Copy multiple texts to clipboard separated by new line:

```
Set-Clipboard -Value @("{{text 1}}", "{{text 2}}", "{{text 3}}")
```

- Copy files or directories to clipboard:

```
Set-Clipboard -Path "{{path/to/files_or_directories}}"
```

- Copy multiple files:

```
Set-Clipboard -Path "{{path/to/file1}}","{{path/to/file2}}","{{path/to/file3}}"
```

- To clear the clipboard:

```
Set-Clipboard ""
```

Set-Date

Changes the system time on the computer to a time that you specify.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/set-date.

- Add three days to the system date:

```
Set-Date -Date (Get-Date).AddDays({{3}})
```

- Set the system clock back 10 minutes:

```
Set-Date -Adjust -0:10:0 -DisplayHint Time
```

- Add 90 minutes to the system clock:

```
$90mins = New-TimeSpan -Minutes {{90}}; Set-Date -Adjust
$90mins
```

Set-Location

Display the current working directory or move to a different directory.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/set-location.

- Go to the specified directory:

```
Set-Location {{path\to\directory}}
```

- Go to a specific directory in a different drive:

```
Set-Location {{C}}:{{path\to\directory}}
```

- Go and display the location of specified directory:

```
Set-Location {{path\to\directory}} -PassThru
```

- Go up to the parent of the current directory:

```
Set-Location ..
```

- Go to the home directory of the current user:

```
Set-Location ~
```

- Go back/forward to the previously chosen directory:

```
Set-Location {{-|+}}
```

- Go to root of current drive:

```
Set-Location \
```

Set-Service

Starts, stops, and suspends a service, and changes its properties.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/set-service.

- Change a display name:

```
Set-Service -Name {{hostname}} -DisplayName "{{name}}"
```

- Change the startup type of services:

```
Set-Service -Name {{service_name}} -StartupType {{Automatic}}
```

- Change the description of a service:

```
Set-Service -Name {{service_name}} -Description
"{{description}}"
```

Set-Volume

Sets or changes the file system label of an existing volume.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/storage/set-volume.

- Change the file system label of a volume identified by drive letter:

```
Set-Volume -DriveLetter "D" -NewFileSystemLabel "DataVolume"
```

- Change the file system label of a volume identified by the system label:

```
Set-Volume -FileSystemLabel "OldLabel" -NewFileSystemLabel
"NewLabel"
```

- Modify the properties of a volume using a volume object:

```
Set-Volume -InputObject $(Get-Volume -DriveLetter "E") -
NewFileSystemLabel "Backup"
```

- Specify the Data Deduplication mode for the volume:

```
Set-Volume -DriveLetter "D" -DedupMode Backup
```

set

Display or set environment variables for the current instance of CMD.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/set.

- List all current environment variables:

```
set
```

- Set an environment variable to a specific value:

```
set {{name}}={{value}}
```

- List environment variables starting with the specified string:

```
set {{name}}
```

- Prompt the user for a value for the specified variable:

```
set /p {{name}}={{prompt_string}}
```

setx

Set persistent environment variables.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/setx.

- Set an environment variable for the current user:

```
setx {{variable}} {{value}}
```

- Set an environment variable for the current machine:

```
setx {{variable}} {{value}} /M
```

- Set an environment variable for a user on a remote machine:

```
setx /s {{hostname}} /u {{username}} /p {{password}} {{variable}} {{value}}
```

- Set an environment variable from a registry key value:

```
setx {{variable}} /k {{registry\key\path}}
```

sfc

Scans the integrity of Windows system files.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/sfc.

