

# tldr pages book

Simplified and community-driven man pages

*Generated on Wed Jun 19 04:21:15 2024*

Website: <https://tldr.sh>

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

# Android

# am

Androids Aktivitäten-Manager.

Weitere Informationen: <https://developer.android.com/tools/adb#am>.

- Starte eine bestimmte Aktivität:

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

- Starte eine Aktivität und übergib ihr Daten:

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

- Starte eine Aktivität, auf die eine bestimmte Aktion und Kategorie zutrifft:

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

- Konvertiere ein bestimmtes Ziel in einen URI:

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

# bugreport

Zeige Android-Fehlerberichte.

Dieser Befehl kann nur mit **adb shell** verwendet werden.

Weitere Informationen: <https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreport>.

- Zeige einen vollständigen Fehlerbericht eines Androidgeräts an:

`bugreport`

# bugreportz

Generiere einen gezippten Android-Fehlerbericht.

Dieser Befehl kann nur mit **adb shell** verwendet werden.

Weitere Informationen: <https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreportz>.

- Generiere einen vollständigen Fehlerbericht eines Androidgeräts:

```
bugreportz
```

- Zeige den Fortschritt eines laufenden bugreportz Prozesses an:

```
bugreportz -p
```

- Zeige die Versionsnummer von bugreportz an:

```
bugreportz -v
```

- Zeige Hilfe an:

```
bugreportz -h
```

# cmd

Androids Service-Manager.

Weitere Informationen: <https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/cmd/>.

- Liste alle laufenden Services auf:

```
cmd -l
```

- Rufe einen bestimmten Service auf:

```
cmd {{alarm}}
```

- Rufe einen bestimmten Service mit Parametern auf:

```
cmd {{vibrator}} {{vibrate 300}}
```

# dalvikvm

Androids Java Virtual Machine.

Weitere Informationen: <https://source.android.com/docs/core/runtime>.

- Starte ein Java-Programm:

```
dalvikvm -classpath {{pfad/zu/datei.jar}} {{klassenname}}
```

# dumpsys

Stelle Informationen über Android-Systemservices bereit.

Dieser Befehl kann nur mit **adb shell** verwendet werden.

Weitere Informationen: <https://developer.android.com/tools/dumpsys>.

- Erhalte diagnostische Informationen über alle Systemservices:

```
dumpsys
```

- Erhalte diagnostische Informationen über einen bestimmten Systemservice:

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

- Liste alle Services, über die **dumpsys** Informationen bereitstellen kann auf:

```
dumpsys -l
```

- Liste Service-spezifische Argumente für einen Service auf:

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

- Schließe einen bestimmten Service von den diagnostischen Informationen aus:

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

- Gib ein Timeout in Sekunden an (standardmäßig 10s):

```
dumpsys -t {{sekunden}}
```



# getprop

Zeige Informationen über Android-Systemeigenschaften.

Weitere Informationen: <https://manned.org/getprop>.

- Gib Informationen über Android-Systemeigenschaften aus:

```
getprop
```

- Gib Informationen über eine bestimmte Systemeigenschaft aus:

```
getprop {{eigenschaft}}
```

- Gib das SDK-API-Level aus:

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

- Gib die Android-Version aus:

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

- Gib das Modell des Geräts aus:

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

- Gib den OEM-Entsperrstatus aus:

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

- Gib die MAC-Adresse der WiFi-Karte aus:

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

# input

Sende Eventcodes oder Touchscreengesten an ein Android-Gerät.

Dieser Befehl kann nur mit **adb shell** verwendet werden.

Weitere Informationen: <https://developer.android.com/reference/android/view/KeyEvent.html#constants> 1.

- Sende einen Eventcode für die Eingabe eines einzelnen Zeichens an ein Gerät:

```
input keyevent {{eventcode}}
```

- Sende einen Text an ein Gerät (%s wird als Leerzeichen verwendet):

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

- Sende ein einzelnes Tippen auf den Touchscreen an ein Gerät:

```
input tap {{x_pos}} {{y_pos}}
```

- Sende ein Wischen über den Touchscreen an ein Gerät:

```
input swipe {{x_start}} {{y_start}} {{x_ende}} {{y_ende}}  
{{dauer_in_ms}}
```

- Sende ein langes Tippen auf den Touchscreen an ein Gerät:

```
input swipe {{x_pos}} {{y_pos}} {{x_pos}} {{y_pos}}  
{{dauer_in_ms}}
```

# logcat

Gib ein Protokoll aller Systemmeldungen aus.

Weitere Informationen: <https://developer.android.com/tools/logcat>.

- Gib ein Protokoll aller Systemmeldungen aus:

```
logcat
```

- Schreibe alle Systemmeldungen in eine Datei:

```
logcat -f {{pfad/zu/datei}}
```

- Gib Zeilen aus, die einem regulären Ausdruck entsprechen:

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

# pkg

Paketverwaltungs-Werkzeug für Termux.

Weitere Informationen: [https://wiki.termux.com/wiki/Package\\_Management](https://wiki.termux.com/wiki/Package_Management).

- Aktualisiere alle installierten Pakete:

```
pkg upgrade
```

- Installiere ein Paket:

```
pkg install {{paket}}
```

- Deinstalliere ein Paket:

```
pkg uninstall {{paket}}
```

- Installiere ein Paket erneut:

```
pkg reinstall {{paket}}
```

- Suche nach einem Paket:

```
pkg search {{paket}}
```

# pm

Zeige Informationen über Apps auf einem Android Gerät.

Weitere Informationen: <https://developer.android.com/tools/adb#pm>.

- Gib eine Liste aller installierten Apps aus:

```
pm list packages
```

- Gib eine Liste aller installierten System-Apps aus:

```
pm list packages -s
```

- Gib eine Liste aller installierten Apps von Drittanbietern aus:

```
pm list packages -3
```

- Gib eine Liste aller Apps, auf die ein bestimmtes Schlüsselwort zutrifft, aus:

```
pm list packages {{Schlüsselwort}}
```

- Gib den Pfad der APK einer bestimmten App aus:

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

# screencap

Nimmt ein Bildschirmfoto von einem Handy-Display auf.

Dieser Befehl kann nur über die 'adb shell' benutzt werden.

Weitere Informationen: <https://developer.android.com/tools/adb#screencap>.

- Nehme ein Bildschirmfoto auf:

```
screencap {{pfad/zu/datei}}
```

# settings

Verwalte Android-Systemeinstellungen.

Weitere Informationen: <https://adbinstaller.com/commands/adb-shell-settings-5b670d5ee7958178a2955536>.

- Gib eine Liste aller Einstellungen im Namespace `global` aus:

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

- Gib den Wert einer bestimmten Einstellung aus:

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

- Setze den Wert einer bestimmten Einstellung:

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

- Lösche eine bestimmte Einstellung:

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

# wm

Zeige Informationen über den Bildschirm eines Android-Geräts.

Dieser Befehl kann nur mit **adb shell** verwendet werden.

Weitere Informationen: <https://adbinstaller.com/commands/adb-shell-wm-5b672b17e7958178a2955538>.

- Gib die physische Größe des Bildschirms eines Geräts aus:

```
wm {{size}}
```

- Gib die physische Pixeldichte des Bildschirms eines Geräts aus:

```
wm {{density}}
```



Common

# Exclamation mark

Ein in Bash integriertes Kommando, welches durch einen Befehl aus dem Befehlsverlauf ersetzt wird.

Weitere Informationen: <https://www.gnu.org/software/bash/manual/bash.html#Event-Designators>.

- Ersetze `!!` durch den vorherigen Befehl und führe ihn mit `sudo` aus:

```
sudo !!
```

- Führe den Befehl Nummer `n` aus. Die Nummer eines Befehls kann mit `history` herausgefunden werden:

```
!{{n}}
```

- Führe den `n`'t letzten ausgeführten Befehl aus:

```
!-{{n}}
```

- Führe den letzten Befehl aus, der mit `zeichenkette` begann:

```
!{{zeichenkette}}
```

- Ersetze durch die Argumente des letzten Befehls:

```
{{befehl}} !*
```

# 7z

Ein Dateiarchivierer mit hoher Kompressionsrate.

Weitere Informationen: <https://manned.org/7z>.

- [a]rchiviere eine Datei oder ein Verzeichnis:

```
7z a {{pfad/zu/archiv.7z}} {{pfad/zu/datei_oder_verzeichnis}}
```

- Verschlüssele ein vorhandenes Archiv (einschließlich Dateinamen):

```
7z a {{pfad/zu/verschlüsselt.7z}} -p{{passwort}} -mhe=on  
{{pfad/zu/archiv.7z}}
```

- E[x]trahiere ein Archiv und behalte die originale Verzeichnisstruktur bei:

```
7z x {{pfad/zu/archiv.7z}}
```

- E[x]trahiere ein Archiv in ein bestimmtes Verzeichnis:

```
7z x {{pfad/zu/archiv.7z}} -o{{pfad/zu/verzeichnis}}
```

- E[x]trahiere ein Archiv nach stdout:

```
7z x {{pfad/zu/archiv.7z}} -so
```

- [a]rchiviere mit einem bestimmten Archivtyp:

```
7z a -t{{7z|bzip2|gzip|lzip|tar|zip}} {{pfad/zu/archiv}}  
{{pfad/zu/datei_oder_verzeichnis}}
```

- [l]iste den Inhalt einer Archivdatei auf:

```
7z l {{pfad/zu/archiv.7z}}
```

# 7za

Ein Dateiarchivierer mit hoher Kompressionsrate.

Eine alleinstehende Version von **7z** mit Unterstützung für neuere Archivtypen.

Weitere Informationen: <https://manned.org/7za>.

- [a]rchiviere eine Datei oder ein Verzeichnis:

```
7za a {{pfad/zu/archiv.7z}} {{pfad/zu/
datei_oder_verzeichnis}}
```

- Verschlüssele ein vorhandenes Archiv (einschließlich Dateinamen):

```
7za a {{pfad/zu/verschlüsselt.7z}} -p{{passwort}} -mhe=on
{{pfad/zu/archiv.7z}}
```

- E[x]trahiere ein Archiv und behalte die originale Verzeichnisstruktur bei:

```
7za x {{pfad/zu/archiv.7z}}
```

- E[x]trahiere ein Archiv in ein bestimmtes Verzeichnis:

```
7za x {{pfad/zu/archiv.7z}} -o{{pfad/zu/verzeichnis}}
```

- E[x]trahiere ein Archiv nach stdout:

```
7za x {{pfad/zu/archiv.7z}} -so
```

- [a]rchiviere mit einem bestimmten Archivtyp:

```
7za a -t{{7z|bzip2|gzip|lzip|tar|zip}} {{pfad/zu/archiv.7z}}
{{pfad/zu/datei_oder_verzeichnis}}
```

- [l]iste den Inhalt einer Archivdatei auf:

```
7za l {{pfad/zu/archiv.7z}}
```

# 7zr

Ein Dateiarchivierer mit hoher Kompressionsrate.

Eine alleinstehende Version von **7z**, die nur 7z Dateien unterstützt.

Weitere Informationen: <https://manned.org/7zr>.

- [a]rchiviere eine Datei oder ein Verzeichnis:

```
7zr a {{pfad/zu/archiv.7z}} {{pfad/zu/
datei_oder_verzeichnis}}
```

- Verschlüssele ein vorhandenes Archiv (einschließlich Dateinamen):

```
7zr a {{pfad/zu/verschlüsselt.7z}} -p{{passwort}} -mhe={{on}}
{{pfad/zu/archiv.7z}}
```

- E[x]trahiere ein Archiv und behalte die originale Verzeichnisstruktur bei:

```
7zr x {{pfad/zu/archiv.7z}}
```

- E[x]trahiere ein Archiv in ein bestimmtes Verzeichnis:

```
7zr x {{pfad/zu/archiv.7z}} -o{{pfad/zu/verzeichnis}}
```

- E[x]trahiere ein Archiv nach stdout:

```
7zr x {{pfad/zu/archiv.7z}} -so
```

- [l]iste den Inhalt einer Archivdatei auf:

```
7zr l {{pfad/zu/archiv.7z}}
```

[

Teste Dateitypen und vergleiche Werte.

Gibt 0 zurück, wenn der Ausdruck wahr ist und 1 wenn nicht.

Weitere Informationen: <https://www.gnu.org/software/bash/manual/bash.html#index-test>.

- Überprüfe, ob eine bestimmte Variable gleich oder ungleich einem bestimmten String ist:

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

- Überprüfe, ob eine Variable gleich/ungleich/größer/kleiner/größer-gleich oder kleiner-gleich als eine bestimmte Zahl ist:

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

- Überprüfe, ob eine bestimmte Variable einen nicht-leeren Wert hat:

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

- Überprüfe, ob eine bestimmte Variable einen leeren Wert hat:

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

- Überprüfe, ob eine bestimmte Datei existiert:

```
[ -f {{pfad/zu/datei}} ]
```

- Überprüfe, ob ein bestimmtes Verzeichnis existiert:

```
[ -d {{pfad/zu/verzeichnis}} ]
```

- Überprüfe, ob eine Datei oder ein Verzeichnis existiert:

```
[ -e {{pfad/zu/datei_oder_verzeichnis}} ]
```

# [[

Teste Dateitypen und vergleiche Werte.

Gibt 0 zurück, wenn der Ausdruck wahr ist und 1 wenn nicht.

Weitere Informationen: [https://www.gnu.org/software/bash/manual/bash.html#index-005b\\_005b](https://www.gnu.org/software/bash/manual/bash.html#index-005b_005b).

- Überprüfe, ob eine bestimmte Variable gleich oder ungleich einem bestimmten String ist:

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

- Überprüfe, ob ein bestimmter String einem bestimmten regulären Ausdruck oder glob-Muster entspricht:

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

- Überprüfe, ob eine Variable gleich/ungleich/größer/kleiner/größer-gleich oder kleiner-gleich als eine bestimmte Zahl ist:

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

- Überprüfe, ob eine bestimmte Variable einen nicht-leeren Wert hat:

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

- Überprüfe, ob eine bestimmte Variable einen leeren Wert hat:

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

- Überprüfe, ob eine bestimmte Datei existiert:

```
[[ -f {pfad/zu/datei} ]]
```

- Überprüfe, ob ein bestimmtes Verzeichnis existiert:

```
[[ -d {pfad/zu/verzeichnis} ]]
```

- Überprüfe, ob eine Datei oder ein Verzeichnis existiert:

```
[[ -e {pfad/zu/datei_oder_verzeichnis} ]]
```

# aapt

Android Asset Packaging Tool.

Kompiliere und verpacke die Ressourcen einer Android App.

Weitere Informationen: [https://elinux.org/Android\\_aapt](https://elinux.org/Android_aapt).

- Liste alle Dateien eines APK Archivs auf:

```
aapt list {{pfad/zu/app.apk}}
```

- Zeige die Metadaten einer App an (Version, Berechtigungen, usw.):

```
aapt dump badging {{pfad/zu/app.apk}}
```

- Erstelle ein neues APK Archiv mit den Dateien eines bestimmten Verzeichnisses:

```
aapt package -F {{pfad/zu/app.apk}} {{pfad/zu/verzeichnis}}
```



# ab

Apache HTTP server Benchmarking Tool.

Weitere Informationen: <https://httpd.apache.org/docs/current/programs/ab.html>.

- Sende 100 HTTP GET Anfragen an eine URL:

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

- Sende 100 HTTP GET Anfragen an eine URL, wovon bis zu 10 gleichzeitig bearbeitet werden:

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

- Sende 100 HTTP POST Anfragen mit der in der angegebenen Datei gespeicherten Payload an eine URL:

```
ab -n 100 -T {{application/json}} -p {{pfad/zu/datei.json}} {{url}}
```

- Wach halten:

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

- Lege die maximale Anzahl an Sekunden fest, die das Benchmarking dauern darf:

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

# adscript

Compiler für Adscript Dateien.

Weitere Informationen: <https://github.com/Amplus2/Adscript>.

- Kompiliere eine Datei zu einer Objektdatei:

```
adscript --output {{pfad/zu/datei.o}} {{pfad/zu/
quelldatei.adscript}}
```

- Kompiliere eine Datei zu einer ausführbaren Binärdatei:

```
adscript --executable --output {{pfad/zu/datei}} {{pfad/zu/
quelldatei.adscript}}
```

- Kompiliere eine Datei zu LLVM IR anstelle von nativem Maschinencode:

```
adscript --llvm-ir --output {{pfad/zu/datei.ll}} {{pfad/zu/
quelldatei.adscript}}
```

- Cross-kompiliere eine Datei zu einer Objektdatei für eine fremde CPU Architektur oder ein fremdes Betriebssystem:

```
adscript --target-triple {{i386-linux-elf}} --output {{pfad/
zu/datei.o}} {{pfad/zu/quelldatei.adscript}}
```

# age

Ein einfaches, modernes und sicheres Dateiverschlüsselungswerkzeug.

Weitere Informationen: <https://github.com/FiloSottile/age>.

- Generiere eine verschlüsselte Datei, die mit einer Passphrase entschlüsselt werden kann:

```
age --passphrase --output {{pfad/zu/verschlüsselter_datei}}  
{{pfad/zu/unverschlüsselter_datei}}
```

- Generiere ein Schlüsselpaar, speichere dabei den privaten Schlüssel in einer unverschlüsselten Datei und gib den öffentlichen Schlüssel zu `stdout` aus:

```
age-keygen --output {{pfad/zu/datei}}
```

- Verschlüsse eine Datei mit einem oder mehr öffentlichen Schlüsseln, die als Zeichenketten eingegeben werden:

```
age --recipient {{öffentlicher_schlüssel_1}} --recipient  
{{öffentlicher_schlüssel_2}} {{pfad/zu/  
unverschlüsselter_datei}} --output {{pfad/zu/  
verschlüsselter_datei}}
```

- Verschlüsse eine Datei mit einem oder mehr öffentlichen Schlüsseln, die in einer Empfängerdatei angegeben sind:

```
age --recipients-file {{pfad/zu/empfängerdatei}} {{pfad/zu/  
unverschlüsselter_datei}} --output {{pfad/zu/  
verschlüsselter_datei}}
```

- Entschlüsse eine Datei mit einer Passphrase:

```
age --decrypt --output {{pfad/zu/entschlüsselter_datei}}  
{{pfad/zu/verschlüsselter_datei}}
```

- Entschlüsse eine Datei mit einer privaten Schlüsseldatei:

```
age --decrypt --identity {{pfad/zu/privater_schlüsseldatei}}  
--output {{pfad/zu/entschlüsselter_datei}} {{pfad/zu/  
verschlüsselter_datei}}
```

# aircrack-ng

WEP und WPA/WPA2 Schlüssel im Kommunikationsaufbau knacken.

Teil des Aircrack-ng Softwarepakets.

Weitere Informationen: <https://www.aircrack-ng.org/doku.php?id=aircrack-ng>.

- Knacke Schlüssel von abgefangenen Paketen mithilfe von Wortlisten:

```
aircrack-ng -w {{pfad/zus/wortliste.txt}} {{pfad/zus/packetdatei.cap}}
```

- Knacke Schlüssel von abgefangenen Paketen mithilfe einer Wortliste und der (E)SSID des Access Points:

```
aircrack-ng -w {{pfad/zus/wortliste.txt}} -e {{essid}} {{pfad/zus/packetdatei.cap}}
```

- Knacke Schlüssel von abgefangenen Paketen mithilfe einer Wortliste und der MAC-Adresse des Access Points:

```
aircrack-ng -w {{pfad/zus/wortliste.txt}} --bssid {{mac}} {{pfad/zus/packetdatei.cap}}
```

# aireplay-ng

Pakete in ein WLAN injizieren.

Teil von **aircrack-ng**.

Weitere Informationen: <https://www.aircrack-ng.org/doku.php?id=aireplay-ng>.

- Sende eine bestimmten Anzahl an Dissoziation-Paketen mit der MAC-Adresse des Access Points, der MAC-Adresse des Clients und eines Interfaces:

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

# airmon-ng

Aktiveren des Überwachungsmodus auf Wireless Network Geräten.

Teil von **aircrack-ng**.

Weitere Informationen: <https://www.aircrack-ng.org/doku.php?id=airmon-ng>.

- Liste Wireless Geräte und deren Status auf:

```
sudo airmon-ng
```

- Aktiviere den Überwachungsmodus für ein bestimmtes Gerät:

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

- Kille störende Prozesse, die das Wireless Gerät verwenden:

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

- Deaktiviere den Überwachungsmodus für ein spezifisches Interface:

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

# airodump-ng

Erfasst Pakete und zeigt Informationen über drahtlose Netzwerke an.

Teil von **aircrack-ng**.

Weitere Informationen: <https://www.aircrack-ng.org/doku.php?id=airodump-ng>.

- Erfasse Pakete und zeige Informationen über ein drahtloses Netzwerk an:

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

- Erfasse Pakete und zeige Informationen über ein drahtloses Netzwerk anhand der MAC-Adresse und des Kanals an, und schreibe diese in eine Datei:

```
sudo airodump-ng --channel {{kanal}} --write {{pfad/zu/
datei}} --bssid {{mac}} {{interface}}
```

# alacritty

Plattformübergreifender, GPU-beschleunigter Terminalemulator.

Weitere Informationen: <https://github.com/alacritty/alacritty>.

- Öffne ein neues Alacritty-Fenster:

```
alacritty
```

- Starte Alacritty in einem bestimmten Arbeitsverzeichnis:

```
alacritty --working-directory {{pfad/zu/verzeichnis}}
```

- Führe einen Befehl in einem neuen Alacritty-Fenster aus:

```
alacritty -e {{befehl}}
```

- Gib eine alternative Konfigurations-Datei an (ist standardmäßig `$XDG_CONFIG_HOME/alacritty/alacritty.toml`):

```
alacritty --config-file {{pfad/zu/konfiguration.toml}}
```

- Starte mit aktiviertem Live-Konfigurations-Neuladen (kann auch standardmäßig in `alacritty.toml` eingestellt werden):

```
alacritty --live-config-reload --config-file {{pfad/zu/konfiguration.toml}}
```



# alias

Erstellt Aliasse - Alternative Namen für Befehle.

Aliasse laufen mit der aktuellen Shell-Sitzung ab, es sei denn, sie werden in der Konfigurationsdatei der Shell definiert, z.B. `~/.bashrc`.

Weitere Informationen: <https://tldp.org/LDP/abs/html/aliases.html>.

- Listet alle Aliasse auf:

```
alias
```

- Erstellt einen Alias:

```
alias {{alias}}="{{befehl}}"
```

- Zeigt den mit einem bestimmten Alias verknüpften Befehl an:

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

- Entferne einen Alias:

```
unalias {{alias}}
```

- Macht `rm` zu einem interaktiven Befehl:

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

- Erstellt den Alias `la` für `ls -a`:

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

# ani-cli

Ein Cli zum Stöbern und Anschauen von Anime.

Weitere Informationen: <https://github.com/pystardust/ani-cli>.

- Anime nach Namen suchen:

```
ani-cli "{{anime_name}}"
```

- Folge herunterladen:

```
ani-cli -d "{{anime_name}}"
```

- VLC als Medienspieler verwenden:

```
ani-cli -v "{{anime_name}}"
```

- Anzuschauende Episode angeben:

```
ani-cli -e {{episoden_nummer}} "{{anime_name}}"
```

- Anime aus Verlauf weiterschauen:

```
ani-cli -c
```

- Aktualisiere ani-cli:

```
ani-cli -U
```

# anki

Leistungsstarkes, intelligentes Lernkartenprogramm.

Weitere Informationen: <https://docs.ankiweb.net>.

- Starte anki:

```
anki
```

- Starte anki mit einem bestimmten Profil:

```
anki -p {{profil_name}}
```

- Starte anki in einer bestimmten Sprache:

```
anki -l {{sprache}}
```

- Starte anki von einem bestimmten Verzeichnis anstelle des Standardverzeichnis (~/.Anki):

```
anki -b {{pfad/zu/verzeichnis}}
```

# ansible-doc

Anzeigen von Informationen über die in den Ansible-Bibliotheken installierten Module.

Anzeigen einer knappen Auflistung von Plugins und deren Kurzbeschreibungen.

Weitere Informationen: <https://docs.ansible.com/ansible/latest/cli/ansible-doc.html>.

- Auflisten der verfügbaren Aktions-Plugin (Module):

```
ansible-doc --list
```

- Auflisten der verfügbare Plugins eines bestimmten Typs:

```
ansible-doc --type {{plugin_typ}} --list
```

- Anzeigen von Informationen über ein bestimmtes Aktions-Plugin (Module):

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

- Anzeigen von Informationen über ein Plugin mit einem bestimmten Typ:

```
ansible-doc --type {{plugin_typ}} {{plugin_name}}
```

- Anzeigen des Playbookausschnittes für ein Actions-Plugin (Module):

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

- Anzeigen von Informationen über ein Aktions-Plugin (Module) als JSON:

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

# ansible-galaxy

Erstelle und verwalte Ansible Rollen.

Weitere Informationen: <https://docs.ansible.com/ansible/latest/cli/ansible-galaxy.html>.

- Installiere eine Rolle:

```
ansible-galaxy install {{benutzername}}.{{rollenname}}
```

- Entferne eine Rolle:

```
ansible-galaxy remove {{benutzername}}.{{rollenname}}
```

- Liste installierte Rollen auf:

```
ansible-galaxy list
```

- Suche nach einer bestimmten Rolle:

```
ansible-galaxy search {{rollenname}}
```

- Erstelle eine neue Rolle:

```
ansible-galaxy init {{rollenname}}
```

- Erhalte Informationen über eine Benutzerrolle:

```
ansible-galaxy role info {{benutzername}}.{{rollenname}}
```

- Erhalte Informationen über eine Kollektion:

```
ansible-galaxy collection info {{benutzername}}.{{kollektionsname}}
```

# ansible-inventory

Anzeigen oder Ausgeben eines Ansible-Inventars.

Siehe auch: **ansible**.

Weitere Informationen: <https://docs.ansible.com/ansible/latest/cli/ansible-inventory.html>.

- Anzeigen des Standardinventars:

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

- Anzeigen eines Benutzerdefinierten Inventars:

```
ansible-inventory --list --inventory {{pfad/zu/
datei_oder_skript_oder_verzeichnis}}
```

- Anzeigen des Standardinventars in YAML:

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

- Ausgabe des Standardinventars in eine Datei:

```
ansible-inventory --list --output {{pfad/zu/datei}}
```

# ansible-playbook

In Playbook definierte Aufgaben auf entfernten Rechnern über SSH ausführen.

Weitere Informationen: <https://docs.ansible.com/ansible/latest/cli/ansible-playbook.html>.

- Führe Aufgaben im Playbook aus:

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

- Führe Aufgaben im Playbook mit benutzerdefiniertem Host-Bestand aus:

```
ansible-playbook {{playbook}} -i {{inventory_datei}}
```

- Führe Aufgaben im Playbook aus, wobei zusätzliche Variablen über die Befehlszeile definiert werden:

```
ansible-playbook {{playbook}} -e "{{variable1}}={{wert1}}  
{{variable2}}={{wert2}}"
```

- Führe Aufgaben in Playbook mit zusätzlichen Variablen aus, die in einer JSON-Datei definiert sind:

```
ansible-playbook {{playbook}} -e "@{{variablen.json}}"
```

- Führe Aufgaben im Playbook für die angegebenen Tags aus:

```
ansible-playbook {{playbook}} --tags {{tag1,tag2}}
```

- Führe Aufgaben in einem Playbook aus, die mit einer bestimmten Aufgabe beginnen:

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

# ansible-pull

Laden eines Ansible-Playbooks aus einem VCS-Repository und ausführen auf dem lokalen Host.

Weitere Informationen: <https://docs.ansible.com/ansible/latest/cli/ansible-pull.html>.

- Laden eines Playbooks aus einem VCS und ausführen des standardmässigen local.yml Playbooks:

```
ansible-pull -U {{repository_url}}
```

- Laden eines Playbooks aus einem VCS und ausführen eines spezifischen Playbooks:

```
ansible-pull -U {{repository_url}} {{playbook}}
```

- Laden eines Playbooks aus einem VCS unter Angabe eines bestimmten branches und ausführen eines spezifischen Playbooks:

```
ansible-pull -U {{repository_url}} -C {{branch}} {{playbook}}
```

- Laden eines Playbooks aus einem VCS und ausführen eines spezifischen Playbooks unter Angabe einer Hosts-Datei:

```
ansible-pull -U {{repository_url}} -i {{hosts_datei}}  
{{playbook}}
```



# ansible-vault

Verschlüsselt und entschlüsselt Werte, Datenstrukturen und Dateien innerhalb von Ansible-Projekten.

Weitere Informationen: [https://docs.ansible.com/ansible/latest/user\\_guide/vault.html#id17](https://docs.ansible.com/ansible/latest/user_guide/vault.html#id17).

- Erstelle eine neue verschlüsselte Vault-Datei mit einer Eingabeaufforderung für ein Passwort:

```
ansible-vault create {{vault_datei}}
```

- Erstelle eine neue verschlüsselte Vault-Datei mit einer Vault-Schlüsseldatei, um sie zu verschlüsseln:

```
ansible-vault create --vault-password-file {{schlüsseldatei}} {{vault_datei}}
```

- Verschlüsse eine vorhandene Datei mit einer optionalen Schlüsseldatei:

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

- Verschlüsse eine Zeichenfolge mit dem verschlüsselten Zeichenfolgenformat von Ansible, wobei interaktive Eingabeaufforderungen angezeigt werden:

```
ansible-vault encrypt_string
```

- Zeige eine verschlüsselte Datei an, wobei eine Kennwortdatei zum Entschlüsseln verwendet wird:

```
ansible-vault view --vault-password-file {{schlüsseldatei}} {{vault_datei}}
```

- Verschlüsse eine bereits verschlüsselte Vault Datei mit einer neuen Kennwortdatei neu:

```
ansible-vault rekey --vault-password-file {{alte_schlüsseldatei}} --new-vault-password-file {{neue_schlüsseldatei}} {{vault_datei}}
```

# ansible

Verwalte Computergruppen per Fernzugriff über SSH (Verwende die Datei **/etc/ansible/hosts**, um neue Gruppen/Hosts hinzuzufügen).

Manche Unterbefehle wie **ansible galaxy** sind separat dokumentiert.

Weitere Informationen: <https://www.ansible.com/>.

- Liste Hosts auf, die zu einer Gruppe gehören:

```
ansible {{gruppe}} --list-hosts
```

- Pinge eine Gruppe von Hosts an:

```
ansible {{gruppe}} -m ping
```

- Zeige Informationen über eine Gruppe von Hosts an:

```
ansible {{gruppe}} -m setup
```

- Führe einen Befehl auf einer Gruppe von Hosts aus:

```
ansible {{gruppe}} -m command -a '{{befehl}}'
```

- Führe einen Befehl mit administrativen Privilegien aus:

```
ansible {{gruppe}} --become --ask-become-pass -m command -a  
'{{befehl}}'
```

- Führe einen Befehl mit einer benutzerdefinierten Inventardatei aus:

```
ansible {{Gruppe}} -i {{inventardatei}} -m command -a  
'{{befehl}}'
```

- Liste alle Gruppen eines Inventars auf:

```
ansible localhost -m debug -a '{{var=groups.keys()}}'
```

# ansiweather

Ein Shell-Skript um die aktuellen Wetterbedingungen in einem Terminal anzuzeigen.