- Display information about the usage of the command:

```
sfc
```

- Scan all system files and, if possible, repair any problems:

```
sfc /scannow
```

- Scan all system files without attempting to repair any:

```
sfc /verifyonly
```

- Scan a specific file and, if possible, repair any problems:

```
sfc /scanfile={{path\to\file}}
```

- Scan a specific file without attempting to repair it:

```
sfc /verifyfile={{path\to\file}}
```

- When repairing offline, specify the boot directory:

```
sfc /offbootdir={{path\to\directory}}
```

- When repairing offline, specify the Windows directory:

```
sfc /offwindir={{path\to\directory}}
```

Show-Markdown

Shows a Markdown file or string in the console in a friendly way using VT100 escape sequences or in a browser using HTML.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/show-markdown.

- Render markdown to console from a file:

```
Show-Markdown -Path {{path\to\file}}
```

- Render markdown to console from string:

```
"{{# Markdown content}}" | Show-Markdown
```

- Open Markdown file in a browser:

```
Show-Markdown -Path {{path\to\file}} -UseBrowser
```

showmount

Display information about NFS filesystems on Windows Server.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/showmount.

- Display all exported filesystems:

```
showmount -e
```

- Display all NFS clients and their mounted directories:

```
showmount -a
```

- Display all NFS mounted directories:

```
showmount -d
```

- Display all exported filesystems for a remote server:

```
showmount -e {{server_address}}
```

shutdown

A tool for shutting down, restarting or logging off a machine.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/shutdown.

- Shutdown the current machine:

```
shutdown /s
```

- Shutdown the current machine force-closing all apps:

```
shutdown /s /f
```

- Restart the current machine immediately:

```
shutdown /r /t 0
```

- Hibernate the current machine:

```
shutdown /h
```

- Log off the current machine:

```
shutdown /l
```

- Specify a timeout in seconds to wait before shutting down:

```
shutdown /s /t {{8}}
```

- Abort a shutdown sequence whose timeout is yet to expire:

```
shutdown /a
```

- Shutdown a remote machine:

```
shutdown /m {{\\hostname}}
```

sl

In PowerShell, this command is an alias of **`Set-Location`**.

- View documentation for the original command:

```
tldr set-location
```

slmgr

This command is an alias of **`slmgr.vbs`**.

- View documentation for the original command:

```
tldr slmgr.vbs
```

slmgr.vbs

Install, activate, and manage Windows licenses.

This command may override, deactivate, and/or remove your current Windows license. Please proceed with caution.

More information: https://learn.microsoft.com/windows-server/get-started/activation-slmgr-vbs-options.

- [d]isplay the current Windows [l]icense [i]nformation:

```
slmgr.vbs /dli
```

- [d]isplay the ins[t]allation [i]D for the current device. Useful for offline license activation:

```
slmgr.vbs /dti
```

- Display the current license's e[xp]i[r]ation date and time:

```
slmgr.vbs /xpr
```

- [i]nstall a new Windows license [p]roduct [k]ey. Requires Administrator privileges and will override the existing license:

```
slmgr.vbs /ipk {{product_key}}
```

- [a]c[t]ivate the Windows product license [o]nline. Requires Administrator privileges to do so:

```
slmgr.vbs /ato
```

- [a]c[t]ivate the Windows [p]roduct license offline. Requires Administrator privileges and an Confirmation ID provided by Microsoft Product Activation Center:

```
slmgr.vbs /atp {{confirmation_id}}
```

- [c]lear the current license's [p]roduct [k]e[y] from the Windows Registry. This will not deactivate or uninstall the current license, but prevents the key from being stolen by malicious programs in the future:

```
slmgr.vbs /cpky
```

- [u]ninstall the current license (by its [p]roduct [k]ey):

```
slmgr.vbs /upk
```

sls

This command is an alias of **`Select-String`**.

- View documentation for the original command:

  ```
  tldr select-string
  ```

Sort-Object

Sorts objects by property values.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/sort-object.

- Sort the current directory by name:

```
Get-ChildItem | Sort-Object
```

- Sort the current directory by name descending:

```
Get-ChildItem | Sort-Object -Descending
```

- Sort items removing duplicates:

```
"a", "b", "a" | Sort-Object -Unique
```

- Sort the current directory by file length:

```
Get-ChildItem | Sort-Object -Property Length
```

- Sort processes with the highest memory usage based on their working set (WS) size:

```
Get-Process | Sort-Object -Property WS
```

Start-Service

Starts stopped services.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/start-service.