Weitere Informationen: <https://github.com/fcambus/ansiweather>.

- Zeige eine Vorhersage für die nächsten fünf Tage für Rzeszow, Polen in metrischen Einheiten an:

```
ansiweather -u {{metric}} -f {{5}} -l {{Rzeszow,PL}}
```

- Zeige eine Vorhersage mit Symbolen und Tageslichtdaten für den aktuellen Standort an:

```
ansiweather -s {{true}} -d {{true}}
```

- Zeige eine Vorhersage mit Wind- und Luftfeuchtigkeitsdaten für den aktuellen Standort an:

```
ansiweather -w {{true}} -h {{true}}
```

# ant

Apache Ant.

Tool zum Bauen und Verwalten von Projekten, die auf Java basieren.

Weitere Informationen: <https://ant.apache.org>.

- Baue ein Projekt mit der Standard build-Datei `build.xml`:

```
ant
```

- Baue ein Projekt mit einer anderen build-Datei als `build.xml`:

```
ant -f {{buildfile.xml}}
```

- Zeige Informationen über mögliche targets für dieses Projekt:

```
ant -p
```

- Zeige Debugging-Informationen:

```
ant -d
```

- Führe alle targets aus, die nicht von fehlgeschlagenen targets abhängen:

```
ant -k
```

# apropos

Durchsuche die Handbuchseiten nach Namen und Beschreibungen.

Weitere Informationen: <https://manned.org/apropos>.

- Suche nach einem Schlüsselwort mit einem regulären Ausdruck:

```
apropos {{regulären_ausdruck}}
```

- Suche ohne Beschränkung der Ausgabe auf die Terminal Breite:

```
apropos -l {{regulären_ausdruck}}
```

- Suche nach Seiten, die alle angegebenen Ausdrücke enthalten:

```
apropos {{regulären_ausdruck_1}} -a {{regulären_ausdruck_2}}  
-a {{regulären_ausdruck_3}}
```

# arp-scan

ARP Pakete an Host (spezifiziert mit IP Adresse oder Hostname) senden um das lokale Netzwerk zu scannen.

Weitere Informationen: <https://github.com/royhills/arp-scan>.

- Scanne das lokale Netzwerk:

```
arp-scan --localnet
```

- Scanne ein IP Netzwerk mit einer benutzerdefinierten Bitmaske:

```
arp-scan {{netzwerk_adresse}}/{{netzwerk_subnet}}
```

- Scanne ein IP Netzwerk innerhalb einer benutzerdefinierten Range:

```
arp-scan {{adresse_a}}-{{adresse_b}}
```

- Scanne ein IP Netzwerk mit einer benutzerdefinierten Netzmaske:

```
arp-scan {{adresse}}:{{netzwerkmaske}}
```

# arp

Den ARP Cache des Systems anzeigen und manipulieren.

Weitere Informationen: <https://manned.org/arp>.

- Zeige die aktuelle ARP Tabelle an:

```
arp -a
```

- Leere den gesamten Cache:

```
sudo arp -a -d
```

- Lösche einen spezifischen Eintrag in der Tabelle:

```
arp -d {{adresse}}
```

- Erstelle einen Eintrag in der ARP Tabelle:

```
arp -s {{adresse}} {{mac_adresse}}
```

# arping

Mit dem ARP Protokoll Hosts in einem Netzwerk entdecken und untersuchen.

Nützlich für die Entdeckung von MAC-Adressen.

Weitere Informationen: <https://github.com/ThomasHabets/arping>.

- Pinge einen Host mit ARP Request Paketen:

```
arping {{host_adresse}}
```

- Pinge einen Host auf einem spezifizierten Interface:

```
arping -I {{interface}} {{host_adresse}}
```

- Pinge einen Host und höre nach der ersten Antwort auf:

```
arping -f {{host_adresse}}
```

- Pinge einen Host für eine bestimmte Anzahl:

```
arping -c {{anzahl}} {{host_adresse}}
```

- Broadcaste ARP Request Pakete um die ARP Caches der Nachbarn zu aktualisieren:

```
arping -U {{broadcast_adresse}}
```

- Sende ARP Requests mit einem 3 Sekunden Timeout um duplizierte IP-Adressen im Netzwerk zu erkennen:

```
arping -D -w {{3}} {{adresse_zum_checken}}
```



# asdf

Verwalte installierte Versionen von verschiedenen Paketen.

Plugins (z.B. asdf-node) werden für spezifische Pakete verwendet.

Weitere Informationen: <https://asdf-vm.com>.

- Liste alle verfügbaren Plugins auf:

```
asdf plugin list all
```

- Installiere ein neues Plugin:

```
asdf plugin add {{name}}
```

- Liste alle verfügbaren Versionen für ein Paket auf:

```
asdf list all {{name}}
```

- Installiere eine spezifische Version eines Pakets:

```
asdf install {{name}} {{version}}
```

- Lege die globale Version für ein Paket fest:

```
asdf global {{name}} {{version}}
```

- Lege die lokale Version für ein Paket fest:

```
asdf local {{name}} {{version}}
```

# atom

Ein plattformübergreifender erweiterbarer Texteditor.

Erweiterungen werden durch **apm** verwaltet.

Weitere Informationen: <https://atom.io/>.

- Öffne eine Datei oder ein Verzeichnis:

```
atom {{pfad/zu/datei_oder_verzeichnis}}
```

- Öffne eine Datei oder ein Verzeichnis in einem neuen Fenster:

```
atom -n {{pfad/zu/datei_oder_verzeichnis}}
```

- Öffne eine Datei oder ein Verzeichnis in einem vorhandenen Fenster:

```
atom --add {{pfad/zu/datei_oder_verzeichnis}}
```

- Starte Atom im sicheren Modus (Es werden keine zusätzlichen Pakete geladen):

```
atom --safe
```

- Verhindert, dass sich Atom in den Hintergrund absetzt und hält es mit dem Terminal verbunden:

```
atom --foreground
```

- Wartet, bis Atom geschlossen wurde, bevor die Eingabeaufforderung wieder aktiv wird (Nützlich als `git commit` Editor):

```
atom --wait
```

# avrdude

Treiberprogramm für Atmel AVR Mikrocontroller-Programmierung.

Weitere Informationen: <https://www.nongnu.org/avrdude/>.

- Schreibt den Speicherinhalt eines AVR-Mikrocontrollers in eine Datei:

```
avrdude -p {{avr_gerät}} -c {{programmer}} -U flash:r:{{pfad/zu/datei.hex}}:i
```

- Schreibt den Inhalt einer Datei in einen AVR-Mikrocontroller:

```
avrdude -p {{avr_gerät}} -c {{programmer}} -U flash:w:{{pfad/zu/datei.hex}}
```

- Liste alle verfügbaren AVR-Geräte auf:

```
avrdude -p \?
```

- Liste alle verfügbaren AVR-Programmer auf:

```
avrdude -c \?
```

# aws backup

Einheitliches Backup-Service zum Schutz der Amazon Web Services und der damit verbundenen Daten.

Weitere Informationen: <https://docs.aws.amazon.com/cli/latest/reference/backup/index.html>.

- Gib Backup-Plan-Details für eine bestimmte Backup-Plan-ID aus:

```
aws backup get-backup-plan --backup-plan-id {{id}}
```

- Erstelle einen Backup-Plan unter Verwendung eines bestimmten Backup-Plan-Namens und von Backup-Regeln:

```
aws backup create-backup-plan --backup-plan {{plan}}
```

- Lösche einen bestimmten Backup-Plan:

```
aws backup delete-backup-plan --backup-plan-id {{id}}
```

- Gib eine Liste aller aktiven Backup-Pläne für das aktuelle Konto aus:

```
aws backup list-backup-plans
```

- Zeige Details über die Report-Aufträge an:

```
aws backup list-report-jobs
```

# aws codecommit

AWS CodeCommit ist ein verwalteter Versionskontrolldienst, der private Git Repositories hostet.

Weitere Informationen: <https://docs.aws.amazon.com/cli/latest/reference/codecommit/>.

- Zeige Hilfe für einen bestimmten Befehl an:

```
aws codecommit {{befehl}} help
```

- Zeige Hilfe an:

```
aws codecommit help
```

# aws cognito-idp

Verwalten des Amazon Cognito-Benutzerpools, seiner Benutzer und Gruppen mit der CLI.

Weitere Informationen: <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/cognito-idp/index.html>.

- Erstelle einen neuen Cognito-Benutzerpool:

```
aws cognito-idp create-user-pool --pool-name {{name}}
```

- Liste alle Benutzerpools auf:

```
aws cognito-idp list-user-pools --max-results {{10}}
```

- Lösche einen bestimmten Benutzerpool:

```
aws cognito-idp delete-user-pool --user-pool-id  
{{benutzerpool_id}}
```

- Erstelle einen Benutzer in einem bestimmten Pool:

```
aws cognito-idp admin-create-user --username {{benutzer}} --  
user-pool-id {{benutzerpool_id}}
```

- Liste die Benutzer eines bestimmten Pool auf:

```
aws cognito-idp list-users --user-pool-id {{benutzerpool_id}}
```

- Lösche einen Benutzer aus einem bestimmten Pool:

```
aws cognito-idp admin-delete-user --username {{benutzer}} --  
user-pool-id {{benutzerpool_id}}
```

# aws ec2

CLI für AWS EC2.

AWS EC2 stellt eine sichere und skalierbare Einheit in der AWS Cloud zur Verfügung, um ein schnelleres Entwickeln und Ausrollen von Software zu ermöglichen.

Weitere Informationen: <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/index.html>.

- Liste Informationen zu einer bestimmten Instanz auf:

```
aws ec2 describe-instances --instance-ids {{instanz_id}}
```

- Liste Informationen zu allen Instanzen auf:

```
aws ec2 describe-instances
```

- Liste Informationen zu allen EC2 Volumen auf:

```
aws ec2 describe-volumes
```

- Lösche ein EC2 Volumen:

```
aws ec2 delete-volume --volume-id {{volumen_id}}
```

- Erstelle einen Snapshot basierend auf einem EC2 Volumen:

```
aws ec2 create-snapshot --volume-id {{volumen_id}}
```

- Liste alle verfügbaren AMIs (Amazon Machine Images) auf:

```
aws ec2 describe-images
```

- Liste alle verfügbaren EC2 Befehle auf:

```
aws ec2 help
```

- Zeige Hilfe für bestimmte EC2 Unterbefehle an:

```
aws ec2 {{unterbefehl}} help
```

# aws-google-auth

CLI, um temporäre AWS credentials (STS) über Google Apps als Single Sign-On Dienstleister zu erhalten.

Weitere Informationen: <https://github.com/cevoaustralia/aws-google-auth>.

- Einloggen mit Google SSO über IDP- und SP-Kennung für die Dauer einer Stunde:

```
aws-google-auth -u {{beispiel@beispiel.com}} -I  
{{${G00GLE_IDP_ID}} -S {{${G00GLE_SP_ID}} -d {{3600}}
```

- Einloggen mit der Option eine Rolle auszuwählen (im Falle mehrerer verfügbarer SAML Rollen):

```
aws-google-auth -u {{beispiel@beispiel.com}} -I  
{{${G00GLE_IDP_ID}} -S {{${G00GLE_SP_ID}} -d {{3600}} -a
```

- Löse Aliasse von AWS Accounts auf:

```
aws-google-auth -u {{beispiel@beispiel.com}} -I  
{{${G00GLE_IDP_ID}} -S {{${G00GLE_SP_ID}} -d {{3600}} -a --  
resolve-aliases
```

- Zeige Hilfs-Informationen:

```
aws-google-auth -h
```



# aws iam

CLI für AWS IAM.

Weitere Informationen: <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/index.html>.

- Zeige die AWS IAM Hilfeseite (beinhaltet auch Hinweise für alle Unterbefehle):

```
aws iam help
```

- Liste aller Benutzer auf:

```
aws iam list-users
```

- Liste aller Richtlinien auf:

```
aws iam list-policies
```

- Liste aller Gruppen auf:

```
aws iam list-groups
```

- Liste aller Benutzer zu einer Gruppe auf:

```
aws iam get-group --group-name {{gruppe}}
```

- Liste einer IAM Richtlinie detailliert auf:

```
aws iam get-policy --policy-arn arn:aws:iam::aws:policy/{{richtlinie}}
```

- Liste alle Zugriffsschlüssel auf:

```
aws iam list-access-keys
```

- Liste alle Zugriffsschlüssel für einen Benutzer auf:

```
aws iam list-access-keys --user-name {{benutzername}}
```

# aws kinesis

Offizielles AWS CLI für die Amazon Kinesis-Streaming-Datenplattform.

Weitere Informationen: <https://docs.aws.amazon.com/cli/latest/reference/kinesis/index.html#cli-aws-kinesis>.

- Liste alle Streams auf:

```
aws kinesis list-streams
```

- Schreibe einen Datensatz in einen Kinesis Stream:

```
aws kinesis put-record --stream-name {{name}} --partition-key  
{{schlüssel}} --data {{base64_codierte_nachricht}}
```

- Schreibe einen Datensatz in einen Kinesis Stream mit base64 inline  
Encodierung:

```
aws kinesis put-record --stream-name {{name}} --partition-key  
{{schlüssel}} --data "$( echo "{{meine nachricht}}" | base64  
)"
```

- Liste alle verfügbaren Shards in einem Stream auf:

```
aws kinesis list-shards --stream-name {{name}}
```

- Rufe einen Shard Iterators auf, um diesen beginnend mit der ältesten  
Nachricht auszulesen:

```
aws kinesis get-shard-iterator --shard-iterator-type  
TRIM_HORIZON --stream-name {{name}} --shard-id {{id}}
```

- Lies einen Datensatz aus einem Shard über einen Shard Iterator:

```
aws kinesis get-records --shard-iterator {{iterator}}
```

# aws quicksight

CLI für AWS QuickSight.

Weitere Informationen: <https://docs.aws.amazon.com/cli/latest/reference/quicksight/>.

- Liste alle Datensätze auf:

```
aws quicksight list-data-sets --aws-account-id  
{{aws_account_id}}
```

- Liste alle Benutzer auf:

```
aws quicksight list-users --aws-account-id {{aws_account_id}}  
--namespace default
```

- Liste alle Gruppen auf:

```
aws quicksight list-groups --aws-account-id  
{{aws_account_id}} --namespace default
```

- Liste alle Dashboards auf:

```
aws quicksight list-dashboards --aws-account-id  
{{aws_account_id}}
```

- Liste einen Datensatz detailliert aus:

```
aws quicksight describe-data-set --aws-account-id  
{{aws_account_id}} --data-set-id {{datensatz_id}}
```

- Liste Zugangsberechtigungen zu einem Datensatz auf:

```
aws quicksight describe-data-set-permissions --aws-account-id  
{{aws_account_id}} --data-set-id {{datensatz_id}}
```

# aws s3

CLI für AWS S3. AWS S3 stellt Speicherplatz in der Cloud zur Verfügung.

Weitere Informationen: <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/index.html>.

- Liste alle Objekte in einem Bucket auf:

```
aws s3 ls {{bucket_name}}
```

- Synchronisiere Dateien und Verzeichnissen zu einem Bucket:

```
aws s3 sync {{pfad/zu/datei_oder_verzeichnis}} s3://  
{{bucket_name}}
```

- Synchronisiere Dateien und Verzeichnissen von einem Bucket:

```
aws s3 sync s3://{{bucket_name}} {{pfad/zu/ziel}}
```

- Synchronisiere Dateien und Verzeichnissen mit Ausnahmen:

```
aws s3 sync {{pfad/zu/datei_oder_verzeichnis}} s3://  
{{bucket_name}} --exclude {{pfad/zu/datei}} --exclude {{pfad/  
zu/verzeichnis}}/*
```

- Entferne ein Objekt von einem Bucket:

```
aws s3 rm s3://{{bucket}}/{{pfad/zu/datei}}
```

- Probelauf eines angegebenen Kommandos ohne dieses auszuführen:

```
aws s3 {{befehl}} --dryrun
```

# aws-vault

Ein Tresor für Entwicklungsumgebungen, um AWS Sicherheitsschlüssel sicher speichern und abrufen zu können.

Weitere Informationen: <https://github.com/99designs/aws-vault>.

- Füge einen Sicherheitsschlüssel als Profil zu einem Tresor hinzu:

```
aws-vault add {{profil}}
```

- Führe einen Befehl mit AWS Sicherheitsschlüsseln aus dem angegebenen Profil aus:

```
aws-vault exec {{profil}} -- {{aws s3 ls}}
```

- Öffne ein Browserfenster für den Login in die AWS Konsole:

```
aws-vault login {{profil}}
```

- Liste alle Profile zusammen mit deren Sicherheitsschlüsseln und Sitzungen auf:

```
aws-vault list
```

- Rotiere die AWS Sicherheitsschlüssel für ein Profil:

```
aws-vault rotate {{profil}}
```

- Entferne Sicherheitsschlüsseln eines Profils aus dem Tresor:

```
aws-vault remove {{profil}}
```

# aws

Das offizielle CLI für Amazon Web Services.

Manche Unterbefehle wie **aws s3** sind separat dokumentiert.

Weitere Informationen: <https://aws.amazon.com/cli>.

- Konfiguriere die AWS Kommandozeile:

```
aws configure wizard
```

- Konfiguriere die AWS Kommandozeile mithilfe von SSO:

```
aws configure sso
```

- Zeige Hilfe für ein Kommando an:

```
aws {{befehl}} help
```

- Zeige Informationen über die eigene angenommene Identität (häufig benutzt zur Fehlersuche):

```
aws sts get-caller-identity
```

- Liste alle AWS Ressourcen in einer Region mit YAML Formatierung auf:

```
aws dynamodb list-tables --region {{us-east-1}} --output yaml
```

- Erstelle einen IAM Benutzer mit Ausführungsassistent:

```
aws iam create-user --cli-auto-prompt
```

- Öffne einen Assistenten für eine AWS Ressource:

```
aws dynamodb wizard {{neue_tabelle}}
```

- Erstelle einen JSON CLI-Aufbau (hilfreich für Infrastruktur-Automation):

```
aws dynamodb update-table --generate-cli-skeleton
```

# awslogs

CLI um Log-Gruppen, Streams und Events von Amazon CloudWatch Logs abzurufen.

Weitere Informationen: <https://github.com/jorgebastida/awslogs>.

- Liste alle Log-Gruppen auf:

```
awslogs groups
```

- Liste alle bestehenden Streams einer angegebenen Loggruppe auf:

```
awslogs streams {/var/log/syslog}
```

- Rufe alle Logs für jegliche Streams in der angegebenen Log-Gruppe für die letzten 1 bis 2 Stunden ab:

```
awslogs get {/var/log/syslog} --start='{{2h ago}}' --end='{{1h ago}}'
```

- Rufe alle Logs für einen bestimmten CloudWatch-Logs Filter-Ausdruck ab:

```
awslogs get {/aws/lambda/meine_lambda_gruppe} --filter-pattern='{{ERROR}}'
```

- Beobachte Logs für jegliche Streams in der angegebenen Log-Gruppe:

```
awslogs get {/var/log/syslog} ALL --watch
```

# az config

Verwalten der Azure CLI-Konfiguration.

Teil von **azure-cli** (auch bekannt als **az**).

Weitere Informationen: <https://learn.microsoft.com/cli/azure/config>.

- Rufe alle Konfigurationen ab:

```
az config get
```

- Rufe alle Konfigurationen in einer Sektion ab:

```
az config get {{sektionsname}}
```

- Setze eine Konfiguration:

```
az config set {{konfigurationsname}}={{wert}}
```

- Hebe eine Konfiguration auf:

```
az config unset {{konfigurationsname}}
```



# az login

Melden Sie sich bei Azure an.

Teil von **azure-cli** (auch bekannt als **az**).

Weitere Informationen: [https://learn.microsoft.com/cli/azure/reference-index#az\\_login](https://learn.microsoft.com/cli/azure/reference-index#az_login).

- Melden Sie sich interaktiv an:

```
az login
```

- Melden Sie sich mit einem Dienstprinzipal mit dem geheimen Clientschlüssel an:

```
az login --service-principal --username {{http://azure-cli-  
service-principal}} --password {{secret}} --tenant  
{{someone.onmicrosoft.com}}
```

- Melden Sie sich mit einem Dienstprinzipal mithilfe des Clientzertifikats an:

```
az login --service-principal --username {{http://azure-cli-  
service-principal}} --password {{path/to/cert.pem}} --tenant  
{{someone.onmicrosoft.com}}
```

- Melden Sie sich mit der vom System zugewiesenen verwalteten Identität eines virtuellen Computers an:

```
az login --identity
```

- Melden Sie sich mit der vom Benutzer zugewiesenen verwalteten Identität eines virtuellen Computers an:

```
az login --identity --username /subscriptions/  
{{subscription_id}}/resourcegroups/{{my_rg}}/providers/  
Microsoft.ManagedIdentity/userAssignedIdentities/{{my_id}}
```

# az logout

Von Azure-Abonnements abmelden.

Teil von **azure-cli** (auch bekannt als **az**).

Weitere Informationen: [https://learn.microsoft.com/cli/azure/reference-index#az\\_logout](https://learn.microsoft.com/cli/azure/reference-index#az_logout).

- Melde das aktuelle aktive Konto ab:

```
az logout
```

- Melde einen spezifischen Benutzernamen ab:

```
az logout --username {{alias@somedomain.com}}
```

# babeld

Routing-Daemon für Babel, der Firewall-ähnliche Filter benutzt.

Weitere Informationen: <https://www.irif.fr/~jch/software/babel/babeld.html>.

- Starte `babeld` mit einer bestimmten Konfigurationsdatei:

```
babeld -c {{pfad/zu/babeld.conf}}
```

- Starte `babeld` mit mehreren Konfigurationsdateien (in der Reihenfolge des Einlesens):

```
babeld -c {{pfad/zu/ports.conf}} -c {{pfad/zu/filters.conf}}  
-c {{pfad/zu/interfaces.conf}}
```

- Starte `babeld` als Daemon:

```
babeld -D
```

- Starte `babeld` und übergib einen Konfigurationsbefehl:

```
babeld -C {{'redistribute metric 256'}}
```

- Starte `babeld` und gib an, auf welchen Schnittstellen gearbeitet werden soll:

```
babeld {{eth0}} {{eth1}} {{wlan0}}
```

# base64

Kodieren oder Dekodieren von Dateien oder Standardeingaben in/aus Base64, zur Standardausgabe.

Weitere Informationen: <https://www.gnu.org/software/coreutils/base64>.

- Kodiere den Inhalt einer Datei als base64 und schreibe das Ergebnis nach `stdout`:

```
base64 {{datei_name}}
```

- Dekodiere den Inhalt einer Datei als base64 und schreibe das Ergebnis nach `stdout`:

```
base64 --decode {{datei_name}}
```

- Kodiere von `stdin`:

```
{{befehl}} | base64
```

- Dekodiere von `stdin`:

```
{{befehl}} | base64 --decode
```

# basename

Entfernt führende Verzeichniskomponenten in einem Pfad.

Weitere Informationen: <https://www.gnu.org/software/coreutils/basename>.

- Ermittle den Dateinamen in einem Pfad:

```
basename {{pfad/zu/datei}}
```

- Ermittle den rechtesten Verzeichnisnamen in einem Pfad:

```
basename {{pfad/zu/verzeichnis/}}
```

- Ermittle den Dateinamen in einem Pfad und entferne den angegebenen Suffix aus diesem:

```
basename {{pfad/zu/datei}} {{suffix}}
```

# bash

Bourne-Again SHell.

**sh**-kompatibler Kommandozeilen-Interpreter.

Weitere Informationen: <https://www.gnu.org/software/bash/>.

- Interaktive Shell starten:

```
bash
```

- Führe einen Befehl aus:

```
bash -c "{{befehl}}"
```

- Führe Befehle aus einer Datei aus:

```
bash {{pfad/zu/datei.sh}}
```

- Führe Befehle aus einer Datei aus und protokolliere alle ausgeführten Befehle an das Terminal:

```
bash -x {{pfad/zu/datei.sh}}
```

- Führe Befehle aus einer Datei aus und stoppe beim ersten Fehler:

```
bash -e {{pfad/zu/datei.sh}}
```

- Führe Befehle von `stdin` aus:

```
bash -s
```

- Gib die Version von Bash aus (verwende `echo $BASH_VERSION`, um nur die Versionszeichenkette anzuzeigen):

```
bash --version
```

# bat

Ausgabe und Verkettung von einzelnen Dateien.

Ein **cat**-Ersatz mit Syntax-Hervorhebung und Git-Integration.

Weitere Informationen: <https://github.com/sharkdp/bat>.

- Gib den Inhalt einer Datei in `stdout` aus:

```
bat {{pfad/zu/datei}}
```

- Verkette mehrere Dateien zu einer Zielfeile:

```
bat {{pfad/zu/datei1}} {{pfad/zu/datei2}} > {{pfad/zu/ziel_datei}}
```

- Hänge mehrere Dateien an eine Zielfeile an:

```
bat {{pfad/zu/datei1}} {{pfad/zu/datei2}} >> {{pfad/zu/ziel_datei}}
```

- Nummeriere alle ausgegebenen Zeilen:

```
bat --number {{pfad/zu/datei}}
```

- Hebe die Syntax einer JSON-Datei hervor:

```
bat --language json {{pfad/zu/datei.json}}
```

- Zeige alle unterstützten Sprachen an:

```
bat --list-languages
```

# bgpgrep

Filtert und gibt BGP-Daten in einem MRT-Dump aus.

Kann mit gzip, bzip2 und xz komprimierte Dateien lesen.

Weitere Informationen: <https://codeberg.org/1414codeforge/ubgpsuite>.

- Gib alle Routen aus:

```
bgpgrep {{master6.mrt}}
```

- Gib alle von einem bestimmten Peer empfangenen Routen aus, bestimmt durch die AS-Nummer des Peers:

```
bgpgrep {{master4.mrt}} -peer {{64498}}
```

- Gib alle von einem bestimmten Peer empfangenen Routen aus, bestimmt durch die IP-Adresse des Peers:

```
bgpgrep {{master4.mrt.bz2}} -peer {{2001:db8:dead:cafe:acd::19e}}
```

- Gib alle Routen aus, die bestimmte ASNs in ihrem AS-Pfad haben:

```
bgpgrep {{master6.mrt.bz2}} -aspath '{{64498 64510}}'
```

- Gib Routen aus, die zu einer bestimmten Adresse führen:

```
bgpgrep {{master6.mrt.bz2}} -supernet  
'{{2001:db8:dead:cafe:aef::5}}'
```

- Gib alle Routen aus, die Communities von einem bestimmten AS haben:

```
bgpgrep {{master4.mrt}} -communities \( '{{64497}}:*' \)
```



# bird

BIRD Internet Routing Daemon.

Routing-Daemon mit Unterstützung für BGP, OSPF, Babel und weitere.

Weitere Informationen: <https://bird.network.cz/>.

- Starte `bird` mit einer bestimmten Konfigurationsdatei:

```
bird -c {{pfad/zu/bird.conf}}
```

- Starte `bird` als spezifischer Benutzer und Gruppe:

```
bird -u {{benutzername}} -g {{gruppe}}
```

# birdc

Bird remote control.

Kommandozeilenwerkzeug zum Abrufen von Informationen wie Routen von bird und zur Durchführung von Konfigurationen während der Laufzeit.

Weitere Informationen: <https://bird.network.cz/>.

- Öffne die remote control Konsole:

```
birdc
```

- Lade die Konfiguration neu, ohne Bird neu zu starten:

```
birdc configure
```

- Zeige den aktuellen Status von Bird an:

```
birdc show status
```

- Zeige alle konfigurierten Protokolle an:

```
birdc show protocols
```

- Zeige alle Details zu einem Protokoll an:

```
birdc show protocols {{upstream1}} all
```

- Zeige alle Routen an, die eine bestimmte AS-Nummer enthalten:

```
birdc "show route where bgp_path ~ [{{4242120045}}]"
```

- Zeige alle besten Routen an:

```
birdc show route primary
```

- Zeige alle Details zu allen Routen von einem bestimmten Präfix an:

```
birdc show route for {{fd00:/8}} all
```

# borg

Deduplizierendes Sicherungswerkzeug.

Erstellt lokale oder entfernte Sicherungen, die als Dateisysteme einhängbar sind.

Weitere Informationen: <https://borgbackup.readthedocs.io/en/stable/usage/general.html>.

- Initialisiere ein lokales Repository:

```
borg init {{pfad/zu/repo_verzeichnis}}
```

- Sichere ein Verzeichnis in das Repository und erstelle ein Archiv mit dem Namen "Montag":

```
borg create --progress {{pfad/zu/repo_verzeichnis}}::  
{{Montag}} {{pfad/zu/quell_verzeichnis}}
```

- Liste alle Archive in einem Repository auf:

```
borg list {{pfad/zu/repo_verzeichnis}}
```

- Extrahiere ein bestimmtes Verzeichnis aus dem "Montag"-Archiv in einem entfernten Repository, unter Ausschluss aller \*.ext-Dateien:

```
borg extract {{benutzer}}@{{host}}:{{pfad/zu/  
repo_verzeichnis}}:{{Montag}} {{pfad/zu/ziel_verzeichnis}}  
--exclude '{{*.ext}}'
```

- Bereinige ein Repository, indem alle Archive gelöscht werden, die älter als 7 Tage sind und Änderungen aufweisen:

```
borg prune --keep-within {{7d}} --list {{pfad/zu/  
repo_verzeichnis}}
```

- Hänge ein Repository als FUSE-Dateisystem ein:

```
borg mount {{pfad/zu/repo_verzeichnis}}:{{Montag}} {{pfad/  
zu/mountpoint}}
```

- Zeige Hilfe zur Erstellung von Archiven an:

```
borg create --help
```

# brew bundle

Bundler für Homebrew, Homebrew Cask und den Mac App Store.

Weitere Informationen: <https://github.com/Homebrew/homebrew-bundle>.

- Installiere Pakete aus einer Brewfile im aktuellen Pfad:

```
brew bundle
```

- Installiere Pakete aus einer bestimmten Brewfile:

```
brew bundle --file {{pfad/zu/brewfile}}
```

- Gib eine Liste mit allen installierten Paketen aus:

```
brew bundle dump
```

- Deinstalliere Pakete, die nicht in der Brewfile aufgelistet sind:

```
brew bundle cleanup --force
```

- Prüfe, ob von einem Paket die aktuellste Version installiert ist:

```
brew bundle check
```

- Zeige alle Pakete, die in der Brewfile aufgelistet sind:

```
brew bundle list --all
```

# bundler

Dieser Befehl ist ein Alias von **bundle**.