- Start a service by using its name:

```
Start-Service -Name {{service_name}}
```

- Display information without starting a service:

```
Start-Service -DisplayName *{{name}}* -WhatIf
```

- Start a disabled service:

```
Set-Service {{service_name}} -StartupType {{manual}}; Start-Service {{service_name}}
```

Stop-Service

Stops running services.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/stop-service.

- Stop a service on the local computer:

  ```
  Stop-Service -Name {{service_name}}
  ```

- Stop a service by using the display name:

  ```
  Stop-Service -DisplayName "{{name}}"
  ```

- Stop a service that has dependent services:

  ```
  Stop-Service -Name {{service_name}} -Force -Confirm
  ```

subst

Associates a path with a virtual drive letter.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/subst.

- List active associations:

```
subst
```

- Add an association:

```
subst {{Z:}} {{C:\Python2.7}}
```

- Remove an association:

```
subst {{Z:}} /d
```

systeminfo

Display operating system configuration for a local or remote machine.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/systeminfo.

- Display system configuration for the local machine:

```
systeminfo
```

- Display system configuration in a specified output format:

```
systeminfo /fo {{table|list|csv}}
```

- Display system configuration for a remote machine:

```
systeminfo /s {{remote_name}} /u {{username}} /p {{password}}
```

- Display help:

```
systeminfo /?
```

takeown

Take ownership of a file or directory.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/takeown.

- Take ownership of the specified file:

```
takeown /f {{path\to\file}}
```

- Take ownership of the specified directory:

```
takeown /d {{path\to\directory}}
```

- Take ownership of the specified directory and all subdirectories:

```
takeown /r /d {{path\to\directory}}
```

- Change ownership to the Administrator group instead of the current user:

```
takeown /a /f {{path\to\file}}
```

taskkill

Terminate a process by its process ID or name.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/taskkill.

- Terminate a process by its ID:

```
taskkill /pid {{process_id}}
```

- Terminate a process by its name:

```
taskkill /im {{process_name}}
```

- Forcefully terminate a specified process:

```
taskkill /pid {{process_id}} /f
```

- Terminate a process and its child processes:

```
taskkill /im {{process_name}} /t
```

- Terminate a process on a remote machine:

```
taskkill /pid {{process_id}} /s {{remote_name}}
```

- Display information about the usage of the command:

```
taskkill /?
```

tasklist

Display a list of currently running processes on a local or remote machine.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/tasklist.

- Display currently running processes:

```
tasklist
```

- Display running processes in a specified output format:

```
tasklist /fo {{table|list|csv}}
```

- Display running processes using the specified `.exe` or `.dll` file name:

```
tasklist /m {{module_pattern}}
```

- Display processes running on a remote machine:

```
tasklist /s {{remote_name}} /u {{username}} /p {{password}}
```

- Display services using each process:

```
tasklist /svc
```

Tee-Object

Saves command output in a file or variable and also sends it down the pipeline.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/tee-object.

- Output processes to a file and to the console:

```
Get-Process | Tee-Object -FilePath {{path\to\file}}
```

- Output processes to a variable and `Select-Object`:

```
Get-Process notepad | Tee-Object -Variable {{proc}} | Select-Object processname,handles
```

Test-Json

Test whether a string is a valid JSON document.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/test-json.

- Test if a string from `stdin` is in JSON format:

```
'{{string}}' | Test-Json
```

- Test if a string JSON format:

```
Test-Json -Json '{{json_to_test}}'
```

- Test if a string from `stdin` matches a specific schema file:

```
'{{string}}' | Test-Json -SchemaFile
{{path\to\schema_file.json}}
```

Test-NetConnection

Display diagnostic information for a connection.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/nettcpip/test-netconnection.