Weitere Informationen: <https://bundler.io/man/bundle.1.html>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr bundle
```

# caddy

Ein unternehmenstauglicher Open-Source-Webserver mit automatischem HTTPS, geschrieben in Go.

Weitere Informationen: <https://caddyserver.com>.

- Starte Caddy im Vordergrund:

```
caddy run
```

- Starte Caddy mit dem angegebenen Caddyfile:

```
caddy run --config {{pfad/zu/Caddyfile}}
```

- Starte Caddy im Hintergrund:

```
caddy start
```

- Stoppe einen im Hintergrund laufenden Caddy-Prozess:

```
caddy stop
```

- Führe einen einfachen Dateiserver auf dem angegebenen Port mit einer durchsuchbaren Oberfläche aus:

```
caddy file-server --listen :{{8000}} --browse
```

- Führe einen Reverse-Proxy-Server aus:

```
caddy reverse-proxy --from :{{80}} --to localhost:{{8000}}
```

# calc

Ein interaktiver Rechner im Terminal mit beliebiger Genauigkeit.

Weitere Informationen: <https://github.com/lcn2/calc>.

- Starte `calc` im interaktiven Modus:

```
calc
```

- Führe eine nicht-interaktive Berechnung durch:

```
calc '{{85 * (36 / 4}}}'
```

- Führe eine Berechnung durch ohne die Ausgabe zu formatieren (für das Benutzen mit Pipes):

```
calc -p '{{4/3 * pi() * 5^3}}'
```

- Führe eine Berechnung durch und wechsle dann in den [i]nteraktiven Modus:

```
calc -i '{{sqrt(2)}}'
```

- Starte `calc` in einem bestimmten Berechtigungs[m]odus (0 bis 7, standardmäßig 7):

```
calc -m {{mode}}
```

- Öffne eine Einführung zu `calc`:

```
calc help intro
```

- Öffne eine Übersicht von `calc`:

```
calc help overview
```

- Öffne die Bedienungsanleitung von `calc`:

```
calc help
```

# cargo

Verwalte Rust-Projekte und deren Abhängigkeiten (crates).

Manche Unterbefehle wie **cargo build** sind separat dokumentiert.

Weitere Informationen: <https://doc.rust-lang.org/cargo>.

- Suche nach Abhängigkeiten (crates):

```
cargo search {{suche}}
```

- Installiere eine Abhängigkeit (crate):

```
cargo install {{abhängigkeit}}
```

- Liste alle installierten Abhängigkeiten (crates) auf:

```
cargo install --list
```

- Erzeuge ein neues Rust-Projekt als Anwendung oder Bibliothek im aktuellen Verzeichnis:

```
cargo init --{{bin|lib}}
```

- Erzeuge ein neues Rust-Projekt als Anwendung oder Bibliothek im angegebenen Verzeichnis:

```
cargo new {{pfad/zu/verzeichnis}} --{{bin|lib}}
```

- Erstelle (bzw. kompiliere) ein Rust-Projekt im aktuellen Verzeichnis:

```
cargo build
```

- Erstelle (bzw. kompiliere) ein Rust-Projekt mit einer bestimmten Anzahl an Threads (standardmäßig die Anzahl der CPU-Kerne):

```
cargo build --jobs {{thread_anzahl}}
```



# cat

Verkette und gib einzelne oder mehrere Dateien aus.

Weitere Informationen: <https://manned.org/cat.1posix>.

- Gib den Inhalt einer Datei aus:

```
cat {{pfad/zu/datei}}
```

- Verkette mehrere Dateien und speichere das Ergebnis in einer neuen Datei:

```
cat {{pfad/zu/datei1 pfad/zu/datei2 ...}} > {{pfad/zu/ziel_datei}}
```

- Verkette mehrere Dateien und hänge das Ergebnis an eine Datei an:

```
cat {{pfad/zu/datei1 pfad/zu/datei2 ...}} >> {{pfad/zu/ziel_datei}}
```

- Kopiere den Inhalt einer Datei in eine Ausgabedatei ohne zu puffern:

```
cat -u {{/dev/tty12}} > {{/dev/tty13}}
```

- Schreibe `stdin` in eine Datei:

```
cat - > {{pfad/zu/datei}}
```

# cd

Ändere das aktuelle Arbeitsverzeichnis.

Weitere Informationen: <https://manned.org/cd>.

- Wechsle in das angegebene Verzeichnis:

```
cd {{pfad/zu/verzeichnis}}
```

- Wechsle zum Verzeichnis über dem aktuellen Verzeichnis:

```
cd ..
```

- Wechsle in das Home-Verzeichnis des aktuellen Benutzers:

```
cd
```

- Wechsle in das Home-Verzeichnis eines bestimmten Benutzers:

```
cd ~{{benutzername}}
```

- Wechsle zum zuletzt gewählten Verzeichnis:

```
cd -
```

- Wechseln in das Stammverzeichnis:

```
cd /
```

# chmod

Ändere die Zugriffsberechtigungen einer Datei oder eines Verzeichnisses.

Weitere Informationen: <https://www.gnu.org/software/coreutils/chmod>.

- Gib dem Besitzer einer Datei ([u]ser) das Recht, sie auszuführen (e[x]ecute):

```
chmod u+x {{pfad/zu/datei}}
```

- Gib dem Besitzer ([u]ser) Rechte zum Lesen ([r]ead) und Schreiben ([w]rite) einer Datei / einem Verzeichnis:

```
chmod u+rw {{pfad/zu/datei_oder_verzeichnis}}
```

- Entferne die Ausführrechte (e[x]ecute) der Besitzer[g]ruppe:

```
chmod g-x {{pfad/zu/datei}}
```

- Gib [a]llen Benutzern Rechte zum Lesen ([r]ead) und Ausführen (e[x]ecute) einer Datei:

```
chmod a+rx {{pfad/zu/datei}}
```

- Gib anderen ([o]thers) (nicht in der Besitzer[g]ruppe) die gleichen Rechte wie der Besitzer[g]ruppe:

```
chmod o=g {{pfad/zu/datei}}
```

- Entferne alle Rechte der anderen ([o]thers):

```
chmod o= {{pfad/zu/datei}}
```

- Ändere Rechte rekursiv, indem der Besitzer[g]ruppe und anderen ([o]thers) die Rechte zum Schreiben ([w]rite) geben werden:

```
chmod -R g+w,o+w {{pfad/zu/verzeichnis}}
```

- Gib [a]llen Benutzern rekursiv Rechte zum Lesen ([r]ead) von Dateien und Ausführen (e[X]ecute) von Unterverzeichnissen innerhalb eines Verzeichnisses:

```
chmod -R a+rX {{pfad/zu/verzeichnis}}
```

# chown

Ändere den Besitzer und die Besitzergruppe von Dateien und Verzeichnissen.

Weitere Informationen: <https://www.gnu.org/software/coreutils/chown>.

- Ändere den Besitzer einer Datei/eines Verzeichnisses:

```
chown {{benutzer}} {{pfad/zu/datei_oder_verzeichnis}}
```

- Ändere den Besitzer und die Besitzergruppe einer Datei/eines Verzeichnisses:

```
chown {{benutzer}}:{{gruppe}} {{pfad/zu/
datei_oder_verzeichnis}}
```

- Ändere den Besitzer eines Verzeichnisses rekursiv:

```
chown -R {{benutzer}} {{pfad/zu/verzeichnis}}
```

- Ändere den Besitzer eines symbolischen Links:

```
chown -h {{benutzer}} {{pfad/zu/symlink}}
```

- Ändere den Besitzer einer Datei/eines Verzeichnisses, damit sie/es mit einer Referenzdatei übereinstimmt:

```
chown --reference {{pfad/zu/referenzdatei_oder_verzeichnis}}
{{pfad/zu/datei_oder_verzeichnis}}
```

# chromium

Open-Source-Webbrowser von Google.

Weitere Informationen: <https://www.chromium.org/developers/how-tos/run-chromium-with-flags/>.

- Öffne eine bestimmte Datei oder URL:

```
chromium {{https://beispiel.com|pfad/zu/datei.html}}
```

- Öffne eine URL im Inkognito-Modus:

```
chromium --incognito {{beispiel.com}}
```

- Öffne eine URL in einem neuen Fenster:

```
chromium --new-window {{beispiel.com}}
```

- Öffne eine URL im Anwendungsmodus (ohne Symbolleisten, Suchleiste, Schaltflächen usw.):

```
chromium --app={{https://beispiel.com}}
```

- Öffne eine URL und verwende einen Proxy-Server:

```
chromium --proxy-server="{{socks5://hostname:66}}"  
{{beispiel.com}}
```

- Öffne Chromium mit einem eigenen Profil-Verzeichnis:

```
chromium --user-data-dir={{pfad/zu/verzeichnis}}
```

- Öffne Chromium ohne CORS-Verifizierung (nützlich, um eine API zu testen):

```
chromium --user-data-dir={{pfad/zu/verzeichnis}} --disable-  
web-security
```

- Öffne Chromium mit einem DevTools-Fenster für jeden geöffneten Tab:

```
chromium --auto-open-devtools-for-tabs
```

# chroot

Führe einen Befehl oder eine interaktive Shell mit einem speziellen root-Verzeichnis aus.

Weitere Informationen: <https://www.gnu.org/software/coreutils/chroot>.

- Führe einen Befehl mit einem neuen root-Verzeichnis aus:

```
chroot {{pfad/zu/root_verzeichnis}} {{befehl}}
```

- Lege einen Benutzer und eine Gruppe (ID oder Name) fest, der benutzt werden soll:

```
chroot --userspec={{benutzer:gruppe}}
```

# chsh

Ändere die Login-Shell eines Benutzers.

Weitere Informationen: <https://manned.org/chsh>.

- Ändere die Login-Shell des aktuellen Benutzers interaktiv:

```
chsh
```

- Ändere die Login-Shell des aktuellen Benutzers:

```
chsh -s {{pfad/zu/shell}}
```

- Ändere die Login-Shell eines Benutzers:

```
chsh -s {{pfad/zu/shell}} {{benutzername}}
```

- Liste alle verfügbaren Shells auf:

```
chsh -l
```

# ClamAV

Dieser Befehl ist ein Alias von **clamscan**.

Weitere Informationen: <https://www.clamav.net>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr clamscan
```



# clang++

Kompiliert C++ Quelldateien.

Teil von LLVM.

Weitere Informationen: <https://clang.llvm.org>.

- Kompiliere eine Quelldatei in eine ausführbare Binärdatei:

```
clang++ {{pfad/zu/quelldatei.cpp}} -o {{pfad/zu/binärdatei}}
```

- Zeige geläufige Fehler und Warnungen an:

```
clang++ {{pfad/zu/quelldatei.cpp}} -Wall -o {{pfad/zu/binärdatei}}
```

- Wähle einen Sprachstandard für die Kompilation aus:

```
clang++ {{pfad/zu/quelldatei.cpp}} -std={{c++20}} -o {{pfad/zu/binärdatei}}
```

- Binde Bibliotheken, die sich an einem anderen Pfad als die Quelldatei befinden, ein:

```
clang++ {{pfad/zu/quelldatei.cpp}} -o {{pfad/zu/binärdatei}} -I{{pfad/zu/headerdatei}} -L{{pfad/zu/bibliothek}} -l{{bibliotheksname}}
```

- Kompiliere eine Quelldatei zu LLVM Intermediate Representation (IR):

```
clang++ -S -emit-llvm {{pfad/zu/quelldatei.cpp}} -o {{pfad/zu/ir_datei.ll}}
```

# clang-cpp

Dieser Befehl ist ein Alias von **clang++**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr clang++
```

# clang-format

Programm zum Auto-Formatieren von C/C++/Java/JavaScript/Objective-C/Protobuf/C#-Code.

Weitere Informationen: <https://clang.llvm.org/docs/ClangFormat.html>.

- Formatiere eine Datei und schreibe das Ergebnis nach `stdout`:

```
clang-format {{pfad/zu/quelldatei.cpp}}
```

- Überschreibe eine Datei mit ihrem formatierten Inhalt:

```
clang-format -i {{pfad/zu/quelldatei.cpp}}
```

- Formatiere eine Datei mit einem bestimmten Code-Stil:

```
clang-format --style {{LLVM|GNU|Google|Chromium|Microsoft|  
Mozilla|WebKit}} {{pfad/zu/quelldatei.cpp}}
```

- Formatiere eine Datei mit der `.clang-format`-Datei aus einem der Überverzeichnisse der Quelldatei:

```
clang-format --style=file {{pfad/zu/quelldatei.cpp}}
```

- Generiere eine eigene `.clang-format`-Datei:

```
clang-format --style {{LLVM|GNU|Google|Chromium|Microsoft|  
Mozilla|WebKit}} --dump-config > {{.clang-format}}
```

# clang-tidy

Ein LLVM-basierter C/C++ Linter zum Finden von Stilverletzungen, Bugs und Sicherheitsmängeln durch statische Codeanalyse.

Weitere Informationen: <https://clang.llvm.org/extra/clang-tidy/>.

- Führe die Standard-Checks für eine Quelldatei aus:

```
clang-tidy {{pfad/zu/quelldatei.cpp}}
```

- Prüfe nur ob eine Datei den `cppcoreguidelines` Checks entspricht:

```
clang-tidy {{pfad/zu/quelldatei.cpp}} -checks={{-  
*,cppcoreguidelines-*}}
```

- Liste alle verfügbaren Checks auf:

```
clang-tidy -checks={{*}} -list-checks
```

- Lege defines und includes als Kompilierungsoptionen fest (nach `--`):

```
clang-tidy {{pfad/zu/quelldatei.cpp}} -- -I{{mein_projekt/  
include}} -D{{definitions}}
```

# clang

Compiler für C, C++ und Objective-C Quelldateien. Kann als Ersatz für GCC genutzt werden.

Weitere Informationen: <https://clang.llvm.org/docs/ClangCommandLineReference.html>.

- Kompiliere eine Quelldatei in eine ausführbare Binärdatei:

```
clang {{pfad/zu/quelldatei.c}} -o {{pfad/zu/binärdatei}}
```

- Zeige geläufige Fehler und Warnungen an:

```
clang {{pfad/zu/quelldatei.c}} -Wall -o {{pfad/zu/binärdatei}}
```

- Binde Bibliotheken, die sich an einem anderen Pfad als die Quelldatei befinden, ein:

```
clang {{pfad/zu/quelldatei.c}} -o {{pfad/zu/binärdatei}} -I{{pfad/zu/headerdatei}} -L{{pfad/zu/bibliothek}} -l{{bibliotheks_name}}
```

- Kompiliere eine Quelldatei zu LLVM Intermediate Representation (IR):

```
clang -S -emit-llvm {{pfad/zu/quelldatei.c}} -o {{pfad/zu/ir_datei.ll}}
```

- Kompiliere eine Quelldatei, ohne zu Linken:

```
clang -c {{pfad/zu/quelldatei.c}}
```

# clear

Leert den Bildschirm eines Terminals.

Weitere Informationen: <https://manned.org/clear>.

- Leere den Bildschirm (äquivalent zu Strg+L in einer Bash Shell):

```
clear
```

- Leere den Bildschirm, aber erhalte den Rückscroll-Puffer des Terminals:

```
clear -x
```

- Lege den Typ des zu leerenden Terminals fest (Standardwert ist die Umgebungsvariable \$TERM):

```
clear -T {{typ_des_terminals}}
```

- Zeige die Version von `ncurses` an, die von `clear` benutzt wird:

```
clear -V
```

# clojure

Dieser Befehl ist ein Alias von **clj**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr clj
```

# cmake

Plattformübergreifendes Build-Automatisierungs-System, das Vorlagen für native Build-Systeme erzeugt.

Weitere Informationen: <https://cmake.org/cmake/help/latest/manual/cmake.1.html>.

- Erzeuge eine Build-Vorlage im aktuellen Verzeichnis mit `CMakeLists.txt` eines Projektordners:

```
cmake {{pfad/zu/projektordner}}
```

- Erzeuge eine Build-Vorlage mit der Build-Art Release:

```
cmake {{pfad/zu/projektordner}} -D  
{{CMAKE_BUILD_TYPE=Release}}
```

- Benutze eine generierte Vorlage, um Artefakte zu erzeugen:

```
cmake --build {{pfad/zu/build_verzeichnis}}
```

- Installiere die Build-Artefakte in `/usr/local/` und entferne Debugsymbole:

```
cmake --install {{pfad/zu/build_verzeichnis}} --strip
```

- Installiere die Build-Artefakte mit einem eigenen Präfix für Pfade:

```
cmake --install {{pfad/zu/build_verzeichnis}} --strip --  
prefix {{pfad/zu/verzeichnis}}
```

- Führe ein bestimmtes Build-Ziel aus:

```
cmake --build {{pfad/zu/build_verzeichnis}} --target  
{{zielname}}
```



# code

Visual Studio Code.

Weitere Informationen: <https://github.com/microsoft/vscode>.

- Öffne Visual Studio Code:

```
code
```

- Öffne bestimmte Dateien und/oder Verzeichnisse:

```
code {{pfad/zu/datei_oder_verzeichnis1 pfad/zu/
datei_oder_verzeichnis2 ...}}
```

- Vergleiche zwei bestimmte Dateien:

```
code --diff {{pfad/zu/datei1}} {{pfad/zu/datei2}}
```

- Öffne bestimmte Dateien und/oder Verzeichnisse in einem neuen Fenster:

```
code --new-window {{pfad/zu/datei_oder_verzeichnis1 pfad/zu/
datei_oder_verzeichnis2 ...}}
```

- Installiere oder lösche bestimmte Erweiterung:

```
code --{{install|uninstall}}-extension
{{herausgeber.erweiterung}}
```

- Liste alle installierten Erweiterungen auf:

```
code --list-extensions
```

- Liste alle installierten Erweiterungen und deren Version auf:

```
code --list-extensions --show-versions
```

- Starte Visual Studio Code als Superuser und speichere Benutzerdaten in einem bestimmten Verzeichnis:

```
sudo code --user-data-dir {{pfad/zu/verzeichnis}}
```

# cola

Dieser Befehl ist ein Alias von **git-cola**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr git-cola
```

# conda create

Erstelle neue Conda-Umgebungen.

Weitere Informationen: <https://docs.conda.io/projects/conda/en/latest/commands/create.html>.

- Erstelle eine neue Umgebung mit dem Namen `py39` und installiere Python 3.9 und NumPy v1.11 (oder höher) darin:

```
conda create --yes --name {{py39}} python={{3.9}}  
"{{numpy>=1.11}}"
```

- Erstelle eine exakte Kopie einer Umgebung:

```
conda create --clone {{py39}} --name {{py39-copy}}
```

- Erstelle eine neue Umgebung mit gegebenem Namen und den zu installierenden Paketen:

```
conda create --name {{umgebungsname}} {{paketname}}
```

# conda

Eine Paket-, Abhängigkeits- und Umgebungsverwaltung für beliebige Programmiersprachen.

Manche Unterbefehle wie **conda create** sind separat dokumentiert.

Weitere Informationen: <https://github.com/conda/conda>.

- Erstelle eine neue Umgebung mit den zu installierenden Paketen:

```
conda create --name {{umgebungsname}} {{python=3.9  
matplotlib}}
```

- Liste alle Umgebungen auf:

```
conda info --envs
```

- Lade eine Umgebung:

```
conda activate {{umgebungs_name}}
```

- Entlade eine Umgebung:

```
conda deactivate
```

- Lösche eine Umgebung (entferne alle Pakete):

```
conda remove --name {{umgebungsname}} --all
```

- Installiere Pakete in die derzeit geladene Umgebung:

```
conda install {{python=3.4 numpy}}
```

- Liste alle installierten Pakete in der derzeit geladenen Umgebung auf:

```
conda list
```

- Lösche alle ungenutzten Pakete und leere den Cache:

```
conda clean --all
```

# cp

Kopiere Dateien und Verzeichnisse.

Weitere Informationen: <https://www.gnu.org/software/coreutils/cp>.

- Kopiere eine Datei an einen anderen Ort:

```
cp {{pfad/zu/datei}} {{pfad/zu/kopie}}
```

- Kopiere eine Datei an einen anderen Ort und behalte den Dateinamen:

```
cp {{pfad/zu/datei}} {{pfad/zu/zielverzeichnis}}
```

- Kopiere ein Verzeichnis rekursiv (falls der Zielort bereits existiert, wird das Verzeichnis in das Zielverzeichnis kopiert):

```
cp -r {{pfad/zu/verzeichnis}} {{pfad/zu/zielverzeichnis}}
```

- Kopiere ein Verzeichnis rekursiv und gib alle Dateien aus, während sie kopiert werden:

```
cp -vr {{pfad/zu/verzeichnis}} {{pfad/zu/zielverzeichnis}}
```

- Kopiere alle Textdateien in einem Verzeichnis und warte auf eine Bestätigung, falls eine Datei überschrieben werden soll:

```
cp -i {{/pfad/zu/*.txt}} {{pfad/zu/zielverzeichnis}}
```

- Folge symbolischen Links vor dem Kopieren:

```
cp -L {{link}} {{pfad/zu/zielverzeichnis}}
```

# cradle deploy

Verwalte Cradle Implementierungen.

Weitere Informationen: <https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#deploy>.

- Implementiere Cradle auf einem Server:

```
cradle deploy production
```

- Implementiere statische Anlagen zu Amazon S3:

```
cradle deploy s3
```

- Implementiere statische Anlagen inklusive den Yarn Komponenten:

```
cradle deploy s3 --include-yarn
```

- Implementiere statische Anlagen inklusive dem "upload" Verzeichnis:

```
cradle deploy s3 --include-upload
```

# cradle elastic

Verwalte Elasticsearch Instanzen einer Cradle Instanz.

Weitere Informationen: <https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#elastic>.

- Entleere den Elasticsearch Index:

```
cradle elastic flush
```

- Entleere den Elasticsearch Index für ein bestimmtes Paket:

```
cradle elastic flush {{paket}}
```

- Sende ein Elasticsearch Schema ab:

```
cradle elastic map
```

- Sende ein Elasticsearch Schema für ein bestimmtes Paket ab:

```
cradle elastic map {{paket}}
```

- Befülle die Elasticsearch Indizes für alle Pakete:

```
cradle elastic populate
```

- Befülle die Elasticsearch Indizes für ein bestimmtes Paket:

```
cradle elastic populate {{paket}}
```

# cradle install

Installiere Cradle PHP Framework Komponenten.

Weitere Informationen: <https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#install>.

- Installiere Cradle Komponenten mithilfe eines Dialogs:

```
cradle install
```

- Installiere Cradle Komponenten gewaltsam:

```
cradle install --force
```

- Überspringe SQL Migrationen:

```
cradle install --skip-sql
```

- Überspringe Paket-Aktualisierungen:

```
cradle install --skip-versioning
```

- Zeige Details über eine benutzer-spezifische Datenbank:

```
cradle install -h {{hostname}} -u {{benutzer}} -p  
{{password}}
```



# cradle package

Verwalte Pakete für Cradle Instanzen.

Weitere Informationen: <https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#package>.

- Liste aller verfügbaren Pakete auf:

```
cradle package list
```

- Suche nach einem Paket:

```
cradle package search {{paket}}
```

- Installiere ein Paket von Packagist:

```
cradle package install {{paket}}
```

- Installiere eine bestimmte Version eines Pakets:

```
cradle package install {{paket}} {{version}}
```

- Aktualisiere ein Paket:

```
cradle package update {{paket}}
```

- Aktualisiere ein Paket zu einer bestimmten Paket-Version:

```
cradle package update {{paket}} {{version}}
```

- Entferne ein Paket:

```
cradle package remove {{paket}}
```

# cradle sql

Verwalte Cradle SQL Datenbanken.

Weitere Informationen: <https://cradlephp.github.io/docs/3.B.-Reference-Command-Line-Tools.html#sql>.

- Generiere ein neues Datenbank-Schema:

```
cradle sql build
```

- Generiere ein neues Datenbank-Schema für ein bestimmtes Paket:

```
cradle sql build {{paket}}
```

- Entleere die gesamte Datenbank:

```
cradle sql flush
```

- Entleere die Datenbank für ein bestimmtes Paket:

```
cradle sql flush {{paket}}
```

- Befülle die Tabellen für alle Pakete:

```
cradle sql populate
```

- Befülle die Tabellen für ein bestimmtes Paket:

```
cradle sql populate {{paket}}
```

# cradle

Das Cradle PHP Framework.

Manche Unterbefehle wie **cradle install** sind separat dokumentiert.

Weitere Informationen: <https://cradlephp.github.io>.

- Stelle eine Verbindung zu einem Server her:

```
cradle connect {{server}}
```

- Zeige die Hilfe-Seite an:

```
cradle help
```

- Zeige die Hilfe-Seite für einen bestimmten Befehl:

```
cradle {{befehl}} help
```

- Führe einen Cradle-Befehl aus:

```
cradle {{befehl}}
```

# cron

Dieser Befehl ist ein Alias von **crontab**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr crontab
```

# csvsql

Generiere SQL-Anweisungen für eine CSV-Datei oder führe diese Anweisungen direkt in einer Datenbank aus.

Teil von csvkit.

Weitere Informationen: <https://csvkit.readthedocs.io/en/latest/scripts/csvsql.html>.

- Generiere eine `CREATE TABLE`-SQL-Anweisung für eine CSV-Datei:

```
csvsql {{pfad/zu/datei.csv}}
```

- Importiere eine CSV-Datei in eine SQL-Datenbank:

```
csvsql --insert --db "{{mysql://benutzer:passwort@host/datenbank}}" {{pfad/zu/datei.csv}}
```

- Führe eine SQL-Abfrage auf einer CSV-Datei aus:

```
csvsql --query "{{select * from 'datei'}}" {{datei.csv}}
```

# curl

Überträgt Daten von oder zu einem Server.

Unterstützt die meisten Protokolle, inklusive HTTP, FTP und POP3.

Weitere Informationen: <https://curl.se/docs/manpage.html>.

- Lade den Inhalt einer URL in eine Datei:

```
curl {{http://beispiel.de}} --output {{pfad/zu/datei}}
```

- Lade eine Datei von einer URL herunter:

```
curl --remote-name {{http://beispiel.de/datei}}
```

- Lade eine Datei herunter, folge Weiterleitungen und setze vergangene Dateitransfers automatisch fort:

```
curl --fail --remote-name --location --continue-at -  
{{http://beispiel.de/datei}}
```

- Sende formular-codierte Daten (POST Anfragen des Typs `application/x-www-form-urlencoded`). Benutze `--data @dateiname` oder `--data @'-'`, um von STDIN zu lesen:

```
curl --data {'name=karl-dieter'}} {{http://beispiel.de/  
formular}}
```

- Sende eine Anfrage mit einem extra Header mit einer eigenen HTTP-Methode:

```
curl --header {'X-Mein-Header: 123'}} --request {{PUT}}  
{{http://beispiel.de}}
```

- Sende Daten im JSON-Format und lege den geeigneten Inhaltstyp-Header fest:

```
curl --data {'{"name":"karl-dieter"}}' --header {'Content-  
Type: application/json'}} {{http://beispiel.de/benutzer/  
1234}}
```

- Übergib einen Benutzernamen und frage nach einem Passwort für die Server-Authentifizierung:

```
curl --user {{benutzername}} {{http://beispiel.de}}
```

- Übergib Client-Zertifikat und -Schlüssel für eine Ressource und überspringe die Zertifikatsüberprüfung:

```
curl --cert {{client.pem}} --key {{key.pem}} --insecure  
{{https://beispiel.de}}
```

# cut

Schneide Felder von **stdin** oder einer Datei aus.

Weitere Informationen: <https://www.gnu.org/software/coreutils/cut>.

- Schneide bestimmte Zeichen oder einen Bereich von Feldern jeder Zeile aus:

```
{{befehl}} | cut --{{characters|fields}} {{1|1,10|1-10|  
1-|-10}}
```

- Schneide einen bestimmten Bereich von Feldern jeder Zeile mit einem bestimmten Trennzeichen aus:

```
{{befehl}} | cut --delimiter "{{{,}}}" --fields {{1}}
```

- Schneide einen bestimmten Bereich von Zeichen jeder Zeile einer bestimmten Datei aus:

```
cut --characters {{1}} {{pfad/zu/datei}}
```



# dart

Das Werkzeug zur Verwaltung von Dart-Projekten.

Weitere Informationen: <https://dart.dev/tools/dart-tool>.

- Initialisiere ein neues Dart-Projekt in einem Verzeichnis mit demselben Namen:

```
dart create {{projekt_name}}
```

- Ausführen einer Dart-Datei:

```
dart run {{pfad/zur/datei.dart}}
```

- Herunterladen der Abhängigkeiten für das aktuelle Projekt:

```
dart pub get
```

- Ausführen von Unit-Tests für das aktuelle Projekt:

```
dart test
```

- Aktualisieren veralteter Projektabhängigkeiten, um Null-Sicherheit zu unterstützen:

```
dart pub upgrade --null-safety
```

- Kompilieren einer Dart-Datei in eine native Binärdatei:

```
dart compile exe {{pfad/zur/datei.dart}}
```

# date

Setze die Systemzeit oder zeige sie an.

Weitere Informationen: <https://www.gnu.org/software/coreutils/date>.

- Zeige das aktuelle Datum im Format der eingestellten Locale an:

```
date +%c
```

- Zeige das aktuelle Datum in koordinierter Weltzeit (UTC) im ISO 8601-Format an:

```
date -u +%Y-%m-%dT%H:%M:%SZ
```

- Zeige das aktuelle Datum in Unixzeit (vergangene Sekunden seit der Unix-Epoche) an:

```
date +%s
```

- Konvertiere ein in Unixzeit gegebenes Datum zum Standardformat:

```
date -d @{{1473305798}}
```

- Konvertiere ein gegebenes Datum zu Unixzeit:

```
date -d "{{2018-09-01 00:00}}" +%s --utc
```

- Zeige das aktuelle Datum im RFC-3339 Format (YYYY-MM-DD hh:mm:ss TZ) an:

```
date --rfc-3339 s
```

- Setze das aktuelle Datum im Format MMDDhhmmYYYY.ss (YYYY und .ss sind optional):

```
date {{093023592021.59}}
```

- Zeige die aktuelle ISO-Wochenzahl an:

```
date +%V
```

# dd

Konvertiere und kopiere eine Datei.

Weitere Informationen: <https://manned.org/dd.1p>.