- Test a connection and display detailed results:

```
Test-NetConnection -InformationLevel Detailed
```

- Test a connection to a remote host using the specified port number:

```
Test-NetConnection -ComputerName {{ip_or_hostname}} -Port {{port_number}}
```

time

Display or set the system time.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/time.

- Display the current system time and prompt to enter a new time (leave empty to keep unchanged):

```
time
```

- Display the current system time without prompting for a new time:

```
time /t
```

title

Set the title of the command prompt window.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/title.

- Set the title of the current command prompt window:

```
title {{new_title}}
```

tracert

Receive information about each step in the route between your PC and the target.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/tracert.

- Trace a route:

```
tracert {{IP}}
```

- Prevent `tracert` from resolving IP addresses to hostnames:

```
tracert /d {{IP}}
```

- Force `tracert` to use IPv4 only:

```
tracert /4 {{IP}}
```

- Force `tracert` to use IPv6 only:

```
tracert /6 {{IP}}
```

- Specify the maximum number of hops in the search for the target:

```
tracert /h {{max_hops}} {{IP}}
```

- Display help:

```
tracert /?
```

tree

Display a graphical tree of the directory structure for a path.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/tree.

- Display the tree for the current directory:

```
tree
```

- Display the tree for a specific directory:

```
tree {{path\to\directory}}
```

- Display the tree for a directory including [f]iles:

```
tree {{path\to\directory}} /f
```

- Display the tree using [a]SCII characters instead of extended characters:

```
tree {{path\to\directory}} /a
```

tskill

Ends a process running in a session on a Remote Desktop Session Host.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/tskill.

- Terminate a process by its process identifier:

```
tskill {{process_id}}
```

- Terminate a process by its name:

```
tskill {{process_name}}
```

type

Display the contents of a file.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/type.

- Display the contents of a specific file:

```
type {{path\to\file}}
```

tzutil

A tool for displaying or configuring the system time zone.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/tzutil.

- Get the current time zone:

```
tzutil /g
```

- Display a list of available time zones:

```
tzutil /l
```

- Set the system time zone to the specific value:

```
tzutil /s {{timezone_id}}
```

uwfmgr

Unified Write Filter (UWF).

Protect drives by redirecting any writes to the drive to a virtual overlay. Writes are discarded upon reboot unless committed by default.

More information: https://learn.microsoft.com/windows/iot/iot-enterprise/customize/unified-write-filter.

- Get the current status:

```
uwfmgr get-config
```

- Set a drive as protected:

```
uwfmgr volume protect {{drive_letter}}:
```

- Remove a drive from protection list:

```
uwfmgr volume unprotect {{drive_letter}}:
```

- Enable or disable protection (Applies after reboot):

```
uwfmgr filter {{enable|disable}}
```

- Commit changes of a file on protected drive:

```
uwfmgr file commit {{drive_letter:\path\to\file}}
```

- Commit deletion of a file on protected drive:

```
uwfmgr file commit-delete {{drive_letter:\path\to\file}}
```

vcvarsall

Setup the environment variables required for using the Microsoft Visual Studio tools.

The path of **`vcvarsall`** for a certain Visual Studio installation can be found using **`vswhere`**.

More information: https://learn.microsoft.com/cpp/build/building-on-the-command-line.

- Setup the environment for native x64:

  ```
  vcvarsall x64
  ```

- Setup the environment for cross-compiled native x86 from the x64 host:

  ```
  vcvarsall x64_x86
  ```

- Setup the environment for cross-compiled native Arm x64 from the x64 host:

  ```
  vcvarsall x64_arm64
  ```

- Setup the environment for native UWP x64:

  ```
  vcvarsall x64 uwp
  ```

ventoy

This command is an alias of **`Ventoy2Disk`**.

- View documentation for the original command:

  ```
  tldr ventoy2disk
  ```

Ventoy2Disk

A tool to create bootable USB drives using ISO files on Windows systems.