- Erstelle ein bootbares USB-Laufwerk von einer isohybriden Datei (wie `archlinux-xxxx.iso`) und zeige den Fortschritt an:

```
dd if={{pfad/zu/datei.iso}} of={{/dev/laufwerk}}  
status=progress
```

- Klone ein USB-Laufwerk in ein anderes in 4MiB Blöcken, ignoriere Fehler und zeige den Fortschritt an:

```
dd if={{/dev/quell_laufwerk}} of={{/dev/ziel_laufwerk}} bs=4M  
conv=noerror status=progress
```

- Erstelle eine Datei mit spezifizierte zufälligen Bytes mithilfe des Zufall-Treibers des Kernels:

```
dd bs={{100}} count={{1}} if=/dev/urandom of={{pfad/zu/  
datei}}
```

- Teste die Schreibgeschwindigkeit eines Laufwerks:

```
dd bs={{1024}} count={{1000000}} if=/dev/zero of={{pfad/zu/  
1GB_datei}}
```

- Erstelle ein System-Backup als IMG Datei und zeige den Fortschritt an:

```
dd if={{/dev/laufwerk}} of={{pfad/zu/datei.img}}  
status=progress
```

- Stelle ein Laufwerk aus einer IMG Datei wieder her und zeige den Fortschritt an:

```
dd if={{pfad/zu/datei.img}} of={{/dev/laufwerk}}  
status=progress
```

# deno

Eine sichere JavaScript- und TypeScript-Laufzeit.

Weitere Informationen: <https://deno.land>.

- Führe eine JavaScript oder TypeScript Datei aus:

```
deno run {{pfad/zu/datei.ts}}
```

- Starte ein REPL (interaktive Konsole):

```
deno
```

- Führe eine Datei mit Netzwerkzugriff aus:

```
deno run --allow-net {{pfad/zu/datei.ts}}
```

- Führe eine Datei von einer URL aus:

```
deno run {{https://deno.land/std/examples/welcome.ts}}
```

- Installiere ein ausführbares Skript von einer URL:

```
deno install {{https://deno.land/std/examples/colors.ts}}
```

# df

Verschafft einen Überblick über verfügbaren Speicherplatz im Dateisystem.

Weitere Informationen: <https://manned.org/df.1posix>.

- Zeige verfügbaren Platz auf allen eingehängten Dateisystemen:

```
df
```

- Zeige verfügbaren Platz auf allen eingehängten Dateisystemen in einem menschenlesbaren Format:

```
df -h
```

- Zeige das Dateisystem und dessen Speicherverbrauch, das die angegebene Datei oder Verzeichnis enthält:

```
df {{pfad/zu/datei_oder_verzeichnis}}
```

- Zeige Statistiken über die Anzahl freier Inodes:

```
df -i
```

- Zeige alle Dateisysteme bis auf die eines bestimmten Typs:

```
df -x {{squashfs}} -x {{tmpfs}}
```

# diff

Vergleiche Dateien und Verzeichnisse.

Weitere Informationen: <https://manned.org/diff>.

- Vergleiche Dateien (Listet jene Veränderungen auf, mit denen `datei1` zu `datei2` wird):

```
diff {{pfad/zu/datei1}} {{pfad/zu/datei2}}
```

- Vergleiche Dateien und ignoriere Leerzeichen:

```
diff --ignore-all-space {{pfad/zu/datei1}} {{pfad/zu/datei2}}
```

- Vergleiche Dateien und zeige Unterschiede nebeneinander:

```
diff --side-by-side {{pfad/zu/datei1}} {{pfad/zu/datei2}}
```

- Vergleiche Dateien und zeige Unterschiede in vereinheitlichtem Format (wie in `git diff`):

```
diff --unified {{pfad/zu/datei1}} {{pfad/zu/datei2}}
```

- Vergleiche Verzeichnisse rekursiv (zeigt sowohl Namen von unterschiedlichen Dateien/Verzeichnissen, als auch Unterschiede zwischen Dateien):

```
diff --recursive {{altes_verzeichnis}} {{neues_verzeichnis}}
```

- Vergleiche Verzeichnisse und zeige nur die Namen der Dateien, die unterschiedlich sind:

```
diff --recursive --brief {{altes_verzeichnis}}  
{{neues_verzeichnis}}
```

- Erstelle ein patch-Datei für Git bestehend aus den Unterschieden zweier Dateien und behandle fehlende Dateien als leer:

```
diff --text --unified --new-file {{pfad/zu/datei1}} {{pfad/  
zu/datei2}} > {{pfad/zu/diff.patch}}
```

# dirs

Zuletzt besuchte Ordner anzeigen und verändern.

Die Liste der zuletzt besuchten Ordner kann mit **pushd** und **popd** verändert werden.

Weitere Informationen: <https://www.gnu.org/software/bash/manual/bash.html#Directory-Stack-Builtins>.

- Zeige die zuletzt besuchten Ordner durch Leerzeichen getrennt an:

```
dirs
```

- Zeige die zuletzt besuchten Ordner mit einem Eintrag pro Zeile an:

```
dirs -p
```

- Zeige den N-ten Eintrag der zuletzt besuchten Ordner an, beginnend mit 0:

```
dirs +{{N}}
```

- Leere die Liste der zuletzt besuchten Ordner:

```
dirs -c
```

# docker build

Baut ein Image aus einem Dockerfile.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/build/>.

- Baue ein Docker Image aus dem Dockerfile im aktuellen Verzeichnis:  
`docker build .`
- Baue ein Docker Image aus einem Dockerfile an einer angegebenen URL:  
`docker build {{github.com/creack/docker-firefox}}`
- Baue ein Docker Image und gib ihm einen Tag:  
`docker build --tag {{name:tag}} .`
- Baue ein Docker Image ohne Build-Kontext:  
`docker build --tag {{name:tag}} - < {{Dockerfile}}`
- Verwende keinen Cache beim Bauen des Docker Images:  
`docker build --no-cache --tag {{name:tag}} .`
- Baue ein Docker Image mit einem spezifischen Dockerfile:  
`docker build --file {{Dockerfile}} .`
- Baue mit benutzerdefinierten Variablen, die während des Bauens zur Verfügung stehen:  
`docker build --build-arg {{HTTP_PROXY=http://10.20.30.2:1234}} --build-arg {{FTP_PROXY=http://40.50.60.5:4567}} .`



# docker compose

Starte und verwalte Anwendungen, welche aus mehreren Docker Containern bestehen.

Weitere Informationen: <https://docs.docker.com/compose/reference/>.

- Liste alle laufenden Container auf:

```
docker compose ps
```

- Erzeuge und starte alle Container im Hintergrund unter der Verwendung der Datei `docker-compose.yml` im aktuellen Verzeichnis:

```
docker compose up --detach
```

- Starte alle Container. Erzeuge zugehörige Docker Images bei Bedarf neu:

```
docker compose up --build
```

- Starte alle Container durch Angabe eines Projektnamens unter Verwendung einer alternativen Compose-Datei:

```
docker compose -p {{Projektname}} --file {{pfad/zu/verzeichnis}} up
```

- Stoppe alle laufenden Container:

```
docker compose stop
```

- Stoppe und entferne alle Container inklusive zugehöriger Netzwerke, Volumes und Images:

```
docker compose down --rmi all --volumes
```

- Zeige die Logs aller Container kontinuierlich an:

```
docker compose logs --follow
```

- Zeige die Logs eines spezifischen Containers kontinuierlich an:

```
docker compose logs --follow {{container_name}}
```

# docker container

Verwalte Docker Container.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/container/>.

- Liste zur Zeit laufende Container auf:

```
docker container ls
```

- Starte einen oder mehrere gestoppte Container:

```
docker container start {{container1_name}}  
{{container2_name}}
```

- Beende einen oder mehrere laufende Container sofort:

```
docker container kill {{container_name}}
```

- Stoppe einen oder mehrere laufende Container:

```
docker container stop {{container_name}}
```

- Pausiere alle Prozesse in einem oder mehreren Containern:

```
docker container pause {{container_name}}
```

- Zeige detaillierte Informationen zu einem oder mehreren Containern an:

```
docker container inspect {{container_name}}
```

- Exportiere das Dateisystem eines Containers als tar Archiv:

```
docker container export {{container_name}}
```

- Erstelle ein neues Image aus den Änderungen eines Containers:

```
docker container commit {{container_name}}
```

# docker cp

Kopiere Dateien oder Verzeichnisse zwischen Host- und Container-Dateisystem.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/cp>.

- Kopiere eine Datei oder ein Verzeichnis vom Host in einen Container:

```
docker cp {{pfad/zu/datei_oder_verzeichnis_auf_host}}  
{{container_name}}:{{pfad/zu/  
datei_oder_verzeichnis_in_container}}
```

- Kopiere eine Datei oder ein Verzeichnis von einem Container zum Host:

```
docker cp {{container_name}}:{{pfad/zu/  
datei_oder_verzeichnis_in_container}} {{pfad/zu/  
datei_oder_verzeichnis_auf_host}}
```

- Kopiere eine Datei oder ein Verzeichnis vom Host in einen Container und folge dabei Symlinks (kopiert die verlinkten Dateien statt der Symlinks):

```
docker cp --follow-link {{pfad/zu/symlink_auf_host}}  
{{container_name}}:{{pfad/zu/  
datei_oder_verzeichnis_in_container}}
```

# docker exec

Führe Befehle in einem bereits laufenden Docker Container aus.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/exec/>.

- Öffne eine Shell innerhalb eines bereits laufenden Containers:

```
docker exec --interactive --tty {{container_name}} {{/bin/bash}}
```

- Führe einen Befehl im Hintergrund (losgelöst) in einem laufenden Container aus:

```
docker exec --detach {{container_name}} {{befehl}}
```

- Bestimme das Arbeitsverzeichnis, in dem der Befehl ausgeführt werden soll:

```
docker exec --interactive --tty --workdir {{pfad/zum/verzeichnis}} {{container_name}} {{befehl}}
```

- Führe einen Befehl im Hintergrund in einem laufenden Container aus, aber lies von der Standardeingabe:

```
docker exec --interactive --detach {{container_name}} {{befehl}}
```

- Setze eine Umgebungsvariable in einer laufenden Bash Sitzung:

```
docker exec --interactive --tty --env {{variablen_name}}={{wert}} {{container_name}} {{/bin/bash}}
```

- Führe einen Befehl als ein bestimmter Benutzer aus:

```
docker exec --user {{benutzer}} {{container_name}} {{befehl}}
```

# docker image

Verwalte Docker Images.

Siehe auch **docker build**, **docker import** und **docker pull**.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/image/>.

- Liste lokale Docker Images auf:

```
docker image ls
```

- Lösche nicht verwendete, lokale Docker Images:

```
docker image prune
```

- Lösche alle nicht verwendeten Docker Images (nicht nur die ohne Tag):

```
docker image prune --all
```

- Zeige die Geschichte eines lokalen Docker Images:

```
docker image history {{image}}
```

# docker images

Verwalte Docker Images.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/images/>.

- Liste alle Docker Images auf:

```
docker images
```

- Liste alle Docker Images inkl. Zwischen-Images auf:

```
docker images --all
```

- Liste nur die IDs der Docker Images auf:

```
docker images --quiet
```

- Liste alle Docker Images auf, die nicht von einem Container verwendet werden:

```
docker images --filter dangling=true
```

- Liste alle Docker Images mit einer bestimmten Zeichenfolge im Namen auf:

```
docker images "{{*name*}}"
```

# docker inspect

Erhalte tiefgehende Informationen zu Docker Objekten.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/inspect/>.

- Zeige Hilfeseite:

```
docker inspect
```

- Zeige Informationen über einen Container, ein Image oder Volume anhand des Namens oder der ID:

```
docker inspect {{container|image|ID}}
```

- Zeige die IP Adresse eines Containers an:

```
docker inspect --format '{{range .NetworkSettings.Networks}}\n{{.IPAddress}}\n{{end}}' {{container}}
```

- Zeige den Pfad zur Logdatei eines Containers:

```
docker inspect --format '{{.LogPath}}' {{container}}
```

- Zeige den Namen des Images eines Containers:

```
docker inspect --format '{{.Config.Image}}' {{container}}
```

- Zeige die Konfiguration als JSON an:

```
docker inspect --format '{{json .Config}}' {{container}}
```

- Zeige alle Port Bindings:

```
docker inspect --format '{{range $p, $conf\n:= .NetworkSettings.Ports}}\n{{.HostPort}}\n{{end}}' {{container}}
```

# docker login

Bei einer Docker Registry einloggen.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/login/>.

- Interaktives Einloggen bei einer Registry:

```
docker login
```

- Einloggen mit einem angegebenen Benutzernamen (fragt nach dem Passwort):

```
docker login --username {{benutzername}}
```

- Einloggen mit einem angegebenen Benutzernamen und Passwort:

```
docker login --username {{benutzername}} --password  
{{password}} {{server}}
```

- Einloggen mit einem Passwort, welches von `stdin` gelesen wird:

```
echo "{{password}}" | docker login --username  
{{benutzername}} --password-stdin
```



# docker logs

Zeige Container Logs.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/logs>.

- Zeige die Logs eines Containers:

```
docker logs {{container_name}}
```

- Zeige die Logs und aktualisiere sie automatisch:

```
docker logs -f {{container_name}}
```

- Zeige die letzten 5 Zeilen:

```
docker logs {{container_name}} --tail {{5}}
```

- Zeige die Logs und füge ihnen Zeitstempel hinzu:

```
docker logs -t {{container_name}}
```

- Zeige Logs vor einem bestimmten Zeitpunkt der Ausführung des Containers (bspw. 23m, 10s, 2013-01-02T13:23:37):

```
docker logs {{container_name}} --until {{time}}
```

# docker-machine

Erstelle und verwalte Maschinen, die Docker ausführen.

Weitere Informationen: <https://github.com/docker/machine>.

- Liste zur Zeit laufende Docker Maschinen auf:

```
docker-machine ls
```

- Erzeuge eine neue Docker Maschine mit einem spezifischen Namen:

```
docker-machine create {{name}}
```

- Zeige den Status einer Maschine:

```
docker-machine status {{name}}
```

- Starte eine Maschine:

```
docker-machine start {{name}}
```

- Stoppe eine Maschine:

```
docker-machine stop {{name}}
```

- Zeige Informationen über eine Maschine:

```
docker-machine inspect {{name}}
```

# docker network

Erzeuge und verwalte Docker Netzwerke.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/network/>.

- Liste alle verfügbaren und konfigurierten Docker Netzwerke auf:

```
docker network ls
```

- Erzeuge ein benutzerdefiniertes Netzwerk:

```
docker network create --driver {{treiber_name}}  
{{netzwerk_name}}
```

- Zeige detaillierte Informationen der mit Leerzeichen separierten Netzwerke an:

```
docker network inspect {{netzwerk_name}}
```

- Verbinde einen Container mit einem Netzwerk anhand des Namens oder der ID:

```
docker network connect {{netzwerk_name}} {{container_name|  
ID}}
```

- Trenne einen Container von einem Netzwerk:

```
docker network disconnect {{netzwerk_name}} {{container_name|  
ID}}
```

- Entferne alle unbenutzten (nicht von Containern referenzierten) Netzwerke:

```
docker network prune
```

- Entferne mehrere - durch Leerzeichen getrennte - Netzwerke:

```
docker network rm {{netzwerk_name}}
```

# docker ps

Liste Docker Container.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/ps/>.

- Liste zur Zeit laufende Container auf:

```
docker ps
```

- Liste laufende und gestoppte Container auf:

```
docker ps --all
```

- Zeige den zuletzt erstellten Container (berücksichtigt jeden Status):

```
docker ps --latest
```

- Zeige nur Container mit einer bestimmten Zeichenkette im Namen:

```
docker ps --filter "name={{name}}"
```

- Zeige nur Container die von einem bestimmten Image abstammen:

```
docker ps --filter "ancestor={{image}}:{{tag}}"
```

- Zeige nur Container mit einem bestimmten Exit-Code:

```
docker ps --all --filter "exited={{code}}"
```

- Zeige nur Container mit einem bestimmten Status (created, running, removing, paused, exited und dead):

```
docker ps --filter "status={{status}}"
```

- Zeige nur Container, welche einen bestimmten Datenträger oder einen Datenträger an einem bestimmten Pfad eingehängt haben:

```
docker ps --filter "volume={{pfad/zum/verzeichnis}}" --format "table {{.ID}}\t{{.Image}}\t{{.Names}}\t{{.Mounts}}"
```

# docker rmi

Lösche eines oder mehrere Docker Images.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/rmi/>.

- Zeige Hilfe:

```
docker rmi
```

- Lösche eines oder mehrere Docker Images anhand der angegebenen Namen:

```
docker rmi {{image1 image2 ...}}
```

- Erzwingen das Löschen eines Images:

```
docker rmi --force {{image}}
```

- Lösche ein Image aber behalte Eltern-Images ohne Tag:

```
docker rmi --no-prune {{image}}
```

# docker run

Führe einen Befehl in einem neuen Docker Container aus.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/run/>.

- Führe einen Befehl in einem neuen Docker Container aus und verwende dabei einen bestimmten Tag eines Docker Images:

```
docker run {{image:tag}} {{befehl}}
```

- Führe einen Befehl in einem neuen Container im Hintergrund aus und zeige die ID:

```
docker run --detach {{image}} {{befehl}}
```

- Führe einen Befehl in einem kurzlebigen Container im interaktiven Modus mit einem Pseudo-TTY aus:

```
docker run --rm --interactive --tty {{image}} {{befehl}}
```

- Führe einen Befehl in einem neuen Container mit übergebenen Umgebungsvariablen aus:

```
docker run --env '{{variable}}={{wert}}' --env {{variable}}  
{{image}} {{befehl}}
```

- Führe einen Befehl in einem neuen Container mit eingebundenen Datenträgern aus:

```
docker run --volume {{pfad/zu/host_verzeichnis}}:{{pfad/zu/  
container_verzeichnis}} {{image}} {{befehl}}
```

- Führe einen Befehl in einem neuen Container mit veröffentlichten Ports aus:

```
docker run --publish {{host_port}}:{{container_port}}  
{{image}} {{befehl}}
```

- Führe einen Befehl in einem neuen Container aus und überschreibe den Einstiegspunkt des Images:

```
docker run --entrypoint {{befehl}} {{image}}
```

- Führe einen Befehl in einem neuen Container aus und verbinde ihn mit einem Netzwerk:

```
docker run --network {{netzwerk}} {{image}}
```

# docker save

Exportiere eines oder mehrere Docker Images in ein Archiv.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/save/>.

- Speichere ein Image über die Standardausgabe in ein Tar-Archiv:

```
docker save {{image}}:{{tag}} > {{pfad/zur/datei.tar}}
```

- Speichere ein Image in ein Tar-Archiv:

```
docker save --output {{pfad/zur/datei.tar}} {{image}}:{{tag}}
```

- Speichere alle Tags eines Images:

```
docker save --output {{pfad/zur/datei.tar}} {{image_name}}
```

- Speichere nur bestimmte Tags eines Images:

```
docker save --output {{pfad/zur/datei.tar}} {{image_name:tag1  
image_name:tag2 ...}}
```

# docker secret

Verwalte Secrets für Docker Swarm.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/secret/>.

- Erstelle ein neues Secret über die Standardeingabe:

```
{{befehl}} | docker secret create {{secret_name}} -
```

- Erstelle ein neues Secret aus einer Datei:

```
docker secret create {{secret_name}} {{pfad/zur/datei}}
```

- Liste alle Secrets auf:

```
docker secret ls
```

- Zeige detaillierte Informationen zu einem oder mehreren Secrets in einem menschenlesbaren Format:

```
docker secret inspect --pretty {{secret_name1 secret_name2 ...}}
```

- Lösche eines oder mehrere Secrets:

```
docker secret rm {{secret_name1 secret_name2 ...}}
```



# docker service

Verwalte Docker Services.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/service/>.

- Liste alle Services auf:

```
docker service ls
```

- Erstelle einen neuen Service:

```
docker service create --name {{service_name}} {{image}}:{{tag}}
```

- Zeige detaillierte Informationen der mit Leerzeichen separierten Services an:

```
docker service inspect {{service_name|ID}}
```

- Liste die Tasks der mit Leerzeichen separierten Services auf:

```
docker service ps {{service_name|ID}}
```

- Skaliere die angegebenen Services auf eine bestimmte Anzahl an Replikaten:

```
docker service scale {{service_name}}  
={{anzahl_an_replikaten}}
```

- Lösche die mit Leerzeichen separierten Services:

```
docker service rm {{service_name|ID}}
```

# docker-slim

Analysiere und optimiere Docker Images.

Weitere Informationen: <https://github.com/docker-slim/docker-slim>.

- Starte DockerSlim im interaktiven Modus:

```
docker-slim
```

- Analysiere die Docker Layer eines bestimmten Images:

```
docker-slim xray --target {{image:tag}}
```

- Linte ein Dockerfile:

```
docker-slim lint --target {{pfad/zum/Dockerfile}}
```

- Analysiere und generiere ein optimiertes Docker Image:

```
docker-slim build {{image:tag}}
```

- Zeige Hilfe für einen Unterbefehl:

```
docker-slim {{unterbefehl}} --help
```

# docker start

Starte einen oder mehrere gestoppte Container.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/start/>.

- Zeige Hilfe:

```
docker start
```

- Starte einen Docker Container:

```
docker start {{container}}
```

- Starte einen Container und verbinde dich mit der Standardausgabe sowie der Standardfehlerausgabe und leite Signale weiter:

```
docker start --attach {{container}}
```

- Starte einen oder mehrere durch Leerzeichen getrennte Container:

```
docker start {{container1 container2 ...}}
```

# docker stats

Zeige den Ressourcen-Verbrauch von Containern in Echtzeit.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/stats/>.

- Zeige sich stetig aktualisierende Statistiken von allen laufenden Containern:

```
docker stats
```

- Zeige sich stetig aktualisierende Statistiken der durch Leerzeichen getrennten Container:

```
docker stats {{container_name}}
```

- Ändere das Spaltenformat um die CPU Auslastung von Containern in Prozent anzuzeigen:

```
docker stats --format "{{.Name}}:\t{{.CPUPerc}}"
```

- Zeige Statistiken für alle Container (laufende und gestoppte):

```
docker stats --all
```

- Deaktiviere die laufende Aktualisierung und zeige nur die aktuellen Statistiken:

```
docker stats --no-stream
```

# docker swarm

Ein Container-Orchestrierungswerkzeug.

Weitere Informationen: <https://docs.docker.com/engine/swarm/>.

- Initialisiere ein Swarm Cluster:

```
docker swarm init
```

- Zeige den Token um als Manager oder Worker beizutreten:

```
docker swarm join-token {{worker|manager}}
```

- Füge dem Cluster eine neue Node hinzu:

```
docker swarm join --token {{token}} {{manager_node_url:2377}}
```

- Entferne einen Worker vom Swarm (führe dies auf der Worker Node aus):

```
docker swarm leave
```

- Zeige die aktuellen CA Zertifikate im PEM Format:

```
docker swarm ca
```

- Rotiere die aktuellen CA Zertifikate und zeige die neuen Zertifikate:

```
docker swarm ca --rotate
```

- Ändere die Gültigkeitsdauer für Node-Zertifikate:

```
docker swarm update --cert-expiry {{stunden}}h{{minuten}}m{{sekunden}}s
```

# docker system

Verwalte Docker Daten und zeige systemweite Informationen an.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/system/>.

- Zeige Hilfe:

```
docker system
```

- Zeige Docker Festplattennutzung:

```
docker system df
```

- Zeige detaillierte Informationen zur Festplattennutzung:

```
docker system df --verbose
```

- Entferne nicht-verwendete Daten:

```
docker system prune
```

- Entferne nicht-verwendete Daten, die älter als die angegebene Zeit sind:

```
docker system prune --filter "until={{stunden}}h{{minuten}}m"
```

- Zeige Echtzeit-Events vom Docker Daemon:

```
docker system events
```

- Zeige Echtzeit-Events von Containern und formatiere jede Zeile als gültiges JSON:

```
docker system events --filter 'type=container' --format '{{json .}}'
```

- Zeige systemweite Informationen:

```
docker system info
```

# docker

Verwalte Docker Container und Images.

Manche Unterbefehle wie **docker run** sind separat dokumentiert.

Weitere Informationen: <https://docs.docker.com/engine/reference/commandline/cli/>.

- Liste laufende und gestoppte Container auf:

```
docker ps --all
```

- Erzeuge einen Container aus einem Image und benenne ihn:

```
docker run --name {{container_name}} {{image}}
```

- Stoppe oder starte einen existierenden Container:

```
docker {{start|stop}} {{container_name}}
```

- Lade ein Image herunter:

```
docker pull {{image}}
```

- Zeige eine Liste der bereits heruntergeladenen Images an:

```
docker images
```

- Öffne eine Konsole innerhalb eines bereits laufenden Containers:

```
docker exec -it {{container_name}} {{sh}}
```

- Lösche einen gestoppten Container:

```
docker rm {{container_name}}
```

- Zeige die Logs eines Containers an und aktualisiere sie automatisch:

```
docker logs -f {{container_name}}
```

# dotnet

Plattformübergreifende Kommandozeilenanwendungen für .NET Core.

Manche Unterbefehle wie **dotnet build** sind separat dokumentiert.

Weitere Informationen: <https://learn.microsoft.com/dotnet/core/tools>.

- Initialisiere ein neues .NET Projekt:

```
dotnet new {{vorlagen_name}}
```

- Stelle nuget-Pakete wieder her:

```
dotnet restore
```

- Baue das .NET Projekt im aktuellen Ordner und führe es aus:

```
dotnet run
```

- Führe eine gebaute dotnet-Applikation aus (Benötigt nur die Laufzeitumgebung. Die anderen Befehle benötigen auch den Entwicklungsteil):

```
dotnet {{pfad/zu/anwendung.dll}}
```



# du

Disk usage: Plattenplatzverbrauch von Dateien und Verzeichnissen ermitteln.

Weitere Informationen: <https://www.gnu.org/software/coreutils/du>.

- Liste die Größe von Verzeichnissen und Unterverzeichnissen in den gegebenen Einheiten (B/KiB/MiB) auf:

```
du -{{b|k|m}} {{pfad/zu/verzeichnis}}
```

- Liste die Größe von Verzeichnissen und Unterverzeichnissen in menschenlesbaren Einheiten auf (d.h. automatische Auswahl der geeigneten Einheit für jede Größe):

```
du -h {{pfad/zu/verzeichnis}}
```

- Zeige die Größe eines einzelnen Verzeichnisses in menschenlesbaren Einheiten:

```
du -sh {{pfad/zu/verzeichnis}}
```

- Liste die Größe von Verzeichnissen und Unterverzeichnissen und aller ihrer Dateien in menschenlesbaren Einheiten auf:

```
du -ah {{pfad/zu/verzeichnis}}
```

- Liste die menschenlesbaren Größen eines Verzeichnisses und aller Unterverzeichnisse, bis zu einer Tiefe von N Ebenen:

```
du -h --max-depth=N {{pfad/zu/verzeichnis}}
```

- Liste die menschenlesbare Größe aller .jpg-Dateien in Unterverzeichnissen des aktuellen Verzeichnisses auf und zeige am Ende die kumulierte Gesamtsumme an:

```
du -ch {{*/*.jpg}}
```

# duf

Festplattennutzung/freie Verwendbarkeit.

Weitere Informationen: <https://github.com/muesli/duf>.

- Liste zugängliche Geräte auf:

```
duf
```

- Liste alles auf (auch pseudo-, doppelte oder unzugängliche Dateisysteme):

```
duf --all
```

- Zeige nur konkrete Geräte oder Mountpoints:

```
duf {{pfad/zu/verzeichnis1 pfad/zu/verzeichnis2 ...}}
```

- Sortiere die Ergebnisse nach einem spezifischen Kriterium:

```
duf --sort {{size|used|avail|usage}}
```

# dust

Dust gib einen sofortigen Überblick, welche Verzeichnisse Festplatten Speicherplatz benutzen.

Weitere Informationen: <https://github.com/bootandy/dust>.

- Informationen für das aktuelle Verzeichnis anzeigen:

```
dust
```

- Informationen für eine durch Leerzeichen getrennte Liste von Verzeichnissen anzeigen:

```
dust {{pfad/zum/verzeichnis1}} {{pfad/zum/verzeichnis2}}
```

- Zeige 30 Verzeichnisse an (Standardwert: 21):

```
dust --number-of-lines {{30}}
```

- Zeigt Informationen für das aktuelle Verzeichnis an, bis zu 3 Ebenen tief:

```
dust --depth {{3}}
```

- Die größten Verzeichnisse in absteigender Reihenfolge oben anzeigen:

```
dust --reverse
```

- Alle Dateien und Verzeichnisse mit einem bestimmten Namen ignorieren:

```
dust --ignore-directory {{datei_oder_verzeichnis_name}}
```

- Keine Prozentbalken und Prozente anzeigen:

```
dust --no-percent-bars
```

# ebook-convert

Kann verwendet werden, um E-Books zu gängigen Dateiformaten umzuwandeln, z.B. PDF, EPUB und MOBI.

Teil der Calibre e-book library.

Weitere Informationen: <https://manual.calibre-ebook.com/generated/en/ebook-convert.html>.

- Konvertiere ein E-Book in ein anderes Format:

```
ebook-convert {{pfad/zu/input_datei}} {{pfad/zu/output_datei}}
```

- Konvertiere eine Markdown- oder HTML Datei zu einem E-Book inklusive Inhaltsverzeichnis, Titel und Autoren:

```
ebook-convert {{pfad/zu/input_datei}} {{pfad/zu/output_datei}} --level1-toc="//h:h1" --level2-toc="//h:h2" --level3-toc="//h:h3" --title={{titel}} --authors={{autor}}
```

# echo

Gib angegebene Argumente aus.

Weitere Informationen: <https://www.gnu.org/software/coreutils/echo>.

- Gib einen Text aus. (Hinweis: Anführungszeichen sind optional):

```
echo "{{Hallo Welt}}"
```

- Gib einen Text mit Umgebungsvariablen aus:

```
echo "{{Liste aller Systempfade: $PATH}}"
```

- Gib einen Text ohne darauffolgenden Zeilenumbruch aus:

```
echo -n "{{Hallo Welt}}"
```

- Füge einen Text in eine Datei ein:

```
echo "{{Hallo Welt}}" >> {{datei.txt}}
```

- Ermögliche Interpretation von Fluchtsymbolen (backslash escape):

```
echo -e "{{Spalte 1\tSpalte 2}}"
```

# emacs

Der erweiterbare, veränderbare und selbst-dokumentierende Echtzeit Text Editor.

Siehe auch **emacsclient**.

Weitere Informationen: <https://www.gnu.org/software/emacs>.

- Öffne eine Datei in Emacs:

```
emacs {{pfad/zu/datei}}
```

- Öffne eine Datei in einer bestimmten Zeile:

```
emacs +{{zeilennummer}} {{pfad/zu/datei}}
```

- Starte Emacs in der Konsole (ohne X-Fenster):

```
emacs --no-window-system
```

- Starte einen Emacs-Server im Hintergrund (aufrufbar mit **emacsclient**):

```
emacs --daemon
```

- Beende einen laufenden Emacs-Server und alle Instanzen und frage nach Bestätigung für ungespeicherte Dateien:

```
emacsclient --eval '(save-buffers-kill-emacs)'
```

- Tastenkombination zum Speichern einer Datei:

```
<Ctrl> + X, <Ctrl> + S
```

- Tastenkombination zum Beenden von Emacs:

```
<Ctrl> + X, <Ctrl> + C
```

# emacsclient

Öffnet Dateien in einem laufenden Emacs-Server.