More information: https://www.ventoy.net/en/doc_windows_cli.html.

- Install Ventoy to drive D: with default settings (MBR, Secure Boot enabled):

```
Ventoy2Disk VTOYCLI /I /Drive:D:
```

- Install Ventoy with GPT partition style and disable Secure Boot:

```
Ventoy2Disk VTOYCLI /I /Drive:D: /GPT /NOSB
```

- Install Ventoy and reserve 4GB space at the end of the disk:

```
Ventoy2Disk VTOYCLI /I /Drive:D: /R:4096
```

- Install Ventoy using physical drive number 1 with NTFS filesystem:

```
Ventoy2Disk VTOYCLI /I /PhyDrive:1 /FS:NTFS
```

- Install Ventoy without USB type checking (for internal drives):

```
Ventoy2Disk VTOYCLI /I /Drive:D: /NOUSBCheck
```

- Update Ventoy on drive D: while keeping current settings:

```
Ventoy2Disk VTOYCLI /U /Drive:D:
```

- Perform non-destructive installation (preserve existing data):

```
Ventoy2Disk VTOYCLI /I /Drive:D: /NonDest
```

- Install Ventoy with all options: GPT, no Secure Boot, 2GB reserved, NTFS, no USB check:

```
Ventoy2Disk VTOYCLI /I /Drive:D: /GPT /NOSB /R:2048 /
FS:NTFS /NOUSBCheck
```

ver

Display the current Windows or MS-DOS version number.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/ver.

- Display the current version number:

```
ver
```

virtualboxvm

Manage VirtualBox virtual machines.

More information: https://www.virtualbox.org.

- Start a virtual machine:

```
virtualboxvm --startvm {{name|uuid}}
```

- Start a virtual machine in fullscreen mode:

```
virtualboxvm --startvm {{name|uuid}} --fullscreen
```

- Mount the specified DVD image file:

```
virtualboxvm --startvm {{name|uuid}} --dvd
{{path\to\image_file}}
```

- Display a command-line window with debug information:

```
virtualboxvm --startvm {{name|uuid}} --debug-command-line
```

- Start a virtual machine in a paused state:

```
virtualboxvm --startvm {{name|uuid}} --start-paused
```

vol

Display information about volumes.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/vol.

- Display the label and serial number for the current drive:

```
vol
```

- Display the label and serial number for a specific volume:

```
vol {{D:}}
```

vol.py

Forensics framework used to analyze volatile memory (RAM) dumps.

With volatility3, plugins are now based on operating system. Examples below will use Windows.

More information: https://volatility3.readthedocs.io/en/latest/index.html.

- Get information about a memory dump file:

```
python3 vol.py {{[-f|--filename]}} {{path/to/
memory_dump_file}} windows.info
```

- List active processes:

```
python3 vol.py {{[-f|--filename]}} {{path/to/
memory_dump_file}} windows.pslist
```

- List hashes of users on system:

```
python3 vol.py {{[-f|--filename]}} {{path/to/
memory_dump_file}} windows.hashdump
```

- List active network connections:

```
python3 vol.py {{[-f|--filename]}} {{path/to/
memory_dump_file}} windows.netstat
```

- Display help:

```
python3 vol.py {{[-h|--help]}}
```

vswhere

Locate Visual Studio 2017 and newer installations.

More information: https://github.com/microsoft/vswhere.

- Find the path of vcvarsall.bat to set environment variables:

```
vswhere -products * -latest -prerelease -find
**\VC\Auxiliary\Build\vcvarsall.bat
```

- Find the directory of the x64 MSVC compiler (cl.exe, etc):

```
vswhere -products * -latest -prerelease -find
**\Hostx64\x64\*
```

- Find the directory of Clang bundled with Visual Studio bundled (clang-cl, clang-tidy, etc):

```
vswhere -products * -latest -prerelease -find **\Llvm\bin\*
```

- Find the path of `MSBuild.exe`:

```
vswhere -products * -latest -prerelease -find
MSBuild\**\Bin\MSBuild.exe
```

w32tm

Query and control the w32time time synchronization service.

More information: https://learn.microsoft.com/windows-server/networking/windows-time-service/windows-time-service-tools-and-settings.

- Show the current status of time synchronization:

```
w32tm /query /status /verbose
```

- Show a time offset graph against a time server:

```
w32tm /stripchart /computer:{{time_server}}
```

- Show an NTP reply from a time server:

```
w32tm /stripchart /packetinfo /samples:1 /computer:
{{time_server}}
```

- Show the state of the currently used time servers:

```
w32tm /query /peers
```

- Show configuration of the w32time service (run in elevated console):

```
w32tm /query /configuration
```

- Force time resynchronization immediately (run in elevated console):

```
w32tm /resync /force
```

- Write w32time debug logs into a file (run in elevated console):

```
w32tm /debug /enable /file:{{path\to\debug.log}} /size:
{{10000000}} /entries:{{0-300}}
```

Wait-Process

Waits for the processes to be stopped before accepting more input.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.management/wait-process.

- Stop a process and wait:

```
Stop-Process -Id {{process_id}}; Wait-Process -Id {{process_id}}
```

- Wait for processes for a specified time:

```
Wait-Process -Name {{process_name}} -Timeout {{30}}
```

wget

In PowerShell, this command may be an alias of **`Invoke-WebRequest`** when the original **`wget`** program (https://www.gnu.org/software/wget) is not properly installed.

Note: If version command returns an error, PowerShell may have substituted this command with **`Invoke-WebRequest`**.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/invoke-webrequest.

- View documentation for the original `wget` command:

```
tldr wget {{[-p|--platform]}} common
```

- View documentation for PowerShell's `Invoke-WebRequest` command:

```
tldr invoke-webrequest
```

- Display version:

```
wget --version
```

Where-Object

Selects objects from a collection based on their property values.

Note: This command can only be used through PowerShell.

More information: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/where-object.

- Filter aliases by its name:

```
Get-Alias | Where-Object -{{Property}} {{Name}} -{{eq}} {{name}}
```

- List all services that are currently stopped. The `$_` automatic variable represents each object that is passed to the `Where-Object` cmdlet:

```
Get-Service | Where-Object {$_.Status -eq "Stopped"}
```

- Use multiple conditions:

```
Get-Module -ListAvailable | Where-Object { $_.Name -NotLike "Microsoft*" -And $_.Name -NotLike "PS*" }
```

where

Display the location of files that match the search pattern.

Defaults to current work directory and paths in the PATH environment variable.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/where.

- Display the location of file pattern:

```
where {{file_pattern}}
```

- Display the location of file pattern including file size and date:

```
where /T {{file_pattern}}
```

- Recursively search for file pattern at specified path:

```
where /R {{path\to\directory}} {{file_pattern}}
```

- Silently return the error code for the location of the file pattern:

```
where /Q {{file_pattern}}
```

whoami

Display details about the current user.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/whoami.

- Display the username of the current user:

```
whoami
```

- Display the groups that the current user is a member of:

```
whoami /groups
```

- Display the privileges of the current user:

```
whoami /priv
```

- Display the user principal name (UPN) of the current user:

```
whoami /upn
```

- Display the logon ID of the current user:

```
whoami /logonid
```

- Display all information for the current user:

```
whoami /all
```

winget

Windows Package Manager.

More information: https://learn.microsoft.com/windows/package-manager/winget.