Siehe auch **emacs**.

Weitere Informationen: <https://www.emacswiki.org/emacs/EmacsClient>.

- Öffne eine Datei in einem laufenden Emacs-Server (mit GUI wenn möglich):

```
emacsclient {{pfad/zu/datei}}
```

- Öffne eine Datei in der Konsole (ohne X-Fenster):

```
emacsclient --no-window-system {{pfad/zu/datei}}
```

- Öffne eine Datei in einem neuen Emacs Fenster:

```
emacsclient --create-frame {{pfad/zu/datei}}
```

- Führe einen Befehl aus und schreibe das Ergebnis in stdout:

```
emacsclient --eval '({{befehl}})'
```

- Gib einen alternativen Editor an für den Fall, dass kein Emacs-Server läuft:

```
emacsclient --alternate-editor {{editor}} {{pfad/zu/datei}}
```

- Beende einen laufenden Emacs-Server und alle Instanzen und frage nach Bestätigung für ungespeicherte Dateien:

```
emacsclient --eval '(save-buffers-kill-emacs)'
```

# eslint

Ein erweiterbarer Linter für JavaScript und JSX.

Weitere Informationen: <https://eslint.org>.

- Erstelle eine ESLint-Konfigurationsdatei:

```
eslint --init
```

- Linte Dateien:

```
eslint {{pfad/zu/datei1.js pfad/zu/datei2.js ...}}
```

- Behebe Lintingfehler:

```
eslint --fix
```

- Linte mit einer Konfigurationsdatei:

```
eslint -c {{pfad/zu/konfigurationsdatei}} {{pfad/zu/datei1.js  
pfad/zu/datei2.js ...}}
```



# exa

Ein moderner Ersatz für `ls` (Verzeichnisinhalte auflisten).

Weitere Informationen: <https://the.exa.website>.

- Liste eine Datei pro Zeile auf:

```
exa --oneline
```

- Liste alle Dateien auf, einschließlich versteckter Dateien:

```
exa --all
```

- Liste alle Dateien im langen Format auf (Berechtigungen, Eigentümer, Größe und Änderungsdatum):

```
exa --long --all
```

- Liste Dateien nach Größe absteigend sortiert auf:

```
exa --reverse --sort={{size}}
```

- Zeige Dateien in einer Baumstruktur an, die drei Ebenen tief ist:

```
exa --long --tree --level={{3}}
```

- Liste Dateien nach Änderungsdatum aufsteigend sortiert auf:

```
exa --long --sort={{modified}}
```

- Liste Dateien inklusive Header, Icons und Git-Status:

```
exa --long --header --icons --git
```

- Liste keine Dateien auf, die in `.gitignore` erwähnt werden:

```
exa --git-ignore
```

# exit

Verlasse die Shell.

Weitere Informationen: <https://manned.org/exit.1posix>.

- Beende die Shell mit dem Exitcode des zuletzt ausgeführten Befehls:

```
exit
```

- Beende die Shell mit dem angegebenen Exitcode:

```
exit {{exitcode}}
```

# export

Befehl zum Markieren von Shell-Variablen in der aktuellen Umgebung, die mit allen neu abgezweigten Unterprozessen exportiert werden sollen.

Weitere Informationen: <https://manned.org/export.1posix>.

- Lege eine neue Umgebungsvariable fest:

```
export {{variable}}={{wert}}
```

- Entferne eine Umgebungsvariable:

```
export -n {{variable}}
```

- Markiere eine Shell-Funktion für den Export:

```
export -f {{funktionsname}}
```

- Hänge etwas an die PATH-Variable an:

```
export PATH=$PATH:{{pfad/zu/anhängen}}
```

# fastd

VPN daemon.

Arbeitet auf Schicht 2 oder Schicht 3, unterstützt verschiedene Verschlüsselungsmethoden, wird von Freifunk verwendet.

Weitere Informationen: <https://fastd.readthedocs.io/en/stable/>.

- Starte `fastd` mit einer bestimmten Konfigurationsdatei:

```
fastd --config {{pfad/zu/fastd.conf}}
```

- Starte einen Schicht-3-VPN mit einer MTU von 1400 und lade den Rest der Konfigurationsparameter aus einer Datei:

```
fastd --mode {{tap}} --mtu {{1400}} --config {{pfad/zu/fastd.conf}}
```

- Validiere eine Konfigurationsdatei:

```
fastd --verify-config --config {{pfad/zu/fastd.conf}}
```

- Generiere einen neuen Schlüssel:

```
fastd --generate-key
```

- Zeige den öffentlichen Schlüssel zu einem privaten Schlüssel in einer Konfigurationsdatei an:

```
fastd --show-key --config {{pfad/zu/fastd.conf}}
```

- Zeige die aktuelle Version an:

```
fastd -v
```

# fdroid

F-Droid Build Tool.

F-Droid ist ein installierbarer Katalog mit FOSS (Freie Open Source Software) Apps für Android.

Weitere Informationen: <https://f-droid.org/>.

- Kompiliere eine bestimmte App:

```
fdroid build {{app_id}}
```

- Kompiliere eine bestimmte App in einer Build-Server-VM:

```
fdroid build {{app_id}} --server
```

- Veröffentliche die App im lokalen Repository:

```
fdroid publish {{app_id}}
```

- Installiere die App auf jedem verbundenen Gerät:

```
fdroid install {{app_id}}
```

- Überprüfe, ob die Metadaten korrekt formatiert sind:

```
fdroid lint --format {{app_id}}
```

- Korrigiere die Formatierung automatisch (wenn möglich):

```
fdroid rewritemeta {{app_id}}
```

# fdroidcl

F-Droid CLI (Command-line Interface) Client.

Weitere Informationen: <https://github.com/mvdan/fdroidcl>.

- Aktualisiere den Index:

```
fdroidcl update
```

- Zeige Informationen über eine App:

```
fdroidcl show {{app_id}}
```

- Lade eine apk-Datei herunter:

```
fdroidcl download {{app_id}}
```

- Suche nach einer App im Index:

```
fdroidcl search {{suchmuster}}
```

- Installiere eine App auf einem verbundenen Gerät:

```
fdroidcl install {{app_id}}
```

# ffmpeg

Programm zum konvertieren von Videos.

Weitere Informationen: <https://ffmpeg.org>.

- Extrahiere den Ton eines Videos und speichere ihn als MP3:

```
ffmpeg -i {{pfad/zu/video.mp4}} -vn {{pfad/zu/audio}}.mp3
```

- Kombiniere nummerierte Bilder (frame\_1.jpg, frame\_2.jpg, etc) in ein Video oder Gif:

```
ffmpeg -i {{pfad/zu/frame_%d.jpg}} -f image2 {{video.mpg|video.gif}}
```

- Extrahiere einen einzelnen Frame von einem Video bei mm:ss and speichere als 128x128 Bild:

```
ffmpeg -ss {{mm:ss}} -i {{pfad/zu/video.mp4}} -frames 1 -s {{128x128}} -f image2 {{pfad/zu/bild.png}}
```

- Trimme ein Video von mm:ss bis mm2:ss2 (Ohne -to bis zum Ende des Videos):

```
ffmpeg -ss {{mm:ss}} -to {{mm2:ss2}} -i {{pfad/zu/video/video.mp4}} -codec copy {{pfad/zu/output.mp4}}
```

- Konvertiere ein AVI Video zu MP4. AAC Audio @ 128kbit, h264 Video @ CRF 23:

```
ffmpeg -i {{pfad/zu/input_video}}.avi -codec:a aac -b:a 128k -codec:v libx264 -crf 23 {{pfad/zu/output_video}}.mp4
```

- Remuxe ein MKV Video zu MP4 ohne Audio oder Video streams neu zu codieren:

```
ffmpeg -i {{pfad/zu/input_video}}.mkv -codec copy {{pfad/zu/output_video}}.mp4
```

- Konvertiere ein MP4 video zu VP9. Für beste Qualität, nutze einen CRF Wert von 15 bis 35 und -b:v MUSS 0 sein:

```
ffmpeg -i {{pfad/zu/input_video}}.mp4 -codec:v libvpx-vp9 -crf {{30}} -b:v 0 -codec:a libopus -vbr on -threads {{thread_anzahl}} {{pfad/zu/output_video}}.webm
```

# ffprobe

Multimedia Stream Analysierer.

Weitere Informationen: <https://ffmpeg.org/ffprobe.html>.

- Zeige alle verfügbaren Stream-Informationen einer Medien-Datei an:

```
ffprobe -v error -show_streams {{datei.mp4}}
```

- Zeige die Spieldauer an:

```
ffprobe -v error -show_entries format=duration -of  
default=noprint_wrappers=1:nokey=1 {{datei.mp4}}
```

- Zeige die Bildfrequenz eines Videos an:

```
ffprobe -v error -select_streams v:0 -show_entries  
stream=avg_frame_rate -of default=noprint_wrappers=1:nokey=1  
{{datei.mp4}}
```

- Zeige die Breite (width) oder Höhe (height) eines Videos an:

```
ffprobe -v error -select_streams v:0 -show_entries  
stream={{width|height}} -of  
default=noprint_wrappers=1:nokey=1 {{datei.mp4}}
```

- Zeige die durchschnittliche Bitrate eines Videos an:

```
ffprobe -v error -select_streams v:0 -show_entries  
stream=bit_rate -of default=noprint_wrappers=1:nokey=1  
{{datei.mp4}}
```



# ffsend

Teile Dateien einfach und sicher in der Command-line.

Weitere Informationen: <https://gitlab.com/timvissee/ffsend>.

- Lade eine Datei hoch:

```
ffsend upload {{datei}}
```

- Lade eine Datei herunter:

```
ffsend download {{url}}
```

- Lade eine Datei mit Passwort hoch:

```
ffsend upload {{datei}} -p {{passwort}}
```

- Lade eine passwortgeschützte Datei herunter:

```
ffsend download {{datei}} -p {{passwort}}
```

- Lade eine Datei hoch und erlaube maximal 4 Downloads:

```
ffsend upload {{datei}} -d {{4}}
```

# fg

Bringt Prozesse in den Vordergrund.

Weitere Informationen: <https://manned.org/fg>.

- Bringe zuletzt suspendierten Prozess in den Vordergrund:

```
fg
```

- Bringe einen bestimmten Prozess in den Vordergrund:

```
fg %{{prozess_id}}
```

# file

Bestimmen des Dateityps.

Weitere Informationen: <https://manned.org/file>.

- Gibt eine Beschreibung des Typs der angegebenen Datei zurück. Funktioniert bei Dateien ohne Dateiendung:

```
file {{dateiname}}
```

- Bestimmt die Dateityp(en) in einer gezippten Datei:

```
file -z {{archiv.zip}}
```

- Zulassen, dass die Datei mit speziellen oder Gerätedateien arbeitet:

```
file -s {{dateiname}}
```

- Hört nicht bei der ersten Dateityp-Übereinstimmung auf; weitermachen bis zum Ende der Datei:

```
file -k {{dateiname}}
```

- Bestimmen des MIME-Codierungstyp einer Datei:

```
file -i {{dateiname}}
```

# firefox

Ein gratis Open Source Internet Browser.

Weitere Informationen: [https://developer.mozilla.org/en-US/docs/Mozilla/Command Line Options](https://developer.mozilla.org/en-US/docs/Mozilla/Command_Line_Options).

- Starte Firefox und öffne eine Website:

```
firefox {{https://www.duckduckgo.com}}
```

- Öffne ein neues Fenster:

```
firefox --new-window {{https://www.duckduckgo.com}}
```

- Öffne ein privates (Inkognito) Fenster:

```
firefox --private-window
```

- Suche nach "wikipedia" mit der Standard-Suchmaschine:

```
firefox --search "{{wikipedia}}"
```

- Starte Firefox im sicheren Modus (alle Erweiterungen sind deaktiviert):

```
firefox --safe-mode
```

- Erstelle eine Bildschirmaufnahme einer Website, ohne die GUI zu starten:

```
firefox --headless --screenshot {{pfad/zu/ausgabedatei.png}}  
{{https://beispiel.de/}}
```

- Verwende ein bestimmtes Profil um mehrere einzelne Instanzen gleichzeitig laufen zu lassen:

```
firefox --profile {{pfad/zu/verzeichnis}} {{https://  
beispiel.de/}}
```

- Lege Firefox als Standard-Browser fest:

```
firefox --setDefaultBrowser
```

# fish

The Friendly Interactive SHell.

Eine benutzerfreundliche Eingabeaufforderung.

Weitere Informationen: <https://fishshell.com>.

- Starte eine interaktive Shell-Sitzung:

```
fish
```

- Starte eine interaktive Shell-Sitzung ohne die Start-Konfiguration zu laden:

```
fish --no-config
```

- Führe einen bestimmten Befehl aus:

```
fish --command "{{echo 'fish wird ausgeführt'}}"
```

- Führe ein bestimmtes Skript aus:

```
fish {{pfad/zu/skript.fish}}
```

- Überprüfe ein bestimmtes Skript auf Syntaxfehler:

```
fish --no-execute {{pfad/zu/skript.fish}}
```

- Starte eine private, interaktive Shell-Sitzung, in der fish weder auf die Shell-History zugreift, noch diese verändert:

```
fish --private
```

- Definiere und exportiere eine Umgebungsvariable, die über mehrere Shell-Neustarts hinweg existiert (builtin):

```
set --universal --export {{variablenname}} {{variablenwert}}
```

# flutter

Googles Cross-Platform Open-Source SDK.

Weitere Informationen: <https://github.com/flutter/flutter/wiki/The-flutter-tool>.

- Initialisiere ein neues Flutter-Projekt in einem gleichnamigen Verzeichnis:

```
flutter create {{Projektname}}
```

- Überprüfe, ob alle externen Tools korrekt installiert sind:

```
flutter doctor
```

- Zeige oder wechsle Flutter Kanäle:

```
flutter channel {{stable|beta|dev|master}}
```

- Starte Flutter auf allen gestarteten Emulatoren und verbundenen Geräten:

```
flutter run -d all
```

- Starte Tests in einem Projekt vom Wurzelverzeichnis aus:

```
flutter test {{test/beispiel_test.dart}}
```

- Baue eine Release APK für die meisten modernen Smartphones:

```
flutter build apk --target-platform {{android-arm}},  
{{android-arm64}}
```

- Zeige Hilfe für einen bestimmten Befehl:

```
flutter help {{befehl}}
```

# fortune

Gib ein zufälliges Glückskeks-Zitat aus.

Weitere Informationen: <https://manned.org/fortune>.

- Gib ein Zitat aus:

```
fortune
```

- Gib ein beleidigendes Zitat aus:

```
fortune -o
```

- Gib ein langes Zitat aus:

```
fortune -l
```

- Gib ein kurzes Zitat aus:

```
fortune -s
```

- Gib alle verfügbaren Zitat-Datenbank-Dateien aus:

```
fortune -f
```

- Gib ein Zitat von einer durch `fortune -f` aufgelisteten Datenbank-Dateien aus:

```
fortune {{dateiname}}
```

# fossil ci

Dieser Befehl ist ein Alias von **fossil commit**.

Weitere Informationen: <https://fossil-scm.org/home/help/commit>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr fossil-commit
```



# fossil delete

Dieser Befehl ist ein Alias von **fossil rm**.

Weitere Informationen: <https://fossil-scm.org/home/help/delete>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr fossil rm
```

# fossil forget

Dieser Befehl ist ein Alias von **fossil rm**.

Weitere Informationen: <https://fossil-scm.org/home/help/forget>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr fossil rm
```

# fossil new

Dieser Befehl ist ein Alias von **fossil init**.

Weitere Informationen: <https://fossil-scm.org/home/help/new>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr fossil-init
```

# fuck

Korrigiert den zuletzt ausgeführten Befehl.

Weitere Informationen: <https://github.com/nvbn/thefuck>.

- Lege den fuck alias für thefuck fest:

```
eval "$(thefuck --alias)"
```

- Versuche den zuletzt ausgeführten Befehl zu korrigieren:

```
fuck
```

- Bestätige das erste Ergebnis sofort (korrektes Argument hängt vom Level der Frustration ab):

```
fuck --{{yes|yeah|hard}}
```

# g++

Kompiliere C++ Quelldateien.

Teil der GCC (GNU Compiler Collection).

Weitere Informationen: <https://gcc.gnu.org>.

- Kompiliere eine Quelldatei in eine ausführbare Binärdatei:

```
g++ {{pfad/zu/quelldatei.cpp}} -o {{pfad/zu/binärdatei}}
```

- Zeige geläufige Fehler und Warnungen an:

```
g++ {{pfad/zu/quelldatei.cpp}} -Wall -o {{pfad/zu/binärdatei}}
```

- Wähle einen Sprachstandard (C++98/C++11/C++14/C++17) für das Kompilieren:

```
g++ {{pfad/zu/quelldatei.cpp}} -std={{C++98|C++11|C++14|C++17}} -o {{pfad/zu/binärdatei}}
```

- Binde Bibliotheken, die sich an einem anderen Pfad als die Quelldatei befinden, ein:

```
g++ {{pfad/zu/quelldatei.cpp}} -o {{pfad/zu/binärdatei}} -I{{pfad/zu/headerdatei}} -L{{pfad/zu/bibliothek}} -l{{bibliotheks_name}}
```

- Kompiliere und linke mehrere Quelldateien in eine ausführbare Binärdatei:

```
g++ -c {{pfad/zu/quelldatei_1.cpp pfad/zu/quelldatei_2.cpp ...}} && g++ -o {{pfad/zu/binärdatei}} {{pfad/zu/quelldatei_1.o pfad/zu/quelldatei_2.o ...}}
```

- Gib die Version von g++ aus:

```
g++ --version
```

# gcc

Präprozessiert und kompiliert C und C++ Quellcodedateien und linkt diese anschließend zusammen.

Weitere Informationen: <https://gcc.gnu.org>.

- Kompiliere mehrere Quellcodedateien zu einer ausführbaren Datei:

```
gcc {{pfad/zu/datei1.c}} {{pfad/zu/datei2.c}} --output  
{{pfad/zu/binärdatei}}
```

- Erlaube Warnungen und debug-Symbole in der Ausgabedatei:

```
gcc {{pfad/zu/datei.c}} -Wall -Og --output {{pfad/zu/  
binärdatei}}
```

- Inkludiere Bibliotheken aus anderen Verzeichnissen:

```
gcc {{pfad/zu/datei.c}} --output {{pfad/zu/binärdatei}} -  
I{{pfad/zu/headerdatei}} -L{{pfad/zu/bibliothek1}} -l{{pfad/  
zu/bibliothek2}}
```

- Kompiliere Quellcodedateien zu Assemblerinstruktionen:

```
gcc -S {{pfad/zu/datei.c}}
```

- Kompiliere eine oder mehrere Quellcodedateien ohne diese zu linken:

```
gcc -c {{pfad/zu/datei.c}}
```

# gcloud

Das offizielle CLI-Tool für die Google Cloud Platform.

Weitere Informationen: <https://cloud.google.com/sdk/gcloud>.

- Liste alle Eigenschaften der aktiven Konfiguration auf:

```
gcloud config list
```

- Mit einem Google-Konto anmelden:

```
gcloud auth login
```

- Lege das aktive Projekt fest:

```
gcloud config set project {{Projektname}}
```

- Stelle eine SSH-Verbindung zu einer virtuellen Maschineninstanz her:

```
gcloud compute ssh {{Benutzer}}@{{Instanz}}
```

- Zeige alle Google Compute Engine-Instanzen in einem Projekt an.  
Standardmäßig werden Instanzen aus allen Zonen aufgelistet:

```
gcloud compute instances list
```

- Aktualisiere eine kubeconfig-Datei mit den entsprechenden Anmeldeinformationen, um kubectl auf einen bestimmten Cluster in Google Kubernetes Engine auszurichten:

```
gcloud container clusters get-credentials {{Clustername}}
```

- Aktualisiere alle gcloud CLI-Komponenten:

```
gcloud components update
```

- Zeige Hilfe für einen bestimmten Befehl an:

```
gcloud help {{Befehl}}
```

# `gdb`

Der GNU Debugger.

Weitere Informationen: <https://www.gnu.org/software/gdb>.

- Debugge eine ausführbare Datei:

```
gdb {{ausführbare_datei}}
```

- Binde einen Prozess an gdb:

```
gdb -p {{prozess_ID}}
```

- Debugge mit einer Kerndatei:

```
gdb -c {{kerndatei}} {{ausführbare_datei}}
```

- Führe angegebene Befehle beim Start von gdb aus:

```
gdb -ex "{{befehle}}" {{ausführbare_datei}}
```

- Starte gdb und übergib Argumente an die ausführbare Datei:

```
gdb --args {{ausführbare_datei}} {{argument1}} {{argument2}}
```



# gh cs

Dieser Befehl ist ein Alias von **gh codespace**.

Weitere Informationen: [https://cli.github.com/manual/gh\\_codespace](https://cli.github.com/manual/gh_codespace).

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr gh-codespace
```

# gh

Arbeite mit GitHub von der Konsole aus.

Manche Unterbefehle wie **gh config** sind separat dokumentiert.

Weitere Informationen: <https://cli.github.com/>.

- Kclone ein GitHub Repository lokal:

```
gh repo clone {{besitzer}}/{{repository}}
```

- Erstelle ein neues Issue:

```
gh issue create
```

- Zeige und filter offene Issues des aktuellen Repositories:

```
gh issue list
```

- Öffne ein Issue im Standard-Webbrowser:

```
gh issue view --web {{issue_nummer}}
```

- Erstelle eine Pull Request:

```
gh pr create
```

- Öffne eine Pull Request im Standard-Webbrowser:

```
gh pr view --web {{pr_nummer}}
```

- Wechsle lokal zu einer bestimmten Pull Request:

```
gh pr checkout {{pr_nummer}}
```

- Zeige den Status der Pull Requests eines Repositories:

```
gh pr status
```

# git add

Füge Dateien zum Index/Stage hinzu.

Weitere Informationen: <https://git-scm.com/docs/git-add>.

- Füge eine bestimmte Datei zum Index hinzu:

```
git add {{pfad/zu/datei}}
```

- Füge alle Dateien zum Index hinzu (nachverfolgte und nicht nachverfolgte):

```
git add -A
```

- Füge nur nachverfolgte Dateien zum Index hinzu (Updaten des Index):

```
git add -u
```

- Füge auch Dateien, welche ignoriert werden (`.gitignore`) hinzu:

```
git add -f
```

- Füge Teile von Dateien zum Index interaktiv hinzu:

```
git add -p
```

- Füge Teile einer bestimmten Datei interaktiv hinzu:

```
git add -p {{pfad/zu/datei}}
```

- Füge Dateien zum Index interaktiv hinzu:

```
git add -i
```

# git am

Patch-Dateien integrieren. Nützlich beim Empfang von Commits per E-Mail.

Siehe auch **git format-patch** zur Erzeugung von Patch-Dateien.

Weitere Informationen: <https://git-scm.com/docs/git-am>.

- Integriere eine Patch-Datei:

```
git am {{pfad/zu/datei.patch}}
```

- Herunterladen und Integrieren einer Patch-Datei:

```
curl -L {{https://beispiel.de/datei.patch}} | git apply
```

- Brich das Integrieren einer Patch-Datei ab:

```
git am --abort
```

- Integriere so viele Patch-Dateien wie möglich und speichere fehlgeschlagene Teile:

```
git am --reject {{pfad/zu/datei.patch}}
```

# git apply

Integriere eine Patch-Datei und/oder füge sie zum Index hinzu.

Weitere Informationen: <https://git-scm.com/docs/git-apply>.

- Gib Informationen über gepatchte Dateien aus:

```
git apply --verbose {{pfad/zu/datei}}
```

- Integriere die Patch-Datei und füge sie zum Index hinzu:

```
git apply --index {{pfad/zu/datei}}
```

- Integriere eine externe Patch-Datei:

```
curl -L {{https://beispiel.de/datei.patch}} | git apply
```

- Gib diffstat des Inputs aus und integriere die Patch-Datei:

```
git apply --stat --apply {{pfad/zu/datei}}
```

- Integriere eine Patch-Datei in umgekehrter Reihenfolge:

```
git apply --reverse {{pfad/zu/datei}}
```

- Speichere das Ergebnis einer Patch-Datei im Index, ohne den Arbeitsbaum zu verändern:

```
git apply --cache {{pfad/zu/datei}}
```

# git archive

Erstelle ein Archiv von Dateien.

Weitere Informationen: <https://git-scm.com/docs/git-archive>.

- Erstelle ein tar-Archiv aus dem Inhalt des aktuellen HEAD und gib dieses aus:

```
git archive --verbose HEAD
```

- Erstelle ein zip-Archiv aus dem Inhalt des aktuellen HEAD und gib dieses aus:

```
git archive --verbose --format zip HEAD
```

- Erstelle ein zip-Archiv aus dem Inhalt des aktuellen HEAD und speichere dieses in eine Datei:

```
git archive --verbose --output {{pfad/zu/datei.zip}} HEAD
```

- Erstelle ein tar-Archiv aus dem Inhalt des letzten Commits eines bestimmten Branches:

```
git archive --output {{pfad/zu/datei.tar}} {{branch_name}}
```

- Erstelle ein tar-Archiv aus dem Inhalt eines bestimmten Verzeichnisses:

```
git archive --output {{pfad/zu/datei.tar}} HEAD:{{pfad/zu/verzeichnis}}
```

- Stelle jeder Datei einen Pfad voran, um sie in einem bestimmten Verzeichnis zu archivieren:

```
git archive --output {{pfad/zu/datei.tar}} --prefix {{pfad/zu/verzeichnis}}/ HEAD
```

# git bisect

Benutzt binäre Suche um den commit ausfindig zu machen, welcher einen Fehler beinhaltet.

Git springt im Commit-Graph automatisch vor und zurück, um den fehlerhaften Commit schrittweise eingrenzen zu können.

Weitere Informationen: <https://git-scm.com/docs/git-bisect>.

- Starte eine Bisect-Session in einem Commit-Bereich, der durch einen bekannten fehlerhaften Commit und einen sauberen Commit begrenzt wird:

```
git bisect start {{fehlerhafter_commit}} {{sauberer_commit}}
```

- Prüfe jeden Commit, den `git bisect` auswählt, und kennzeichne ihn mit "gut" oder "schlecht":

```
git bisect {{good|bad}}
```

- Wechsle zum vorherigen Branch zurück, nachdem der fehlerhafte Commit lokalisiert wurde:

```
git bisect reset
```

- Überspringe einen Commit während der Bisect-Session (z.B. einen, der die Tests aufgrund eines anderen Problems nicht bestanden hat):

```
git bisect skip
```

# git blame

Zeige den Commit-Hash und den letzten Autor jeder Zeile einer Datei.

Weitere Informationen: <https://git-scm.com/docs/git-blame>.

- Gib die Commit-Hashes und dem Autor jeder Zeile einer Datei aus:

```
git blame {{pfad/zu/datei}}
```

- Gib die Commit-Hashes und dem Autor (inklusive E-Mail) jeder Zeile einer Datei aus:

```
git blame -e {{pfad/zu/datei}}
```



# git branch

Verwalte und Arbeite mit Git Branches.

Weitere Informationen: <https://git-scm.com/docs/git-branch>.

- Liste alle Branches auf (Lokal und Remote). Der momentan aktive (ausgecheckte) Branch wird mit \* markiert:

```
git branch --all
```

- Liste alle Branches auf, welche einen spezifischen Git-Commit in ihrer Historie enthalten:

```
git branch --all --contains {{commit_hash}}
```

- Zeige den Namen des aktuellen Branches:

```
git branch --show-current
```

- Erstelle einen neuen Branch auf Basis des letzten Commits:

```
git branch {{branch_name}}
```

- Erstelle einen neuen Branch auf Basis eines bestimmten Commits:

```
git branch {{branch_name}} {{commit_hash}}
```

- Benenne einen Branch um (der Branch muss nicht ausgecheckt sein):

```
git branch -m {{alter_branch_name}} {{neuer_branch_name}}
```

- Lösche einen lokalen Branch (der Branch muss nicht ausgecheckt sein):

```
git branch -d {{branch_name}}
```

- Lösche einen remote-Branch:

```
git push {{remote_name}} --delete {{remote_branch_name}}
```

# git clone

Klone ein existierendes Repository.

Weitere Informationen: <https://git-scm.com/docs/git-clone>.

- Klone ein existierendes Repository in ein bestimmtes Verzeichnis:

```
git clone {{url_zu_repository}} {{pfad/zu/verzeichnis}}
```

- Klone ein existierendes Repository und seine Submodule:

```
git clone --recursive {{url_zu_repository}}
```

- Klone nur das .git Verzeichnis für ein existierendes repository:

```
git clone --no-checkout {{url_zu_repository}}
```

- Klone ein lokales Repository:

```
git clone --local {{pfad/zu/lokalem_repository}}
```

- Klone ohne Meldungen:

```
git clone --quiet {{url_zu_repository}}
```

- Klone ein existierendes Repository und rufe nur die neuesten 10 Commits im Standard-Branch ab (spart Zeit):

```
git clone --depth {{10}} {{url_zu_repository}}
```

- Klone ein existierendes Repository, aber lade nur einen bestimmten Branch herunter:

```
git clone --branch {{name}} --single-branch  
{{url_zu_repository}}
```

- Klone ein existierendes Repository mit einem bestimmten SSH Befehl:

```
git clone --config core.sshCommand="{{ssh -i pfad/zu/  
privat_ssh_schlüssel}}" {{url_zu_repository}}
```

# git commit

Committe Dateien in ein Repository.

Weitere Informationen: <https://git-scm.com/docs/git-commit>.

- Committe die gestagten Dateien mit einer Nachricht in das Repository:

```
git commit --message "{{nachricht}}"
```

- Committe alle gestagten Dateien zum Repository mit einer Nachricht aus einer Datei:

```
git commit --file {{pfad/zu/commit_nachricht_datei}}
```

- Stage alle modifizierten Dateien und committe sie mit einer Nachricht:

```
git commit --all --message "{{nachricht}}"
```

- Committe gestagten Dateien und signiere sie mit dem definierten GPG Schlüssel (oder mit dem in der Konfigurationsdatei definierten, wenn kein Argument angegeben ist):

```
git commit --gpg-sign {{key_id}} --message "{{nachricht}}"
```

- Ersetze den letzten Commit mit den gerade auf dem Stage liegenden Änderungen:

```
git commit --amend
```

- Comitte nur spezifische Dateien (die Dateien müssen schon auf dem Stage liegen):

```
git commit {{pfad/zu/datei1}} {{pfad/zu/datei2}}
```

- Erzeuge einen Commit, auch wenn keine Dateien auf dem Stage liegen:

```
git commit --message "{{nachricht}}" --allow-empty
```

# git config

Verwalten von benutzerdefinierten Optionen für Git Repositories.

Diese Optionen können lokal (für das aktive Repository) oder global (für den aktiven Benutzer) sein.

Weitere Informationen: <https://git-scm.com/docs/git-config>.

- Liste nur lokale Konfigurationseinträge (gespeichert unter `.git/config` im aktiven Repository) auf:

```
git config --list --local
```

- Liste nur globale Konfigurationseinträge (gespeichert unter `~/.gitconfig`) auf:

```
git config --list --global
```

- Liste nur System-Konfigurationseinträge (gespeichert unter `/etc/gitconfig`) und deren Speicherort auf:

```
git config --list --system --show-origin
```

- Gib den Wert eines bestimmten Konfigurationseintrags aus:

```
git config alias.unstage
```

- Setze den globalen Wert eines bestimmten Konfigurationseintrags:

```
git config --global alias.unstage "reset HEAD --"
```

- Setze den globalen Wert eines bestimmten Konfigurationseintrags auf seinen Standardwert zurück:

```
git config --global --unset alias.unstage
```

- Bearbeite die Git-Konfiguration für das aktuelle Repository mit dem Standard-Editor:

```
git config --edit
```

- Bearbeite die globale Git-Konfiguration mit dem Standard-Editor:

```
git config --global --edit
```

# git fetch

Lade Objekte und Referenzen (refs) von einem entfernten Repository.

Weitere Informationen: <https://git-scm.com/docs/git-fetch>.

- Hole die neuesten Änderungen von dem standardmäßigen Repository im Upstream (wenn gesetzt):

```
git fetch
```

- Hole neue Branches von einem bestimmten Repository im Upstream:

```
git fetch {{name_des_upstream_repository}}
```

- Hole die neuesten Änderungen von allen Repositories im Upstream:

```
git fetch --all
```

- Lade auch die Tags des Repository im Upstream:

```
git fetch --tags
```

- Lösche lokale Referenzen auf entfernte Branches, die im Upstream-Repository nicht mehr existieren:

```
git fetch --prune
```

# git format-patch

Erstelle **.patch** Dateien. Ermöglicht das Senden von Commits per E-Mail.

Siehe auch **git am**, womit **.patch** Dateien lokal angewandt werden.

Weitere Informationen: <https://git-scm.com/docs/git-format-patch>.

- Erzeuge eine **.patch** Datei aus allen nicht gepushten Commits:

```
git format-patch {{origin}}
```

- Erstelle eine **.patch** Datei aus allen Commits zwischen den angegebenen Revisionen und schreibe diese nach stdout:

```
git format-patch {{revision_1}}..{{revision_2}}
```

- Erstelle eine **.patch** Datei aus den letzten 3 Commits:

```
git format-patch -{{3}}
```

# git help

Zeige Hilfe für Git an.

Weitere Informationen: <https://git-scm.com/docs/git-help>.

- Zeige Hilfe für einen bestimmten Git-Unterbefehl an:

```
git help {{unterbefehl}}
```

- Zeige Hilfe für einen bestimmten Git-Unterbefehl im Web-Browser an:

```
git help --web {{unterbefehl}}
```

- Zeige eine Liste von allen verfügbaren Git-Unterbefehlen an:

```
git help --all
```

- Liste die verfügbaren Handbücher auf:

```
git help --guide
```

- Liste alle Konfigurationsvariablen auf:

```
git help --config
```

# git ignore-io

Erstelle **.gitignore** Dateien aus vorgefertigten Vorlagen.

Weitere Informationen: <https://github.com/tj/git-extras/blob/master/Commands.md#git-ignore-io>.

- Liste alle verfügbaren Vorlagen auf:

```
git ignore-io list
```

- Erstelle eine .gitignore Vorlage:

```
git ignore-io {{eintrag_a,eintrag_b,eintrag_n}}
```



# git init

Erstelle eine neues lokales Git-Repository.

Weitere Informationen: <https://git-scm.com/docs/git-init>.

- Erstelle eine neues lokales Repository:

```
git init
```

- Erstelle eine neues Repository mit einem bestimmten Namen für den ersten Branch:

```
git init --initial-branch={{branch_name}}
```

- Erstelle ein neues Repository, welches SHA256 für Objekt-Hashes verwendet (benötigt Git 2.29+):

```
git init --object-format={{sha256}}
```

- Erstelle eine neues minimales Repository, welches sich für die Verwendung als Remote-Repository über SSH eignet:

```
git init --bare
```

# git log

Zeigt die Commit-Historie an.

Weitere Informationen: <https://git-scm.com/docs/git-log>.

- Zeige die Sequenz der Commits des Git-Repository im aktuellen Verzeichnis, beginnend mit dem aktuellen, an:

```
git log
```

- Zeige die Historie einer bestimmten Datei oder eines Verzeichnisses, inklusive Unterschiede, an:

```
git log -p {{pfad/zu/datei_oder_verzeichnis}}
```

- Zeige einen Überblick der Commits an und welche Dateien jeweils verändert wurden:

```
git log --stat
```

- Zeige einen Graphen von Commits im aktuellen Branch, wobei jeweils nur die erste Zeile der Commit-Nachricht angezeigt wird:

```
git log --oneline --graph
```

- Zeige einen Graphen von allen Commits, Tags und Branches im gesamten Repository:

```
git log --oneline --decorate --all --graph
```

- Zeige nur Commits, deren Commit-Nachricht einen bestimmten Text enthalten (Ohne Beachtung von Groß- und Kleinschreibung):

```
git log -i --grep {{text}}
```

- Zeige die letzten N Commits eines bestimmten Autors:

```
git log -n {{anzahl}} --author={{autor}}
```

- Zeige alle Commits zwischen zwei Zeitpunkten an (yyyy-mm-dd):

```
git log --before="{{2017-01-29}}" --after="{{2017-01-17}}"
```

# git pull

Hole Branches von einem entfernten Repository und binde sie in das lokale Repository ein.

Weitere Informationen: <https://git-scm.com/docs/git-pull>.

- Lade Änderungen vom Standard-Repository herunter und führe diese zusammen:

```
git pull
```

- Lade Änderungen vom Standard-Repository herunter und wende einen Rebase an:

```
git pull --rebase
```

- Lade Änderungen vom Standard-Repository herunter und führe diese in den HEAD zusammen:

```
git pull {{remote_name}} {{branch}}
```

# git push

Lade Commits in ein Remote-Repository hoch.

Weitere Informationen: <https://git-scm.com/docs/git-push>.

- Sende lokale Änderungen des aktuellen Branches zu seinem entfernten Repository (Remote Branch):

```
git push
```

- Sende lokale Änderungen des angegebenen Branches zu seinem entfernten Repository:

```
git push {{remote_name}} {{lokaler_branch}}
```

- Lade den aktuellen Branch in ein entferntes Repository mit Angabe des Namens des entfernten Branches hoch:

```
git push -u {{remote_name}} {{remote_branch}}
```

- Lade Änderungen eines bestimmten lokalen Branches zu einem bestimmten entfernten Branch hoch:

```
git push {{remote_name}} {{lokaler_branch}}:  
{{entfernter_branch}}
```

- Lade Änderungen aller lokalen Branches zu ihrem entfernten Repository hoch:

```
git push --all {{remote_name}}
```

- Lösche einen Branch in einem entfernten Repository:

```
git push {{remote_name}} --delete {{remote_branch}}
```

- Entferne alle remote Branches, welche kein lokales Gegenstück besitzen:

```
git push --prune {{remote_name}}
```

- Veröffentliche Tags, welche noch nicht im entfernten Repository vorhanden sind:

```
git push --tags
```

# git rebase

Wende Commits von einem Branch auf einen anderen Branch an.

Die Änderungen eines Branches werden auf einen bestehenden Branch "übertragen" und am Ende der Historie als neue Commits eingefügt.

Weitere Informationen: <https://git-scm.com/docs/git-rebase>.

- Verwende einen anderen, angegebenen Branch als Basis für den aktuellen Branch:

```
git rebase {{neuer_basisbranch}}
```

- Starte einen interaktiven Rebase, bei dem Commits umsortiert, weggelassen, kombiniert oder verändert werden können:

```
git rebase -i {{ziel_basisbranch_oder_commithash}}
```

- Setze einen Rebase fort, der durch einen Mergefehler unterbrochen wurde, nachdem die Konflikte aufgelöst wurden:

```
git rebase --continue
```

- Setze einen Rebase fort, der durch einen Mergefehler unterbrochen wurde, durch Auslassen des in Konflikt stehenden Commits:

```
git rebase --skip
```

- Brich einen laufenden Rebase ab (z.B. wenn er durch Mergekonflikte unterbrochen wurde):

```
git rebase --abort
```

- Verschiebe einen Teil des aktuellen Branches auf eine neue Basis und gib die alte Basis an, ab der die Änderungen verwendet werden sollen:

```
git rebase --onto {{neue_basis}} {{alte_basis}}
```

- Bearbeite die 5 letzten Commits der aktuellen Basis um diese neu zu ordnen, auszulassen, kombinieren oder zu bearbeiten:

```
git rebase -i {{HEAD~5}}
```

- Löse Konflikte automatisch auf, indem der aktuelle Branch bevorzugt wird (das Schlüsselwort `theirs` hat in diesem Fall eine umgekehrte Bedeutung):

```
git rebase -X theirs {{branch_name}}
```

# git remote

Verwalte eine gewisse Anzahl an Repositories (remotes).

Weitere Informationen: <https://git-scm.com/docs/git-remote>.

- Liste alle existierenden Remotes, ihre Namen und ihre URLs auf:

```
git remote -v
```

- Zeige Informationen über ein Remote an:

```
git remote show {{remote_name}}
```

- Füge ein neues Remote hinzu:

```
git remote add {{remote_name}} {{remote_url}}
```

- Ändere die URL eines Remotes (benutze `--add` um die existierende URL zu behalten):

```
git remote set-url {{remote_name}} {{remote_url}}
```

- Entferne ein Remote:

```
git remote remove {{remote_name}}
```

- Benenne ein Remote um:

```
git remote rename {{alter_name}} {{neuer_name}}
```

# git rm

Entferne Dateien aus dem Index des Repositories und vom lokalen Dateisystem.

Weitere Informationen: <https://git-scm.com/docs/git-rm>.

- Entferne eine Datei aus dem Index und vom lokalen Dateisystem:

```
git rm {{pfad/zu/datei}}
```

- Entferne ein Verzeichnis:

```
git rm -r {{pfad/zu/verzeichnis}}
```

- Entferne eine Datei aus dem Index des Repositories, aber behalte sie lokal:

```
git rm --cached {{pfad/zu/datei}}
```



# git send-email

Sende eine Menge von Patches als E-Mail. Patches können als Dateien, Ordner oder Liste von Revisionen spezifiziert werden.

Weitere Informationen: <https://git-scm.com/docs/git-send-email>.

- Sende den letzten Commit des aktuellen Branches:

```
git send-email -1
```

- Sende einen spezifischen Commit:

```
git send-email -1 {{commit}}
```

- Sende die letzten (z.B. 10) Commits des aktuellen Branches:

```
git send-email {{-10}}
```

- Editiere eine E-Mail mit einer Reihe von Patches im Standardmailclienten:

```
git send-email -{{anzahl_an_commits}} --compose
```

- Bearbeite den E-Mail Text jedes der zu versendenden Patches:

```
git send-email -{{anzahl_an_commits}} --annotate
```

# git setup

Erstelle ein Git Repository in einem bestimmten Verzeichnis und committe alle Dateien.

Teil der **git-extras**.

Weitere Informationen: <https://github.com/tj/git-extras/blob/master/Commands.md#git-setup>.

- Erstelle ein Git Repository im aktuellen Verzeichnis und committe alle Dateien:

```
git setup
```

- Erstelle ein Git Repository in einem bestimmten Verzeichnis und committe alle Dateien:

```
git setup {{pfad/zu/verzeichnis}}
```

# git show-merged-branches

Gibt alle Branches aus, welche bereits in die aktuelle HEAD-Datei gemerget wurden.

Weitere Informationen: <https://github.com/tj/git-extras/blob/master/Commands.md#git-show-merged-branches>.

- Gib alle Branches aus, die bereits in die aktuelle HEAD-Datei gemerget wurden:

```
git show-merged-branches
```

# git show-unmerged-branches

Gibt alle Branches aus, welche noch nicht in die aktuelle HEAD-Datei gemerged wurden.

Weitere Informationen: <https://github.com/tj/git-extras/blob/master/Commands.md#git-show-unmerged-branches>.

- Gib alle Branches aus, die noch nicht in die aktuelle HEAD-Datei gemerged wurden:

```
git show-unmerged-branches
```

# git status

Zeige die Änderungen an Dateien in einem Git-Repository an.

Weitere Informationen: <https://git-scm.com/docs/git-status>.

- Zeige veränderte Dateien an, die noch nicht für den Commit hinzugefügt wurden:

```
git status
```

- Zeige eine kurze Version an:

```
git status -s
```

- Zeige nur verfolgte (getrackte) Dateien an:

```
git status --untracked-files=no
```

- Zeige eine kurze Version mit zusätzlichen Informationen über den Branch an:

```
git status -sb
```

# git switch

Wechsle zwischen Branches. Verfügbar ab Git Version 2.23+.

Siehe auch **git checkout**.

Weitere Informationen: <https://git-scm.com/docs/git-switch>.

- Wechsle zu einem existierenden Branch:

```
git switch {{branche_name}}
```

- Erstelle einen neuen Branch und wechsele zu diesem:

```
git switch --create {{branch_name}}
```

- Erstelle einen neuen Branch basierend auf einem existierenden Commit und wechsele zu diesem:

```
git switch --create {{branch_name}} {{commit}}
```

- Wechsele zum vorherigen Branch:

```
git switch -
```

- Wechsele zu einem Branch und aktualisiere alle Submodule entsprechend:

```
git switch --recurse-submodules {{branch_name}}
```

- Wechsele zu einem Branch und merge automatisch den aktuellen Branch und alle Änderungen, die nicht committed wurden:

```
git switch --merge {{branch_name}}
```

# git tag

Erstelle, lösche, überprüfe und liste Tags auf.

Weitere Informationen: <https://git-scm.com/docs/git-tag>.

- Liste alle Tags auf:

```
git tag
```

- Erstelle einen Tag mit Namen, welcher auf den aktuellen Commit zeigt:

```
git tag {{tag_name}}
```

- Erstelle einen Tag mit Namen, welcher auf einen bestimmten Commit zeigt:

```
git tag {{tag_name}} {{commit}}
```

- Erstelle einen Tag mit Anmerkung:

```
git tag {{tag_name}} -m {{anmkerung}}
```

- Lösche einen Tag mit bestimmten Namen:

```
git tag -d {{tag_name}}
```

- Lade die aktualisierten Tags aus dem Upstream:

```
git fetch --tags
```

- Liste alle Tags auf, bei denen sich in den vorangegangenen Commits ein bestimmter Commit findet:

```
git tag --contains {{commit}}
```

# git

Verteiltes Versionskontrollsystem.

Manche Unterbefehle wie **git commit** sind separat dokumentiert.

Weitere Informationen: <https://git-scm.com/>.

- Gib die installierte Git Version aus:

```
git --version
```

- Zeige die generelle Hilfeseite an:

```
git --help
```

- Zeige die Hilfeseite eines Unterbefehls an:

```
git help {{unterbefehl}}
```

- Führe einen Git-Unterbefehl aus:

```
git {{unterbefehl}}
```

- Führe einen Git-Unterbefehl auf einem benutzerdefinierten Repository aus:

```
git -C {{pfad/zu/repository}} {{unterbefehl}}
```

- Führe einen Git-Unterbefehl mit der angegebenen Konfiguration aus:

```
git -c '{{config.key}}={{wert}}' {{unterbefehl}}
```



# gnmic sub

Dieser Befehl ist ein Alias von **gnmic subscribe**.

Weitere Informationen: <https://gnmic.kmrd.dev/cmd/subscribe>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr gnmic subscribe
```

# gobuster

Findet versteckte Pfade auf Webservern und mehr.

Weitere Informationen: <https://github.com/OJ/gobuster>.

- Finde Verzeichnisse und Dateien, die den Wörtern der Wortliste entsprechen:

```
gobuster dir --url {{https://beispiel.com/}} --wordlist  
{{pfad/zu/datei}}
```

- Finde Subdomains:

```
gobuster dns --domain {{beispiel.com}} --wordlist {{pfad/zu/  
datei}}
```

- Finde Amazon S3-Buckets:

```
gobuster s3 --wordlist {{pfad/zu/datei}}
```

- Finde andere virtuelle Hosts eines Servers:

```
gobuster vhost --url {{https://beispiel.com/}} --wordlist  
{{pfad/zu/datei}}
```

- Fuzz den Wert eines URL-Parameters:

```
gobuster fuzz --url {{https://beispiel.com/?parameter=FUZZ}}  
--wordlist {{pfad/zu/datei}}
```

- Fuzz den Namen eines URL-Parameters:

```
gobuster fuzz --url {{https://beispiel.com/?FUZZ=wert}} --  
wordlist {{pfad/zu/datei}}
```

# google-chrome

Dieser Befehl ist ein Alias von **chromium**.

Weitere Informationen: <https://chrome.google.com>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr chromium
```

# gpg

GNU Privacy Guard.

Siehe **gpg2** für GNU Privacy Guard 2.

Weitere Informationen: <https://gnupg.org>.

- Erstelle einen öffentlichen und privaten GPG Schlüssel interaktiv:

```
gpg --full-generate-key
```

- Signiere `doc.txt` ohne Verschlüsselung (Ausgabe nach `doc.txt.asc`):

```
gpg --clearsign {{doc.txt}}
```

- Verschlüsse und signiere `doc.txt` für `alice@beispiel.de` und `bob@example.org` (Ausgabe nach `doc.txt.gpg`):

```
gpg --encrypt --sign --recipient {{alice@beispiel.de}} --recipient {{bob@example.com}} {{doc.txt}}
```

- Verschlüsse `doc.txt` nur mit Passwort (Ausgabe nach `doc.txt.gpg`):

```
gpg --symmetric {{doc.txt}}
```

- Entschlüsse `doc.txt.gpg` (Ausgabe nach `stdout`):

```
gpg --decrypt {{doc.txt.gpg}}
```

- Importiere einen öffentlichen Schlüssel:

```
gpg --import {{schlüssel.gpg}}
```

- Exportiere den öffentlichen Schlüssel von `alice@beispiel.de` (Ausgabe nach `stdout`):

```
gpg --export --armor {{alice@beispiel.de}}
```

- Exportiere den privaten Schlüssel von `alice@beispiel.de` (Ausgabe nach `stdout`):

```
gpg --export-secret-keys --armor {{alice@beispiel.de}}
```

# gpg2

GNU Privacy Guard 2.

Siehe **gpg** für GNU Privacy Guard 1.

Weitere Informationen: <https://docs.renong.linuxfoundation.org/en/latest/gpg.html>.

- Liste alle importierten Schlüssel auf:

```
gpg2 --list-keys
```

- Verschlüsse eine bestimmte Datei für einen bestimmten Empfänger und schreibe den Output in eine neue .gpg Datei:

```
gpg2 --encrypt --recipient {{hans@beispiel.de}} {{pfad/zu/datei.txt}}
```

- Verschlüsse eine bestimmte Datei nur mit einem Passwort und schreibe den Output in eine neue .gpg Datei:

```
gpg2 --symmetric {{pfad/zu/datei.txt}}
```

- Verschlüsse eine bestimmte Datei und schreibe das Ergebnis auf STDOUT:

```
gpg2 --decrypt {{pfad/zu/datei.txt.gpg}}
```

- Importiere einen öffentlichen Schlüssel:

```
gpg2 --import {{pfad/zu/öffentlichem_schlüssel.gpg}}
```

- Exportiere den öffentlichen Schlüssel einer bestimmten Email-Adresse nach STDOUT:

```
gpg2 --export --armor {{hans@beispiel.de}}
```

- Exportiere den privaten Schlüssel einer bestimmten Email-Adresse nach STDOUT:

```
gpg2 --export-secret-keys --armor {{hans@beispiel.de}}
```

# grep

Findet Ausdrücke in einem Eingabetext.

Unterstützt einfache Muster und reguläre Ausdrücke.

Weitere Informationen: <https://www.gnu.org/software/grep/manual/grep.html>.

- Suche nach einem Ausdruck in einer Datei:

```
grep "{{ausdruck}}" {{pfad/zu/datei}}
```

- Suche nach einem exakten Ausdruck:

```
grep {{-F|--fixed-strings}} "{{exakter_ausdruck}}" {{pfad/zu/datei}}
```

- Benutze erweiterte reguläre Ausdrücke (unterstützt ?, +, {}, () und |) ohne Beachtung der Groß-, Kleinschreibung:

```
grep {{-E|--extended-regexp}} {{-i|--ignore-case}}  
"{{ausdruck}}" {{pfad/zu/datei}}
```

- Zeige 3 Zeilen Kontext um [C], vor [B] oder nach [A] jedem Ergebnis:

```
grep --{{context|before-context|after-context}} 3  
"{{ausdruck}}" {{pfad/zu/datei}}
```

# head

Gibt den ersten Teil einer Datei aus.

Weitere Informationen: <https://manned.org/head.1p>.

- Gib die ersten paar Zeilen einer Datei aus:

```
head -n {{anzahl_an_zeilen}} {{datei}}
```

- Gib die ersten Bytes einer Datei aus:

```
head -c {{anzahl_an_bytes}} {{datei}}
```

- Gib alle bis auf die letzten Zeilen einer Datei aus:

```
head -n -{{anzahl_an_zeilen}} {{datei}}
```

- Gib alle bis auf die letzten Bytes einer Datei aus:

```
head -c -{{anzahl_an_bytes}} {{datei}}
```

# hx

Dieser Befehl ist ein Alias von **helix**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr helix
```



# ical

Ein islamischer (Hijri) Kalender und Konverter für das Terminal.

Weitere Informationen: <https://manned.org/ical>.

- Zeige den Kalender von diesem Monat an:

```
ical
```

- Konvertiere ein gregorianisches Datum zu einem islamischen Datum:

```
ical --gregorian {{yyyymmdd}}
```

- Konvertiere ein islamisches Datum zu einem gregorianischen Datum:

```
ical --hijri {{yyyymmdd}}
```

# id

Zeige den aktuellen Benutzer und dessen Gruppenzugehörigkeit an.

Weitere Informationen: <https://www.gnu.org/software/coreutils/id>.

- Zeige die Benutzer-ID (UID), die Gruppen-ID (GID) und die Gruppen des aktuellen Benutzers an:

```
id
```

- Zeige die Benutzer-ID (UID) an:

```
id -u
```

- Zeige die Gruppen-ID (GID) an:

```
id -g
```

- Zeige die Benutzer-ID (UID), die Gruppen-ID (GID) und die Gruppen eines beliebigen Benutzers an:

```
id {{benutzername}}
```

# jq

Ein JSON-Verarbeiter für die Kommandozeile mit einer domänenspezifischen Sprache.

Weitere Informationen: <https://jqlang.github.io/jq/manual/>.

- Führe den angegebenen Ausdruck aus (gib farbiges und formatiertes JSON aus):

```
{{cat pfad/zu/datei.json}} | jq '.'
```

- Führe ein gegebenes Skript aus:

```
{{cat pfad/zu/datei.json}} | jq --from-file {{pfad/zu/skript.jq}}
```

- Übergib bestimmte Argumente:

```
{{cat pfad/zu/datei.json}} | jq {{--arg "name1" "wert1" --arg "name2" "wert2" ...}} '{{. + $ARGS.named}}'
```

- Gib bestimmte Schlüssel aus:

```
{{cat pfad/zu/datei.json}} | jq '{{.schlüssel1, .schlüssel2, ...}}'
```

- Gib bestimmte Listenelemente aus:

```
{{cat pfad/zu/datei.json}} | jq '{{.[index1], .[index2], ...}}'
```

- Gib alle Listenelemente/Objektschlüssel aus:

```
{{cat pfad/zu/datei.json}} | jq '.[[]]'
```

- Füge bestimmte Schlüssel hinzu/lösche bestimmte Schlüssel:

```
{{cat pfad/zu/datei.json}} | jq '. {{+|-}} {{{"schlüssel1": "wert1", "schlüssel2": "wert2", ...}}}'
```

# k6

Open-Source Tool und SaaS für Lasttests.

Weitere Informationen: <https://k6.io>.

- Führe einen Lasttest lokal aus:

```
k6 run {{skript.js}}
```

- Führe einen Lasttest lokal mit einer festgelegten Anzahl von virtuellen Benutzern und Dauer aus:

```
k6 run --vus {{10}} --duration {{30s}} {{skript.js}}
```

- Führe einen Lasttest lokal mit einer gesetzten Umgebungsvariablen aus:

```
k6 run -e {{HOSTNAME=beispiel.com}} {{skript.js}}
```

- Führe einen Lasttest lokal aus und speichere die Ergebnisse in InfluxDB:

```
k6 run --out influxdb={{http://localhost:8086/k6db}}  
{{skript.js}}
```

- Führe einen Lasttest lokal aus und verwirfe Antworten (deutlich schneller):

```
k6 run --discard-response-bodies {{skript.js}}
```

- Führe einen Lasttest lokal im JavaScript-Kompatibilitätsmodus aus (deutlich schneller):

```
k6 run --compatibility-mode=base {{skript.js}}
```

- Melde dich bei einem Cloud-Dienst mit einem geheimen Token an:

```
k6 login cloud --token {{geheim}}
```

- Führe einen Lasttest auf Cloud-Infrastruktur aus:

```
k6 cloud {{skript.js}}
```

# kafkacat

Dieser Befehl ist ein Alias von **kcat**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr kcat
```

# khal

Eine textbasierte Kalender- und Planungsanwendung für die Kommandozeile.

Weitere Informationen: <https://lostpackets.de/khal>.

- Starte khal im interaktiven Modus:

```
ikhal
```

- Gib alle Termine aus, die im Kalender "privat" in den nächsten sieben Tagen geplant sind:

```
khal list -a {{privat}} {{today}} {{7d}}
```

- Gib alle Termine aus, die in Kalendern außer "privat" für morgen um 10 Uhr geplant sind:

```
khal at -d {{privat}} {{tomorrow}} {{10:00}}
```

- Gib einen Kalender mit einer Liste an Terminen für die nächsten drei Monate aus:

```
khal calendar
```

- Füge dem Kalender "privat" einen neuen Termin hinzu:

```
khal new -a {{privat}} {{2020-09-08}} {{18:00}} {{18:30}}  
"{{Zahnarzttermin}}"
```

# kitty

Ein schneller, funktionsreicher, auf der GPU basierender Terminal-Emulator.

Weitere Informationen: <https://sw.kovidgoyal.net/kitty/>.

- Öffne ein neues Terminal:

```
kitty
```

- Öffne ein Terminal mit einem festgelegten Titel für das Fenster:

```
kitty --title "{{Titel}}"
```

- Starte die integrierte Farbschema-Auswahl:

```
kitty +kitten themes
```

- Zeige ein Bild im Terminal an:

```
kitty +kitten icat {{pfad/zum/bild}}
```

- Kopiere den Inhalt von `stdin` in die Zwischenablage:

```
echo {{Beispiel}} | kitty +kitten clipboard
```

# kubectl delete

Lösche Kubernetes-Ressourcen.

Weitere Informationen: <https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#delete>.

- Lösche einen bestimmten Pod:

```
kubectl delete pod {{pod_name}}
```

- Lösche ein bestimmtes Deployment:

```
kubectl delete deployment {{deployment_name}}
```

- Lösche eine bestimmte Node:

```
kubectl delete node {{node_name}}
```

- Lösche alle Pods in einem bestimmten Namespace:

```
kubectl delete pods --all --namespace {{namespace}}
```

- Lösche alle Deployments und Services in einem bestimmten Namespace:

```
kubectl delete deployments,services --all --namespace {{namespace}}
```

- Lösche alle Nodes:

```
kubectl delete nodes --all
```

- Lösche Ressourcen, die in einer YAML Datei definiert sind:

```
kubectl delete --filename {{pfad/zu/manifest.yaml}}
```



# kubectl describe

Details von Kubernetes-Objekten und -Ressourcen anzeigen.

Weitere Informationen: <https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#describe>.

- Zeige Details zu Pods in einem bestimmten [n]amespace an:

```
kubectl describe pods --namespace {{namespace}}
```

- Zeige Details zu Nodes in einem bestimmten [n]amespace an:

```
kubectl describe nodes --namespace {{namespace}}
```

- Zeige Details zu einem bestimmten Pod in einem bestimmten [n]amespace an:

```
kubectl describe pods {{pod_name}} --namespace {{namespace}}
```

- Zeige Details zu einer bestimmten Node in einem bestimmten [n]amespace an:

```
kubectl describe nodes {{node_name}} --namespace {{namespace}}
```

- Zeige Details zu Ressourcen, die in einer YAML Datei definiert sind, an:

```
kubectl describe --file {{pfad/zu/manifest.yaml}}
```

# kubectl get

Abfragen von Kubernetes Ressourcen und Objekten.

Weitere Informationen: <https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#get>.

- Zeige alle Namespaces im Cluster an:

```
kubectl get namespaces
```

- Frage alle Nodes in einem bestimmten [n]amespace ab:

```
kubectl get nodes --namespace {{namespace}}
```

- Frage alle Pods in einem bestimmten [n]amespace ab:

```
kubectl get pods --namespace {{namespace}}
```

- Frage alle Deployments in einem bestimmten [n]amespace ab:

```
kubectl get deployments --namespace {{namespace}}
```

- Frage alle Services in einem bestimmten [n]amespace ab:

```
kubectl get services --namespace {{namespace}}
```

- Frage alle Ressourcen in einem bestimmten [n]amespace ab:

```
kubectl get all --namespace {{namespace}}
```

- Frage alle Ressourcen ab, die in einer YAML Datei definiert sind:

```
kubectl get --file {{pfad/zum/manifest.yaml}}
```

# kubectl logs

Logs für Container in einem Pod anzeigen.

Weitere Informationen: <https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#logs>.

- Zeige Logs für einen Einzelcontainer-Pod an:

```
kubectl logs {{pod_name}}
```

- Zeige Logs für einen bestimmten Container in einem Pod an:

```
kubectl logs --container {{container_name}} {{pod_name}}
```

- Zeige Logs für alle Container in einem Pod an:

```
kubectl logs --all-containers={{true}} {{pod_name}}
```

- Folge den Pod-Logs (stream):

```
kubectl logs --follow {{pod_name}}
```

- Folge den Pod-Logs (stream) für einen bestimmten Container in einem Pod:

```
kubectl logs --follow --container {{container_name}}  
{{pod_name}}
```

- Zeige Pod-Logs die neuer einer relativen Zeit sind 10s, 5m, or 1h:

```
kubectl logs --since={{relative_time}} {{pod_name}}
```

- Zeige die 10 neuesten Logzeilen in einem Pod:

```
kubectl logs --tail={{10}} {{pod_name}}
```

# kubectl rollout

Verwalten des Rollouts einer Kubernetes-Ressource (deployments, daemonsets, and statefulsets).

Weitere Informationen: <https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#rollout>.

- Starte einen rollenden Neustart einer Ressource:

```
kubectl rollout restart {{resource_type}}/{{resource_name}}
```

- Überwache den fortlaufenden Aktualisierungsstatus einer Ressource:

```
kubectl rollout status {{resource_type}}/{{resource_name}}
```

- Setze eine Ressource auf die vorherige Version zurück:

```
kubectl rollout undo {{resource_type}}/{{resource_name}}
```

- Zeige den Rollout-Verlauf einer Ressource an:

```
kubectl rollout history {{resource_type}}/{{resource_name}}
```

# kubectl run

Pods in Kubernetes ausführen. Gibt den Pod-Generator an, um einen deprecation Fehler in einigen Kubernetes Versionen zu vermeiden.

Weitere Informationen: <https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#run>.

- Starte einen nginx-Pod und gib Port 80 frei:

```
kubectl run {{nginx-dev}} --image=nginx --port 80
```

- Starte einen nginx-Pod und setze die Umgebungsvariable TEST\_VAR:

```
kubectl run {{nginx-dev}} --image=nginx --env="{{TEST_VAR}}={{testing}}"
```

- Zeige API-Aufrufe an, die zur Erstellung eines Nginx-Containers erfolgen würden:

```
kubectl run {{nginx-dev}} --image=nginx --dry-run={{none|server|client}}
```

- Führe einen Ubuntu-Pod interaktiv aus, starte ihn nie neu und entferne ihn, wenn er beendet wird:

```
kubectl run {{temp-ubuntu}} --image=ubuntu:22.04 --restart=Never --rm -- /bin/bash
```

- Führe einen Ubuntu-Pod aus, überschreibe den Standardbefehl mit echo und gib eigene Argumente an:

```
kubectl run {{temp-ubuntu}} --image=ubuntu:22.04 --command -- echo {{argument1 argument2 ...}}
```

# kubectl

Befehlszeilenschnittstelle zur Ausführung von Befehlen gegen Kubernetes-Cluster.

Einige Unterbefehle wie **kubectl run** haben ihre eigene Dokumentation zur Verwendung.

Weitere Informationen: <https://kubernetes.io/docs/reference/kubectl/>.

- Liste Informationen über eine Ressource mit weiteren Details auf:

```
kubectl get {{pod|service|deployment|ingress|...}} -o wide
```

- Aktualisiere die angegebenen Pods mit dem Label 'unhealthy' und dem Wert 'true':

```
kubectl label pods {{name}} unhealthy=true
```

- Liste alle Ressourcen aller Typen auf:

```
kubectl get all
```

- Zeige die Ressourcennutzung (CPU/Memory/Storage) von Knoten oder Pods:

```
kubectl top {{pod|node}}
```

- Zeige die Adresse der Master- und Clusterdienste:

```
kubectl cluster-info
```

- Zeige eine Erklärung zu einem bestimmten Feld an:

```
kubectl explain {{pods.spec.containers}}
```

- Zeige Logs für einen Container in einem Pod oder einer bestimmten Ressource:

```
kubectl logs {{pod_name}}
```

- Führe einen Befehl in einem bestehenden Pod aus:

```
kubectl exec {{pod_name}} -- {{ls /}}
```

# lastcomm

Zeige die zuletzt ausgeführten Befehle an.

Weitere Informationen: <https://manpages.debian.org/latest/acct/lastcomm.1.en.html>.

- Gib Informationen zu allen Befehlen in acct aus (Aufzeichnungsdatei):

```
lastcomm
```

- Zeige die ausgeführten Befehle eines bestimmten Benutzers an:

```
lastcomm --user {{benutzer}}
```

- Zeige Informationen zu einem bestimmten Befehl an, der auf dem System ausgeführt wird:

```
lastcomm --command {{befehl}}
```

- Zeige Informationen zu Befehlen an, die auf einem bestimmten Terminal ausgeführt wurden:

```
lastcomm --tty {{terminal_name}}
```

# latex

Kompiliere eine LaTeX Quelldatei in ein DVI Dokument.

Weitere Informationen: <https://www.latex-project.org>.

- Kompiliere ein DVI Dokument:

```
latex {{quelldatei.tex}}
```

- Kompiliere ein DVI Dokument und gib ein bestimmtes Output-Verzeichnis an:

```
latex -output-directory={{pfad/zu/verzeichnis}}  
{{quelldatei.tex}}
```

- Kompiliere ein DVI Dokument und stoppe bei jedem Fehler:

```
latex -halt-on-error {{quelldatei.tex}}
```



# less

Öffne eine Datei für interaktives lesen, erlaubt scrollen und suchen.

Weitere Informationen: <https://greenwoodsoftware.com/less/>.

- Öffne eine Datei:

```
less {{pfad/zu/datei}}
```

- Scrolle eine Seite runter / hoch:

```
<Leertaste> (runter), b (hoch)
```

- Springe zum Ende / Anfang der Datei:

```
G (Ende), g (Pos1)
```

- Suche nach einer Zeichenkette vorwärts (n/N um zur nächsten/vorherigen Übereinstimmung zu springen):

```
/{{suche}}
```

- Suche nach einer Zeichenkette rückwärts (n/N um zur nächsten/vorherigen Übereinstimmung zu springen):

```
?{{suche}}
```

- Folge der Ausgabe des geöffneten Buffers:

```
F
```

- Öffne die Datei in einem Editor:

```
v
```

- Beende less:

```
q
```

# llvm-ar

Dieser Befehl ist ein Alias von **ar**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr ar
```

# llvm-g++

Dieser Befehl ist ein Alias von **clang++**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr clang++
```

# llvm-gcc

Dieser Befehl ist ein Alias von **clang**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr clang
```

# llvm-nm

Dieser Befehl ist ein Alias von **nm**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr nm
```

# llvm-objdump

Dieser Befehl ist ein Alias von **objdump**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr objdump
```

# llvm-strings

Dieser Befehl ist ein Alias von **strings**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr strings
```

# lolcat

Färbe Text in Regenbogenfarben ein.

Weitere Informationen: <https://github.com/busyloop/lolcat>.

- Gib den Inhalt einer Datei in Regenbogenfarben in der Konsole aus:

```
lolcat {{pfad/zu/datei}}
```

- Gib die Ausgabe eines Befehls in Regenbogenfarben in der Konsole aus:

```
{{fortune}} | lolcat
```

- Gib den Inhalt einer Datei in animierten Regenbogenfarben in der Konsole aus:

```
lolcat -a {{pfad/zu/datei}}
```

- Gib den Inhalt einer Datei in 24-bit (truecolor) Regenbogenfarben in der Konsole aus:

```
lolcat -t {{pfad/zu/datei}}
```



# lp

Druckt Dateien.

Weitere Informationen: <https://manned.org/lp>.

- Drucke die Ausgabe eines Befehls mit dem Standard-Drucker (siehe `lpstat`):

```
echo "test" | lp
```

- Drucke eine Datei mit dem Standard-Drucker:

```
lp {{pfad/zu/datei}}
```

- Drucke eine Datei mit einem bestimmten Drucker (siehe `lpstat`):

```
lp -d {{druckername}} {{pfad/zu/datei}}
```

- Drucke N Kopien einer Datei mit dem Standarddrucker (wobei N die Anzahl gewünschter Kopien ist):

```
lp -n {{N}} {{pfad/zu/datei}}
```

- Drucke nur bestimmte Seiten mit dem Standarddrucker (drucke Seiten 1, 3-5 und 16):

```
lp -P 1,3-5,16 {{pfad/zu/datei}}
```

- Führe einen aufgehaltene Druckauftrag durch:

```
lp -i {{job_id}} -H resume
```

# lpinfo

Liste verbundene Drucker und installierte Treiber für den CUPS Druckserver.

Weitere Informationen: <https://openprinting.github.io/cups/doc/man-lpinfo.html>.

- Liste alle aktuell verbundenen Drucker auf:

```
lpinfo -v
```

- Liste alle aktuell installierten Druckertreiber auf:

```
lpinfo -m
```

- Suche installierte Druckertreiber nach Hersteller oder Modell:

```
lpinfo --make-and-model "{{druckermodell}}" -m
```

# lpr

CUPS-Programm zum Drucken von Dateien.

Siehe auch: **lpstat** und **lpadmin**.

Weitere Informationen: <https://openprinting.github.io/cups/doc/man-lpr.html>.

- Drucke eine Datei mit dem Standarddrucker:

```
lpr {{pfad/zu/datei}}
```

- Drucke 2 Kopien einer Datei:

```
lpr -# {{2}} {{pfad/zu/datei}}
```

- Drucke eine Datei mit einem bestimmten Drucker:

```
lpr -P {{druckername}} {{pfad/zu/datei}}
```

- Drucke entweder eine einzelne Seite (z. B. 2) oder mehrere Seiten (z. B. 2-16):

```
lpr -o page-ranges={{2|2-16}} {{pfad/zu/datei}}
```

- Drucke doppelseitig entweder gespiegelt an der langen oder an der kurzen Seite:

```
lpr -o sides={{two-sided-long-edge|two-sided-short-edge}}  
{{pfad/zu/datei}}
```

- Drucke mit festgelegter Papiergröße (je nach Drucker-Konfiguration gibt es mehr Optionen):

```
lpr -o media={{a4|letter|legal}} {{pfad/zu/datei}}
```

- Drucke mehrere Seiten pro Blatt:

```
lpr -o number-up={{2|4|6|9|16}} {{pfad/zu/datei}}
```

# lpstat

Zeige Statusinformationen von Druckern.

Weitere Informationen: <https://manned.org/lpstat>.

- Liste alle aktuell verbundenen Drucker mit ihrer Druckverfügbarkeit auf:

```
lpstat -p
```

- Zeige den Standarddrucker an:

```
lpstat -d
```

- Zeige alle Statusinformationen an:

```
lpstat -t
```

- Liste alle Druckaufträge von einem gegebenen Nutzer auf:

```
lpstat -u {{nutzer}}
```

# ls

Liste den Inhalt eines Verzeichnisses auf.

Weitere Informationen: <https://www.gnu.org/software/coreutils/ls>.

- Liste den Inhalt in einer Datei pro Zeile auf:

```
ls -l
```

- Liste alle Dateien inklusive versteckter Dateien auf:

```
ls -a
```

- Liste alle Dateien mit einem abschließenden / bei Verzeichnis-Namen auf:

```
ls -F
```

- Liste alle Dateien mit Berechtigungen, Besitzer, Größe und Änderungsdatum auf:

```
ls -la
```

- Liste alle Dateien mit Dateigröße in für Menschen lesbaren Einheiten (KiB, MiB, GiB):

```
ls -lh
```

- Liste Dateien nach sortiert nach Dateigröße mit größter beginnend auf:

```
ls -ls
```

- Liste alle Dateien sortiert nach dem Änderungsdatum mit ältester beginnend auf:

```
ls -ltr
```

- Liste nur Verzeichnisse auf:

```
ls -d */
```

# lzcat

Dieser Befehl ist ein Alias von **xz**.

Weitere Informationen: <https://manned.org/lzcat>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr xz
```

# lzma

Dieser Befehl ist ein Alias von **xz**.

Weitere Informationen: <https://manned.org/lzma>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr xz
```

# magick compare

Zeige Unterschiede von zwei Bildern.

Weitere Informationen: <https://imagemagick.org/script/compare.php>.

- Vergleiche 2 Bilder:

```
magick compare {{pfad/zu/bild1.png}} {{pfad/zu/bild2.png}}  
{{pfad/zu/diff.png}}
```

- Vergleiche 2 Bilder mit einer bestimmten Metrik (standardmäßig NCC):

```
magick compare -verbose -metric {{PSNR}} {{pfad/zu/  
bild1.png}} {{pfad/zu/bild2.png}} {{pfad/zu/diff.png}}
```



# magick convert

ImageMagick Bildkonvertierungswerkzeug.

Weitere Informationen: <https://imagemagick.org/script/convert.php>.

- Konvertiere ein Bild von JPEG nach PNG:

```
magick convert {{pfad/zu/bild.jpg}} {{pfad/zu/bild.png}}
```

- Skaliere ein Bild auf 50% seiner Originalgröße:

```
magick convert {{pfad/zu/bild.png}} -resize 50% {{pfad/zu/bild2.png}}
```

- Skaliere ein Bild unter Beibehaltung des ursprünglichen Seitenverhältnisses auf eine maximale Größe von 640x480:

```
magick convert {{pfad/zu/bild.png}} -resize 640x480 {{pfad/zu/bild2.png}}
```

- Hänge Bilder horizontal aneinander:

```
magick convert {{pfad/zu/bild1.png pfad/zu/bild2.png ...}}  
+append {{pfad/zu/bild.png}}
```

- Hänge Bilder vertikal aneinander:

```
magick convert {{pfad/zu/bild1.png pfad/zu/bild2.png ...}} -  
append {{pfad/zu/bild.png}}
```

- Erstelle ein animiertes GIF aus einer Serie von Bildern mit einer Verzögerung von 100 ms zwischen den Bildern:

```
magick convert {{pfad/zu/bild1.png pfad/zu/bild2.png ...}} -  
delay {{10}} {{pfad/zu/animation.gif}}
```

- Erstelle ein Bild mit nichts als einem festen Hintergrund:

```
magick convert -size {{800x600}} "xc:{{#ff0000}}" {{pfad/zu/bild.png}}
```

- Erstelle ein Favicon aus mehreren Bildern verschiedener Größe:

```
magick convert {{pfad/zu/bild1.png pfad/zu/bild2.png ...}}  
{{pfad/zu/bild.ico}}
```

# meshnamed

Verteiltes Namenssystem für IPv6 Mesh-Netzwerke.

Weitere Informationen: <https://github.com/zhoreeq/meshname/>.

- Starte einen lokalen meshname DNS-Server:

```
meshnamed
```

- Wandle eine IPv6-Adresse in einen meshname um:

```
meshnamed -getname {{200:6fc8:9220:f400:5cc2:305a:4ac6:967e}}
```

- Wandle einen meshname in eine IPv6-Adresse um:

```
meshnamed -getip {{aiag7sesed2aaxgcgbnevruwpy}}
```

# minisign

Ein denkbar einfaches Werkzeug, um Dateien zu signieren und Signaturen zu verifizieren.

Weitere Informationen: <https://jedisct1.github.io/minisign/>.

- Generiere ein neues Schlüsselpaar im Standardpfad:

```
minisign -G
```

- Signiere eine Datei:

```
minisign -Sm {{pfad/zu/datei}}
```

- Signiere eine Datei und füge dabei einen vertrauenswürdigen (signierten) und einen nicht vertrauenswürdigen (unsignierten) Kommentar in der Signatur an:

```
minisign -Sm {{pfad/zu/datei}} -c "{{Nicht vertrauenswürdiger  
Kommentar}}" -t "{{Vertrauenswürdiger Kommentar}}"
```

- Verifiziere eine Datei und die vertrauenswürdigen Kommentare in ihrer Signatur gegen die angegebene Datei mit dem öffentlichen Schlüssel:

```
minisign -Vm {{pfad/zu/datei}} -p {{pfad/zu/  
öffentlicher_schlüssel.pub}}
```

- Verifiziere eine Datei und die vertrauenswürdigen Kommentare in ihrer Signatur gegen den angegebenen, in Base64 codierten öffentlichen Schlüssel:

```
minisign -Vm {{pfad/zu/datei}} -P  
"{{öffentlicher_schlüssel_base64}}"
```

# mount

Ermöglicht den Zugriff auf ein gesamtes Dateisystem in einem Verzeichnis.

Weitere Informationen: <https://manned.org/mount.8>.

- Zeige alle eingehängten Dateisysteme:

```
mount
```

- Hänge ein Gerät in ein Verzeichnis ein:

```
mount -t {{dateisystemtyp}} {{pfad/zu/gerätedatei}} {{pfad/zu/zielverzeichnis}}
```

- Hänge ein CD-ROM-Gerät (Dateisystemtyp ISO9660) in das Verzeichnis /cdrom schreibgeschützt ein:

```
mount -t {{iso9660}} -o ro {{/dev/cdrom}} {{/cdrom}}
```

- Hänge alle Dateisysteme ein, die in /etc/fstab definiert sind:

```
mount -a
```

- Hänge ein Dateisystem ein, das in /etc/fstab beschrieben ist (z. B. /dev/sda1 /meine\_platte ext2 defaults 0 2):

```
mount {{/meine_platte}}
```

- Hänge ein Verzeichnis in ein anderes Verzeichnis ein (danach sind die Inhalte über beide Pfade verfügbar):

```
mount --bind {{pfad/zu/altem_verzeichnis}} {{pfad/zu/neuem_verzeichnis}}
```

# mscore

Dieser Befehl ist ein Alias von **musescore**.

Weitere Informationen: <https://musescore.org/handbook/command-line-options>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr musescore
```

# musescore

MuseScore 3 Notenblatt-Editor.

Weitere Informationen: <https://musescore.org/en/handbook/3/command-line-options>.

- Verwende einen bestimmten Audio-Treiber:

```
musescore --audio-driver {{jack|alsa|portaudio|pulse}}
```

- Setze die MP3 Output-Bitrate in kbit/s:

```
musescore --bitrate {{bitrate}}
```

- Starte MuseScore im Debug-Modus:

```
musescore --debug
```

- Aktiviere experimentelle Funktionen, wie Layer:

```
musescore --experimental
```

- Exportiere eine Datei in ein anderes Format. Dieses hängt von der Dateierweiterung ab:

```
musescore --export-to {{output_datei}} {{input_datei}}
```

- Zeige Unterschiede zwischen zwei Partituren:

```
musescore --diff {{pfad/zu/datei1}} {{pfad/zu/datei2}}
```

- Gib eine MIDI-Importoperationsdatei an:

```
musescore --midi-operations {{pfad/zu/datei}}
```

# mv

Verschiebe Dateien oder Verzeichnisse oder benenne diese um.

Weitere Informationen: <https://www.gnu.org/software/coreutils/mv>.

- Verschiebe eine Datei an einen beliebigen Ort:

```
mv {{pfad/zu/datei}} {{pfad/zu/zieldatei}}
```

- Verschiebe mehrere Dateien in ein anderes Verzeichnis und behalte deren Namen bei:

```
mv {{pfad/zu/datei1 pfad/zu/datei2 ...}} {{pfad/zu/ziel_verzeichnis}}
```

- Überschreibe bereits existierende Dateien ohne vorherige Bestätigung:

```
mv -f {{pfad/zu/datei}} {{pfad/zu/zieldatei}}
```

- Überschreibe bereits existierende Dateien nach Bestätigung (unabhängig von Dateirechten):

```
mv -i {{pfad/zu/datei}} {{pfad/zu/zieldatei}}
```

- Verhindere das Überschreiben existierender Dateien am Zielort:

```
mv -n {{pfad/zu/datei}} {{pfad/zu/zieldatei}}
```

- Liste Dateien und deren Details auf während sie verschoben werden:

```
mv -v {{pfad/zu/datei}} {{pfad/zu/zieldatei}}
```

# nativefier

Befehlszeilen-Tool zum Erstellen einer Desktop-Anwendung für jede Website mit minimaler Konfiguration.

Weitere Informationen: <https://github.com/jiahaog/nativefier>.

- Erstelle einer Desktop-Anwendung für eine Website:

```
nativefier {{url}}
```

- Erstelle eine Desktop-Anwendung mit einem benutzerdefinierten Namen:

```
nativefier --name {{name}} {{url}}
```

- Verwende ein benutzerdefiniertes Icon:

```
nativefier --icon {{pfad/zu/icon.png}} {{url}}
```



# nginx

Nginx Webserver.

Weitere Informationen: <https://nginx.org/en/>.

- Starte den Server mit der standardmäßigen Konfigurationsdatei:

```
nginx
```

- Starte den Server mit einer benutzerdefinierten Konfigurationsdatei:

```
nginx -c {{konfigurationsdatei}}
```

- Starte den Server mit einem Präfix für alle relativen Pfade in der Konfigurationsdatei:

```
nginx -c {{konfigurationsdatei}} -p {{präfix/für/relative/pfade}}
```

- Teste die Konfigurationsdatei ohne den laufenden Server zu beeinflussen:

```
nginx -t
```

- Lade die Konfigurationsdatei durch das Senden eines Signales ohne Pause neu:

```
nginx -s reload
```

# ngrok

Reverse-Proxy, welcher einen sicheren Tunnel von einem öffentlichen Endpunkt zu einem lokal verfügbaren Webservice erstellt.

Weitere Informationen: <https://ngrok.com>.

- Veröffentliche einen lokalen HTTP-Service auf dem angegebenen Port:

```
ngrok http {{80}}
```

- Veröffentliche einen lokalen HTTP-Service auf einem bestimmten Host:

```
ngrok http {{beispiel.dev}}:{{80}}
```

- Veröffentliche einen lokalen HTTPS-Server:

```
ngrok http https://localhost
```

- Veröffentliche den TCP-Traffic auf dem angegebenen Port:

```
ngrok tcp {{22}}
```

- Veröffentliche den TLS-Traffic für einen bestimmten Host und Port:

```
ngrok tls -hostname={{beispiel.de}} {{443}}
```

# nix-build

Erstellen eines Nix-Ausdrucks.

Weitere Informationen: <https://nixos.org/manual/nix/stable/command-ref/nix-build.html>.

- Erstelle einen Nix-Ausdruck:

```
nix-build '<nixpkgs>' --attr {{firefox}}
```

- Erstelle einen gesandboxten Nix-Ausdruck (auf nicht-NixOS):

```
nix-build '<nixpkgs>' --attr {{firefox}} --option sandbox true
```

# nix-collect-garbage

Löschen von unbenutzten und unerreichbaren Nix-Speicherpfaden.

Generationen können mit **nix-env --list-generations** aufgelistet werden.

Weitere Informationen: <https://nixos.org/manual/nix/stable/command-ref/nix-collect-garbage.html>.

- Lösche alle Speicherpfade, die von den aktuellen Generationen der einzelnen Profile nicht verwendet werden:

```
sudo nix-collect-garbage --delete-old
```

- Simuliere die Löschung alter Speicherpfade:

```
sudo nix-collect-garbage --delete-old --dry-run
```

- Lösche alle Speicherpfade, die älter als 30 Tage sind:

```
sudo nix-collect-garbage --delete-older-than 30d
```

# nix-env

Manipulieren oder Abfragen von Nix-Benutzerumgebungen.

Weitere Informationen: <https://nixos.org/manual/nix/stable/#sec-nix-env>.

- Liste alle installierten Pakete auf:

```
nix-env -q
```

- Frage installierte Pakete ab:

```
nix-env -q {{suchbegriff}}
```

- Frage verfügbare Pakete ab:

```
nix-env -qa {{suchbegriff}}
```

- Installiere Paket:

```
nix-env -iA nixpkgs.{{paket_name}}
```

- Installiere ein Paket von einer URL:

```
nix-env -i {{paket_name}} --file {{beispiel.com}}
```

- Deinstalliere ein Paket:

```
nix-env -e {{paket_name}}
```

- Upgrade ein Pakets:

```
nix-env -u {{paket_name}}
```

- Upgrade alle Pakete:

```
nix-env -u
```

# nix-shell

Startet eine interaktive Shell, die auf einem Nix-Ausdruck basiert.

Weitere Informationen: <https://nixos.org/manual/nix/stable/command-ref/nix-shell.html>.

- Starte mit Nix-Ausdruck in `shell.nix` oder `default.nix` im aktuellen Verzeichnis:

```
nix-shell
```

- Führe Shell-Befehl in nicht-interaktiver Shell aus und beende:

```
nix-shell --run "{{befehl}}" {{argument1 argument2 ...}}"
```

- Starte mit Ausdruck in `default.nix` im aktuellen Verzeichnis:

```
nix-shell {{default.nix}}
```

- Starte mit aus nixpkgs geladenen Paketen:

```
nix-shell --packages {{paket_name_1 paket_name_2 ...}}
```

- Starte mit Paketen, die aus einer bestimmten Nixpkgs-Revision geladen wurden:

```
nix-shell --packages {{paket_name_1 paket_name_2 ...}} -I  
nixpkgs={{https://github.com/NixOS/nixpkgs/archive/  
nixpkgs_revision.tar.gz}}
```

- Führe den Rest der Datei mit einem bestimmten Interpreter aus, zur Verwendung in `#!-scripts` (siehe <https://nixos.org/manual/nix/stable/#use-as-a-interpret>):

```
nix-shell -i {{interpreter}} --packages {{paket_name_1  
paket_name_2 ...}}
```

# nix

Dienstprogramme für die Nix-Sprache und den Nix-Speicher.

Weitere Informationen: <https://nixos.org/manual/nix>.

- Suche nach einem Paket über seinen Namen oder seine Beschreibung:

```
nix search {{suchbegriff}}
```

- Starte eine Nix-Shell, die die angegebenen Pakete zur Verfügung stellt:

```
nix run {{nixpkgs.pkg1 nixpkgs.pkg2 ...}}
```

- Optimierte die Festplattennutzung des Nix-Speichers durch Zusammenfassen doppelter Dateien:

```
nix store optimise
```

- Starte eine interaktive Umgebung zum Ausführen von Nix-Ausdrücken:

```
nix repl
```

- Upgrade Nix auf die neueste stabile Version:

```
nix upgrade-nix
```

# nm-classic

Dieser Befehl ist ein Alias von **nm**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr nm
```



# nmap

Netzwerk-Erkundungs-Werkzeug und Security / Port Scanner.

Manche Funktionen können nur benutzt werden, wenn Nmap mit Root Rechten ausgeführt wird.

Weitere Informationen: <https://nmap.org/book/man.html>.

- Überprüfe ob eine IP-Adresse online ist und versuche, das Betriebssystem herauszufinden:

```
nmap -O {{ip_oder_hostname}}
```

- Überprüfe nur ob die angegebenen Hosts online sind (Ping Scan) und ihre Domain-Namen:

```
sudo nmap -sn {{ip_oder_hostname}}  
{{optional_noch_eine_adresse}}
```

- Scanne zusätzlich mit Skripten, Service-Erkennung, Betriebssystem-Fingerprinting und Traceroute:

```
nmap -A {{adresse_oder_adressen}}
```

- Scanne eine spezifische Liste an Ports (benutze '-p-' für alle Ports von 1 bis 65535):

```
nmap -p {{port1,port2,...,portN}} {{adresse_oder_adressen}}
```

- Führe Dienst- und Versions-Erkennung auf den top 1000 Ports mit den Standard NSE Skripten aus; und schreibe das Ergebnis ('-oN') in der Ausgabe Datei:

```
nmap -sC -sV -oN {{ergebnis.txt}} {{adresse_oder_adressen}}
```

- Scanne Ziel(e) vorsichtig mit 'default and safe' NSE Scripts:

```
nmap --script "default and safe" {{adresse_oder_adressen}}
```

- Scanne einen Web-Server, der auf den Standard Ports 80 und 443 läuft, mit allen verfügbaren 'http-\*' NSE Skripten:

```
nmap --script "http-*" {{adresse_oder_adressen}} -p 80,443
```

- Führe einen sehr langsamen verborgenen Scan ('-T0') aus um die Entdeckung von IDS/IPS zu umgehen und benutze Köder IP-Adressen ('-D'):

```
nmap -T0 -D  
{{köder1_ipadresse,köder2_ipadresse,...,köderN_ipadresse}}  
{{adresse_oder_addressen}}
```

# node

Server-seitige JavaScript Plattform (Node.js).

Weitere Informationen: <https://nodejs.org>.

- Führe eine JavaScript Datei aus:

```
node {{pfad/zu/datei}}
```

- Starte eine REPL (Interaktive Shell):

```
node
```

- Evaluiere als Argument übergebenen JavaScript Code:

```
node -e "{{code}}"
```

- Evaluierung und Ausgabe des Ergebnisses. Nützlich, um die Versionen der Abhängigkeiten von Node zu sehen:

```
node -p "{{process.versions}}"
```

- Aktiviere Inspector und pausiere die Ausführung bis sich ein Debugger verbindet sobald der Quellcode vollständig geparsed ist:

```
node --no-lazy --inspect-brk {{pfad/zu/datei}}
```

# nodemon

Beobachtet Dateien und startet Node Applikationen automatisch neu, wenn Änderungen erkannt wurden.

Weitere Informationen: <https://nodemon.io>.

- Führe die angegebene Datei aus und warte auf Änderungen:

```
nodemon {{pfad/zu/datei.js}}
```

- Manueller Neustart von Nodemon (beachte, dass Nodemon dabei aktiv sein muss):

```
rs
```

- Ignoriere bestimmte Dateien:

```
nodemon --ignore {{pfad/zu/datei_oder_verzeichnis}}
```

- Übergib Argumente an die Node Applikation:

```
nodemon {{pfad/zu/datei.js}} {{argumente}}
```

- Übergib Argumente an node selbst, wenn sie nicht bereits Argumente von nodemon sind (z.B. `--inspect`):

```
nodemon {{argumente}} {{pfad/zu/datei.js}}
```

- Führe Nicht-Node Skripte aus:

```
nodemon --exec "{{befehl_um_skript_auszufuehren}}  
{{argumente}}" {{pfad/zu/skript}}
```

- Führe ein Python-Skript aus:

```
nodemon --exec "python {{argumente}}" {{pfad/zu/datei.py}}
```

# nodenv

Ein Tool, um Node.js Versionen zu verwalten.

Weitere Informationen: <https://github.com/nodenv/nodenv>.

- Installiere eine bestimmte Node.js Version:

```
nodenv install {{version}}
```

- Zeige eine Liste von verfügbaren Versionen:

```
nodenv install --list
```

- Verwende systemweit eine bestimmte Node.js Version:

```
nodenv global {{version}}
```

- Verwende eine bestimmte Node.js Version im aktuellen Verzeichnis:

```
nodenv local {{version}}
```

- Zeige die verwendete Node.js Version im aktuellen Verzeichnis:

```
nodenv version
```

- Zeige den Ort, wo ein Node.js Befehl installiert ist (bspw. npm):

```
nodenv which {{befehl}}
```

# npm-check

Prüft veraltete, inkorrekte und nicht verwendete Abhängigkeiten.

Weitere Informationen: <https://github.com/dylang/npm-check>.

- Zeige einen Bericht von veralteten, inkorrekten und nicht verwendeten Abhängigkeiten:

```
npm-check
```

- Aktualisiere interaktiv veraltete Pakete:

```
npm-check --update
```

- Aktualisiere alle Pakete ohne Nachfrage:

```
npm-check --update-all
```

- Überspringe die Prüfung auf nicht verwendete Pakete:

```
npm-check --skip-unused
```

# npm-why

Identifiziert, warum ein npm Paket installiert ist.

Weitere Informationen: <https://github.com/amio/npm-why>.

- Zeige, warum ein npm Paket installiert ist:

```
npm-why {{paket_name}}
```

# npm

Ein Kommandozeilenwerkzeug für die Verwaltung von JavaScript und Node.js Paketen (Packages).

Weitere Informationen: <https://www.npmjs.com>.

- Erstelle eine `package.json` Datei interaktiv:

```
npm init
```

- Installiere alle in der `package.json` Datei gelisteten Abhängigkeiten:

```
npm install
```

- Installiere eine spezifische Version eines Packages und füge es automatisch der `package.json` Datei hinzu:

```
npm install {{package_name}}@{{version}}
```

- Installiere ein Package und füge es als Entwicklungs-Abhängigkeit der `package.json` Datei hinzu:

```
npm install {{package_name}} --save-dev
```

- Installiere ein Package global:

```
npm install --global {{package_name}}
```

- Deinstalliere ein Package und entferne es automatisch aus der `package.json` Datei:

```
npm uninstall {{package_name}}
```

- Gib eine Liste aller lokal installierten Packages aus:

```
npm list
```

- Gib eine Liste aller global installierten Packages aus:

```
npm list --global --depth={{0}}
```



# npx

Führt Binärdateien von **npm** Paketen aus.

Weitere Informationen: <https://github.com/npm/npx>.

- Führe die Binärdatei eines bestimmten npm Pakets aus:

```
npx {{modulname}} {{befehlsargumente}}
```

- Übergib den konkreten Namen, falls das Paket mehrere Binärdateien besitzt:

```
npx --package {{paketname}} {{modulname}}
```

- Führe einen Befehl aus, wenn er im aktuellen Verzeichnis oder in `node_modules/.bin` gefunden wird:

```
npx --no-install {{befehl}} {{befehlsargumente}}
```

- Führe die Binärdatei eines bestimmten npm Moduls aus und unterdrücke jede Ausgabe von `npx` selbst:

```
npx --quiet {{modulname}} {{befehlsargumente}}
```

- Zeige eine Hilfe an:

```
npx --help
```

# ntl

Dieser Befehl ist ein Alias von **netlify**.

Weitere Informationen: <https://cli.netlify.com>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr netlify
```

# nvm

Installiere, deinstalliere oder wechsele zwischen Node.js Versionen.