- Install a package (use `--source` when package is available from multiple sources):

```
winget {{[add|install]}} {{package}}
```

- Remove a package (Note: `remove` can also be used instead of `uninstall`):

```
winget {{[rm|uninstall]}} {{package}}
```

- Display information about a package:

```
winget show {{package}}
```

- Search for a package:

```
winget search {{package}}
```

- Upgrade all packages to the latest versions:

```
winget upgrade {{[-r|--all]}}
```

- List all packages installed that can be managed with `winget`:

```
winget {{[ls|list]}} {{[-s|--source]}} winget
```

- Import packages from a file, or export installed packages to a file:

```
winget {{import|export}} {{--import-file|--output}} {{path/to/file}}
```

- Validate manifests before submitting a PR to the winget-pkgs repository:

```
winget validate {{path/to/manifest}}
```

wmic

Interactive shell for detailed information about running processes.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/wmic.

- Fundamental grammar:

```
wmic {{alias}} {{where_clause}} {{verb_clause}}
```

- Show brief details about the currently running processes:

```
wmic process list brief
```

- Show full details about the currently running processes:

```
wmic process list full
```

- Access specific fields such as process name, process ID and parent process ID:

```
wmic process get {{name,processid,parentprocessid}}
```

- Display information about a specific process:

```
wmic process where {{name="example.exe"}} list full
```

- Display specific fields for a specific process:

```
wmic process where processid={{pid}} get {{name,commandline}}
```

- Kill a process:

```
wmic process {{pid}} delete
```

wsl-open

Open a file or URL from within Windows Subsystem for Linux in the user's default Windows GUI application.

More information: https://gitlab.com/4U6U57/wsl-open.

- Open the current directory in Windows Explorer:

```
wsl-open {{.}}
```

- Open a URL in the user's default web browser in Windows:

```
wsl-open {{https://example.com}}
```

- Open a specific file in the user's default application in Windows:

```
wsl-open {{path\to\file}}
```

- Set `wsl-open` as the shell's web browser (open links with `wsl-open`):

```
wsl-open -w
```

- Display help:

```
wsl-open -h
```

wsl

Manage the Windows Subsystem for Linux.

More information: https://learn.microsoft.com/windows/wsl/reference.

- Start a Linux shell (in the default distribution):

```
wsl {{shell_command}}
```

- Run a Linux command without using a shell:

```
wsl {{[-e|--exec]}} {{command}} {{command_arguments}}
```

- Specify a particular distribution:

```
wsl {{[-d|--distribution]}} {{distribution}}
{{shell_command}}
```

- List available distributions:

```
wsl {{[-l|--list]}}
```

- Export a distribution to a `.tar` file:

```
wsl --export {{distribution}} {{path\to\distro_file.tar}}
```

- Import a distribution from a `.tar` file:

```
wsl --import {{distribution}} {{path\to\install_location}}
{{path/to/distro_file.tar}}
```

- Change the version of wsl used for the specified distribution:

```
wsl --set-version {{distribution}} {{version}}
```

- Shut down Windows Subsystem for Linux:

```
wsl --shutdown
```

xcopy

Copy files and directory trees.

More information: https://learn.microsoft.com/windows-server/administration/windows-commands/xcopy.

- Copy the file(s) to the specified destination:

```
xcopy {{path\to\file_or_directory}}
{{path\to\destination_directory}}
```

- List files that will be copied before copying:

```
xcopy {{path\to\file_or_directory}}
{{path\to\destination_directory}} /p
```

- Copy the directory structure only, excluding files:

```
xcopy {{path\to\file_or_directory}}
{{path\to\destination_directory}} /t
```

- Include empty directories when copying:

```
xcopy {{path\to\file_or_directory}}
{{path\to\destination_directory}} /e
```

- Keep the source ACL in the destination:

```
xcopy {{path\to\file_or_directory}}
{{path\to\destination_directory}} /o
```

- Allow resuming when network connection is lost:

```
xcopy {{path\to\file_or_directory}}
{{path\to\destination_directory}} /z
```

- Disable the prompt when the file exists in the destination:

```
xcopy {{path\to\file_or_directory}}
{{path\to\destination_directory}} /y
```

- Display help:

```
xcopy /?
```