Unterstützt Versionsnummern wie "12.8" oder "v16.13.1", und Label wie "stable", "system", etc.

Weitere Informationen: <https://github.com/creationix/nvm>.

- Installiere eine bestimmte Node.js Version:

```
nvm install {{node_version}}
```

- Verwende eine bestimmte Node.js Version in der aktuellen Shell:

```
nvm use {{node_version}}
```

- Setze die Node.js-Standardversion:

```
nvm alias default {{node_version}}
```

- Zeige alle verfügbaren Node.js Versionen und hebe die Standardversion hervor:

```
nvm list
```

- Deinstalliere die angegebene Node.js Version:

```
nvm uninstall {{node_version}}
```

- Starte eine REPL mit einer bestimmten Node.js Version:

```
nvm run {{node_version}} --version
```

- Führe ein Skript mit einer bestimmten Node.js Version aus:

```
nvm exec {{node_version}} node {{app.js}}
```

# omz

Oh My Zsh Kommandozeilenwerkzeug.

Weitere Informationen: <https://github.com/ohmyzsh/ohmyzsh>.

- Aktualisiere Oh My Zsh:

```
omz update
```

- Zeige die Änderungen im neuesten Update von Oh My Zsh an:

```
omz changelog
```

- Starte die aktuelle Zsh-Sitzung und Oh My Zsh neu:

```
omz reload
```

- Zeige eine Liste aller verfügbaren Plugins an:

```
omz plugin list
```

- Aktiviere/Deaktiviere ein Oh My Zsh-Plugin:

```
omz plugin {{enable|disable}} {{plugin}}
```

- Zeige eine Liste aller verfügbaren Themen an:

```
omz theme list
```

- Wähle ein Oh My Zsh-Thema als Standardthema:

```
omz theme set {{theme}}
```

# pass

Programm zum Speichern und Lesen von Passwörtern und anderen sensiblen Daten.

Die Daten sind mit GPG verschlüsselt und werden mit einem Git repository verwaltet.

Weitere Informationen: <https://www.passwordstore.org>.

- Initialisiere oder verschlüssele einen neuen oder bestehenden Speicher mit einer oder mehreren GPG IDs neu:

```
pass init {{gpg_id_1}} {{gpg_id_2}}
```

- Speichere das Passwort und zusätzliche Informationen (Str + D auf neuer Zeile zum abschließen):

```
pass insert --multiline {{pfad/zu/datei}}
```

- Bearbeite einen bestimmten Eintrag:

```
pass edit {{pfad/zu/datei}}
```

- Kopiere das Passwort (die erste Zeile des Eintrags) in die Zwischenablage:

```
pass -c {{pfad/zu/datei}}
```

- Zeige die Baumstruktur des Passwort-Stores an:

```
pass
```

- Generiere ein neues, zufälliges Passwort mit Länge n und kopiere es in die Zwischenablage:

```
pass generate -c {{pfad/zu/datei}} {{n}}
```

- Initialisiere ein Git Repository (Alle durch pass durchgeführten Änderungen werden automatisch committed):

```
pass git init
```

- Führe einen Git-Befehl für den Passwort-Store aus:

```
pass git {{befehl}}
```

# pcapfix

Repariere beschädigte oder korruptierte **pcap**- und **pcapng**-Dateien.

Weitere Informationen: <https://f00l.de/pcapfix/>.

- Repariere eine pcap/pcapng-Datei (Hinweis: bei pcap-Dateien werden nur die ersten 262144 Bytes jedes Pakets gescannt):

```
pcapfix {{pfad/zu/datei.pcapng}}
```

- Repariere eine ganze pcap-Datei:

```
pcapfix --deep-scan {{pfad/zu/datei.pcap}}
```

- Repariere eine pcap/pcapng-Datei und schreibe die Reparierte Datei an einen bestimmten Speicherort:

```
pcapfix --outfile {{pfad/zu/Repariere.pcap}} {{pfad/zu/datei.pcap}}
```

- Behandle die zu reparierende Datei als pcapng-Datei, unabhängig von der automatischen Typenerkennung:

```
pcapfix --pcapng {{pfad/zu/datei.pcapng}}
```

- Repariere eine Datei und zeige den Reparaturprozess im Detail:

```
pcapfix --verbose {{pfad/zu/datei.pcap}}
```

# pdflatex

Kompiliere eine LaTeX Quelldatei in ein PDF Dokument.

Weitere Informationen: <https://manned.org/pdflatex>.

- Kompiliere ein PDF Dokument:

```
pdflatex {{quelldatei.tex}}
```

- Kompiliere ein PDF Dokument und gib ein bestimmtes Output-Verzeichnis an:

```
pdflatex -output-directory={{pfad/zu/verzeichnis}}  
{{quelldatei.tex}}
```

- Kompiliere ein PDF Dokument und stoppe bei jedem Fehler:

```
pdflatex -halt-on-error {{quelldatei.tex}}
```

# pdfseparate

Extrahiere die Seiten einer Portable Document Format (PDF) Datei.

Weitere Informationen: <https://manpages.debian.org/latest/poppler-utils/pdfseparate.1.en.html>.

- Extrahiere die Seiten einer PDF Datei und speichere jede Seite als neue PDF Datei ab:

```
pdfseparate {{pfad/zu/quelldatei.pdf}} {{pfad/zu/zieldatei-%d.pdf}}
```

- Gib die erste Seite zum Extrahieren an:

```
pdfseparate -f {{3}} {{pfad/zu/quelldatei.pdf}} {{pfad/zu/zieldatei-%d.pdf}}
```

- Gib die letzte Seite zum Extrahieren an:

```
pdfseparate -l {{10}} {{pfad/zu/quelldatei.pdf}} {{pfad/zu/zieldatei-%d.pdf}}
```



# pdftex

Kompiliere eine TeX Quelldatei in ein PDF Dokument.

Weitere Informationen: <https://www.tug.org/applications/pdftex/>.

- Kompiliere ein PDF Dokument:

```
pdftex {{quelldatei.tex}}
```

- Kompiliere ein PDF Dokument und gib ein bestimmtes Output-Verzeichnis an:

```
pdftex -output-directory={{pfad/zu/verzeichnis}}  
{{quelldatei.tex}}
```

- Kompiliere ein PDF Dokument und stoppe bei jedem Fehler:

```
pdftex -halt-on-error {{quelldatei.tex}}
```

# pdftotext

Konvertiere PDF Dateien zum plain text Format.

Weitere Informationen: <https://www.xpdfreader.com/pdftotext-man.html>.

- Konvertiere `datei.pdf` zu plain text und gib sie über die Standardausgabe aus:

```
pdftotext {{datei.pdf}} -
```

- Konvertiere `datei.pdf` zu plain text und speichere sie als `datei.txt`:

```
pdftotext {{datei.pdf}}
```

- Konvertiere `datei.pdf` zu plain text und erhalte das Layout:

```
pdftotext -layout {{datei.pdf}}
```

- Konvertiere `quelldatei.pdf` zu plain text und speichere sie als `zieldatei.txt`:

```
pdftotext {{quelldatei.pdf}} {{zieldatei.txt}}
```

- Konvertiere Seite 2, 3 und 4 von `quelldatei.pdf` zu plain text und speichere sie als `zieldatei.txt`:

```
pdftotext -f {{2}} -l {{4}} {{quelldatei.pdf}}  
{{zieldatei.txt}}
```

# pdfunite

Füge PDF Dateien zusammen.

Weitere Informationen: <https://github.com/mtgrosser/pdfunite>.

- Füge zwei PDF Dateien zu einer PDF Datei zusammen:

```
pdfunite {{pfad/zu/datei1.pdf}} {{pfad/zu/datei2.pdf}}  
{{pfad/zu/zieldatei.pdf}}
```

- Füge alle PDF Dateien in einem Verzeichnis zu einer PDF Datei zusammen:

```
pdfunite {{pfad/zu/verzeichnis/*.pdf}} {{pfad/zu/  
zieldatei.pdf}}
```

# peerindex

Liest MRT TABLE\_DUMPV2 Peer Index Table aus.

Kann mit gzip, bzip2 und xz komprimierte Dateien lesen.

Weitere Informationen: <https://codeberg.org/1414codeforge/ubgpsuite>.

- Gib alle Peers aus:

```
peerindex {{master6.mrt}}
```

- Zeige alle Peers an, die Routing-Informationen bereitgestellt haben:

```
peerindex -r {{master6.mrt}}
```

# php

PHP Befehlszeilenschnittstelle.

Weitere Informationen: <https://php.net>.

- Analysiere ein PHP-Skript und führe es aus:

```
php {{pfad/zu/datei}}
```

- Überprüfe die Syntax eines PHP-Skripts:

```
php -l {{pfad/zu/datei}}
```

- Führen PHP interaktiv aus:

```
php -a
```

- Führe einen PHP-Befehl aus:

```
php -r "{{befehl}}"
```

- Starte den in PHP integrierten Webserver im aktuellen Verzeichnis:

```
php -S {{host:port}}
```

- Zeige eine Liste der installierten PHP-Erweiterungen:

```
php -m
```

- Zeige Informationen zur aktuellen PHP-Konfiguration an:

```
php -i
```

- Zeige Informationen zu einer bestimmten Funktion an:

```
php --rf {{funktionsname}}
```

# phpbu

Ein Backup framework für PHP.

Weitere Informationen: <https://phpbu.de>.

- Führe ein Backup mit der Standard `phpbu.xml` Konfigurationsdatei aus:

```
phpbu
```

- Führe ein Backup mit einer bestimmten Konfigurationsdatei aus:

```
phpbu --configuration={{pfad/zu/konfiguration.xml}}
```

- Führe nur die angegebenen Backups aus:

```
phpbu --limit={{backup_art}}
```

- Simuliere Aktionen die ausgeführt werden würden:

```
phpbu --simulate
```

# ping

Sendet ICMP ECHO\_REQUEST Pakete an andere Geräte im Netzwerk.

Weitere Informationen: <https://manned.org/ping>.

- Sende Pings an ein Gerät im Netzwerk:

```
ping {{ziel}}
```

- Sende nur eine bestimmte Anzahl an Pings:

```
ping -c {{anzahl}} {{ziel}}
```

- Sende Pings und bestimme das Interval in Sekunden zwischen diesen (standardmäßig ist es eine Sekunde):

```
ping -i {{sekunden}} {{ziel}}
```

- Sende Pings ohne symbolische Namen nach Adressen aufzulösen:

```
ping -n {{ziel}}
```

- Sende Pings und signalisiere eine erfolgreiche Antwort durch ein Bell Signal (wenn das Terminal es unterstützt):

```
ping -a {{ziel}}
```

- Zeige auch eine Nachricht, wenn keine Antwort empfangen wurde:

```
ping -0 {{ziel}}
```

# ping6

Sendet ICMP ECHO\_REQUEST Pakete an andere Geräte im Netzwerk, welche mittels IPv6-Adressen identifiziert werden.

Weitere Informationen: <https://manned.org/ping6>.

- Sende Pings an ein Gerät im Netzwerk:

```
ping6 {{ziel}}
```

- Sende nur eine bestimmte Anzahl an Pings:

```
ping6 -c {{anzahl}} {{ziel}}
```

- Sende Pings und bestimme das Interval in Sekunden zwischen diesen (standardmäßig ist es eine Sekunde):

```
ping6 -i {{sekunden}} {{ziel}}
```

- Sende Pings ohne symbolische Namen nach Adressen aufzulösen:

```
ping6 -n {{ziel}}
```

- Sende Pings und signalisiere eine erfolgreiche Antwort durch ein Bell Signal (wenn das Terminal es unterstützt):

```
ping6 -a {{ziel}}
```



# pio init

Dieser Befehl ist ein Alias von **pio project**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr pio project
```

# piodebuggdb

Dieser Befehl ist ein Alias von **pio debug**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr pio debug
```

# pip install

Installiere Python-Pakete.

Weitere Informationen: <https://pip.pypa.io>.

- Installiere ein Paket:

```
pip install {{paketname}}
```

- Installiere eine spezifische Paketversion:

```
pip install {{paketname}}=={{paketversion}}
```

- Installiere die Pakete aus einer Datei:

```
pip install -r {{requirements.txt}}
```

- Installiere die Pakete von einer URL oder einem lokalen Archiv (.tar.gz | .whl):

```
pip install --find-links {{url|pfad/zur/datei}}
```

- Installiere das lokale Paket im aktuellen Verzeichnis im Entwicklungs-/ Bearbeitungsmodus:

```
pip install --editable {{.}}
```

# pip

Python package manager.

Einige Unterbefehle wie **pip install** sind separat dokumentiert.

Weitere Informationen: <https://pip.pypa.io>.

- Installiere ein Paket (siehe `pip install` für weitere Beispiele):

```
pip install {{paketname}}
```

- Installiere ein Paket im Benutzerverzeichnis, anstatt systemweit:

```
pip install --user {{paketname}}
```

- Aktualisiere ein Paket:

```
pip install --upgrade {{paketname}}
```

- Deinstalliere ein Paket:

```
pip uninstall {{paketname}}
```

- Speichere eine Liste aller installierten Pakete in eine Datei:

```
pip freeze > {{requirements.txt}}
```

- Zeige Informationen über ein installiertes Paket an:

```
pip show {{paketname}}
```

- Installiere Pakete, die in einer Datei gelistet sind:

```
pip install --requirement {{requirements.txt}}
```

# plantuml

Erstelle UML-Diagramme aus einer reinen Textsprache und rendere sie in verschiedenen Formaten.

Weitere Informationen: <https://plantuml.com/en/command-line>.

- Rendere Diagramme im Standardformat (PNG):

```
plantuml {{pfad/zu/diagramm1.puml}} {{pfad/zu/diagramm2.puml}}
```

- Rendere ein Diagramm im vorgegebenen Format (z.B. png, pdf, svg, txt):

```
plantuml -t {{format}} {{pfad/zu/diagramm.puml}}
```

- Rendere alle Diagramme eines Verzeichnisses:

```
plantuml {{pfad/zu/verzeichnis}}
```

- Rendere ein Diagramm in ein bestimmtes Ausgabeverzeichnis:

```
plantuml -o {{pfad/zu/verzeichnis}} {{pfad/zu/diagramm.puml}}
```

- Rendere ein Diagramm mit einer bestimmten Konfigurationsdatei:

```
plantuml -config {{pfad/zu/konfig.cfg}} {{pfad/zu/diagramm.puml}}
```

- Zeige Hilfe an:

```
plantuml -help
```

# platformio

Dieser Befehl ist ein Alias von **pio**.

Weitere Informationen: <https://docs.platformio.org/en/latest/core/userguide/>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr pio
```

# pnpm

Schneller, speicherplatzsparender Paketmanager für Node.js.

Ein Kommandozeilenwerkzeug für die Verwaltung von JavaScript und Node.js Paketen (Packages).

Weitere Informationen: <https://pnpm.io>.

- Erstelle eine `package.json` Datei interaktiv:

```
pnpm init
```

- Installiere alle in der `package.json` Datei gelisteten Abhängigkeiten:

```
pnpm install
```

- Installiere eine spezifische Version eines Packages und füge es automatisch der `package.json` Datei hinzu:

```
pnpm add {{modul_name}}@{{version}}
```

- Installiere ein Package und füge es als Entwicklungs-Abhängigkeit der `package.json` Datei hinzu:

```
pnpm add -D {{modul_name}}
```

- Installiere ein Package global:

```
pnpm add -g {{modul_name}}
```

- Deinstalliere ein Package und entferne es automatisch aus der `package.json` Datei:

```
pnpm remove {{modul_name}}
```

- Gib eine Liste aller lokal installierten Packages aus:

```
pnpm list
```

- Gib eine Liste aller global installierten Packages aus:

```
pnpm list -g --depth={{0}}
```

# poetry

Verwalte Python-Pakete und -Abhängigkeiten.

Weitere Informationen: <https://python-poetry.org/docs/cli/>.

- Erstelle ein neues Poetry-Projekt im Verzeichnis mit dem angegebenen Namen:

```
poetry new {{projekt_name}}
```

- Installiere eine Abhängigkeit und alle Unterabhängigkeiten:

```
poetry add {{abhängigkeit}}
```

- Installiere eine Entwicklungsabhängigkeit und alle Unterabhängigkeiten:

```
poetry add --dev {{abhängigkeit}}
```

- Initialisiere ein neues Poetry-Projekt interaktiv im aktuellen Verzeichnis:

```
poetry init
```

- Aktualisiere alle Abhängigkeiten und `poetry.lock`:

```
poetry update
```

- Führe einen Befehl innerhalb der virtuellen Umgebung des Projekts aus:

```
poetry run {{befehl}}
```

- Erhöhe die Minor-Version des Projekts in `pyproject.toml`:

```
poetry version minor
```

- Liste alle poetry Unterbefehle auf:

```
poetry list
```



# ptpython3

Dieser Befehl ist ein Alias von **ptpython**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr ptpython
```

# puppet agent

Ruft die Client-Konfiguration eines Puppetservers ab und setzt diese auf dem System um.

Weitere Informationen: <https://puppet.com/docs/puppet/7/man/agent.html>.

- Registriere die Node bei einem Puppetserver und wende den empfangenen Katalog an:

```
puppet agent --test --server {{puppetserver_fqdn}} --  
serverport {{port}} --waitforcert {{poll_zeit}}
```

- Führe den Agenten im Hintergrund aus (nutzt die Einstellungen von `/opt/puppetlabs/puppet/puppet.conf`):

```
puppet agent
```

- Führe den Agenten einmal im Vordergrund aus und beende:

```
puppet agent --test
```

- Führe den Agenten im Dry-Modus aus:

```
puppet agent --test --noop
```

- Protokolliere jede ausgewertete Ressource (selbst wenn sich nichts geändert hat):

```
puppet agent --test --evaltrace
```

- Deaktiviere den Agenten:

```
puppet agent --disable "{{nachricht}}"
```

- Aktiviere den Agenten:

```
puppet agent --enable
```

# puppet apply

Wende ein Puppet-Manifest lokal an.

Weitere Informationen: <https://puppet.com/docs/puppet/7/man/apply.html>.

- Wende ein Manifest an:

```
puppet apply {{pfad/zu/manifest}}
```

- Führe Puppetcode aus:

```
puppet apply --execute {{code}}
```

- Benutze ein bestimmtes Modulverzeichnis und Hiera-Konfigurationsdatei:

```
puppet apply --modulepath {{pfad/zu/ordner}} --hiera_config  
{{pfad/zu/datei}} {{pfad/zu/manifest}}
```

# puppet

Puppet bietet Tools, an um die Konfiguration von Servern zu automatisieren.

Einige Unterbefehle wie **puppet agent** sind separat dokumentiert.

Weitere Informationen: <https://puppet.com/>.

- Führe einen Puppet-Unterbefehl aus:

```
puppet {{unterbefehl}}
```

- Gib die installierte Version von Puppet aus:

```
puppet --version
```

- Zeige die generelle Hilfeseite an:

```
puppet --help
```

- Zeige die Hilfeseite eines Unterbefehls an (z.B. **agent**, **apply**, etc.):

```
puppet help {{unterbefehl}}
```

# pystun3

Classic STUN-Client, geschrieben in Python.

Weitere Informationen: <https://github.com/talkiq/pystun3>.

- Stelle eine STUN-Anfrage:

```
pystun3
```

- Stelle eine STUN-Anfrage und spezifiziere den STUN-Server:

```
pystun3 --stun-host {{stun.lund1.de}}
```

- Stelle eine STUN-Anfrage und spezifiziere den Quellport:

```
pystun3 --source-port {{7932}}
```

# python3

Dieser Befehl ist ein Alias von **python**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr python
```

# r2

Dieser Befehl ist ein Alias von **radare2**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr radare2
```

# rails

Ein Server-seitiges MVC Webframework in Ruby.

Manche Unterbefehle wie **rails generate** sind separat dokumentiert.

Weitere Informationen: [https://guides.rubyonrails.org/command\\_line.html](https://guides.rubyonrails.org/command_line.html).

- Erstelle ein neues Rails-Projekt:

```
rails new "{{projektname}}"
```

- Starte einen lokalen Server für das aktuelle Project unter Port 3000:

```
rails server
```

- Starte einen lokalen Server für das aktuelle Project unter einem bestimmten Port:

```
rails server -p "{{port}}"
```

- Starte eine interactive Konsole (REPL):

```
rails console
```

- Zeige die Versionsnummer von rails an:

```
rails --version
```



# rcat

Dieser Befehl ist ein Alias von **rc**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr rc
```

# rev

Kehre die Reihenfolge von Text um.

Weitere Informationen: <https://manned.org/rev>.

- Kehre die Reihenfolge des Textes "Hallo" um:

```
echo "Hallo" | rev
```

- Kehre die Reihenfolge einer Datei um:

```
rev {{pfad/zu/datei}}
```

# ripgrep

Dieser Befehl ist ein Alias von **rg**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr rg
```

# rm

Lösche Dateien oder Verzeichnisse.

Weitere Informationen: <https://www.gnu.org/software/coreutils/rm>.

- Lösche Dateien an beliebigen Speicherorten:

```
rm {{pfad/zu/datei1 pfad/zu/datei2 ...}}
```

- Lösche mehrere Dateien mit Eingabeaufforderung zur Bestätigung für jede Datei:

```
rm -i {{pfad/zu/datei1 pfad/zu/datei2 ...}}
```

- Liste jede Datei auf, wenn sie gelöscht wird:

```
rm -v {{pfad/zu/verzeichnis/*}}
```

- Lösche ein Verzeichnis und alle seine Unterverzeichnisse rekursiv:

```
rm -r {{pfad/zu/verzeichnis}}
```

# sc\_analysis\_dump

Ausgabe von Traceroute-Pfaden in einem leicht zu parsenden Format.

Weitere Informationen: <https://www.caida.org/catalog/software/scamper/>.

- Gib die traceroute in `warts`-Dateien nacheinander in einem leicht zu parsendem Format aus:

```
sc_analysis_dump {{path/to/file1.warts path/to/file2.warts  
...}}
```

# sc\_tracediff

Anzeige von Traceroute-Pfaden, bei denen sich der Pfad geändert hat.

Weitere Informationen: <https://www.caida.org/catalog/software/scamper/>.

- Zeige den Unterschied zwischen den traceroutes in zwei warts-Dateien:

```
sc_tracediff {{path/to/file1.warts}} {{path/to/file2.warts}}
```

- Zeige den Unterschied zwischen den traceroutes in zwei warts-Dateien, einschließlich derer, die sich nicht geändert haben:

```
sc_tracediff -a {{path/to/file1.warts}} {{path/to/
file2.warts}}
```

- Zeige den Unterschied zwischen den traceroutes in zwei `warts'-Dateien und versuche, wenn möglich, DNS-Namen und nicht IP-Adressen anzuzeigen:

```
sc_tracediff -n {{path/to/file1.warts}} {{path/to/
file2.warts}}
```

# sc\_ttlexp

Ausgabe der Quelladressen von ICMP TTL expire-Nachrichten in **warts**-Dateien.

Weitere Informationen: <https://www.caida.org/catalog/software/scamper/>.

- Gib die Quelladressen von ICMP TTL expire-Nachrichten in einer warts-Datei nacheinander aus:

```
sc_ttlexp {{path/to/file1.warts path/to/file2.warts ...}}
```

# sc\_warts2csv

Umwandlung von traceroutes aus **warts**-Dateien in das CSV-Format.

Weitere Informationen: <https://www.caida.org/catalog/software/scamper/>.

- Wandle traceroute-Daten in einer warts-Datei in CSV um und gebe dieses aus:

```
sc_warts2csv {{path/to/file1.warts path/to/file2.warts ...}}
```



# sc\_warts2json

Wandelt eine WARTS-Datei in eine JSON-Datei um.

Weitere Informationen: <https://www.caida.org/catalog/software/scamper/>.

- Wandle `warts`-Dateien in JSON um und gib diese aus:

```
sc_warts2json {{path/to/file1.warts path/to/file2.warts ...}}
```

# sc\_warts2pcap

Schreibt die in den **warts**-Dateien enthaltenen Pakete in eine **pcap**-Datei.

Dies ist nur bei tbit, sting und sniff möglich.

Weitere Informationen: <https://www.caida.org/catalog/software/scamper/>.

- Wandle die Daten aus mehreren warts-Dateien in eine pcap-Datei um:

```
sc_warts2pcap -o {{path/to/output.pcap}} {{path/to/  
file1.warts path/to/file2.warts ...}}
```

- Wandle die Daten aus einer warts-Datei in eine pcap-Datei um und sortiere die Pakete nach Zeitstempel:

```
sc_warts2pcap -s -o {{path/to/output.pcap}} {{path/to/  
file.warts}}
```

# sc\_warts2text

Einfache Ausgabe der in einer **warts**-Datei enthaltenen Informationen.

Weitere Informationen: <https://www.caida.org/catalog/software/scamper/>.

- Gib die Informationen in einer warts-Datei als Text aus:

```
sc_warts2text {{path/to/file1.warts path/to/file2.warts ...}}
```

# sc\_wartscat

Füge mehrere **warts**-Dateien zusammen.

Weitere Informationen: <https://www.caida.org/catalog/software/scamper/>.

- Verkette mehrere warts-Dateien zu Einer:

```
sc_wartscat -o {{path/to/output.warts}} {{path/to/file1.warts  
path/to/file2.warts ...}}
```

# sc\_wartsdump

Ausführliche Ausgabe der in einer **warts**-Datei enthaltenen Daten.

Weitere Informationen: <https://www.caida.org/catalog/software/scamper/>.

- Gib den Inhalt von warts-Dateien ausführlich aus:

```
sc_wartsdump {{path/to/file1.warts path/to/file2.warts ...}}
```

# sc\_wartsfilter

Filtert bestimmte Datensätze aus einer **warts**-Datei.

Weitere Informationen: <https://www.caida.org/catalog/software/scamper/>.

- Filtere alle Datensätze, welche ein bestimmtes Ziel haben und schreibe sie in eine separate Datei:

```
sc_wartsfilter -i {{path/to/input.warts}} -o {{path/to/output.warts}} -a {{192.0.2.5}} -a {{192.0.2.6}}
```

- Filtere alle Datensätze, welche ein Ziel in einem bestimmten Prefix haben und schreibe sie in eine separate Datei:

```
sc_wartsfilter -i {{path/to/input.warts}} -o {{path/to/output.warts}} -a {{2001:db8::/32}}
```

- Filtere alle Datensätze, welche durch eine bestimmte Aktion entstanden sind und gebe sie als JSON aus:

```
sc_wartsfilter -i {{path/to/input.warts}} -t {{ping}} |  
sc_warts2json
```

# sc\_wartsfix

Rettet beschädigte **warts**-Dateien.

Weitere Informationen: <https://www.caida.org/catalog/software/scamper/>.

- Speichere alle Datensätze (in einer separaten Datei) bis zum letzten intakten Datensatz:

```
sc_wartsfix {{path/to/file1.warts path/to/file2.warts ...}}
```

# scamper

Sondiert aktiv das Internet, um die Topologie und Leistung zu analysieren.

Liefert einige Werkzeuge mit, welche mit **sc\_** starten, beispielsweise **sc\_warts2text** oder **sc\_ttlexp**.

Weitere Informationen: <https://www.caida.org/catalog/software/scamper/>.

- Führe die Standardoption (Traceroute) auf ein Ziel aus:

```
scamper -i {{192.0.2.1}}
```

- Führe zwei Aktionen (ping und traceroute) auf zwei verschiedenen Zielen aus:

```
scamper -I "{{ping}} {{192.0.2.1}}" -I "{{trace}}  
{{192.0.2.2}}
```

- Pinge mehrere Hosts mit UDP an, verwende eine bestimmte Portnummer für den ersten Ping und erhöhe sie für jeden weiteren Ping:

```
scamper -c "{{ping}} -P {{UDP-dport}} -d {{33434}}" -i  
{{192.0.2.1}} -i {{192.0.2.2}}
```

- Verwende den Multipath Discovery Algorithm (MDA), um das Vorhandensein von lastverteilten Pfaden zum Ziel zu ermitteln, und verwende für die Sondierung ICMP-Echopakete mit maximal drei Versuchen, und schreibe das Ergebnis in eine warts-Datei:

```
scamper -O {{warts}} -o {{path/to/output.warts}} -I  
"{{tracelb}} -P {{ICMP-echo}} -q {{3}} {{192.0.2.1}}"
```

- Führe eine Paris-Traceroute mit ICMP zu einem Ziel aus und speichere das Ergebnis in einer komprimierten warts-Datei:

```
scamper -O {{warts.gz}} -o {{path/to/output.warts}} -I  
"{{trace}} -P {{icmp-paris}} {{2001:db8:dead:beaf::4}}"
```

- Zeichne alle ICMP-Pakete, die an einer bestimmten IP-Adresse ankommen und eine bestimmte ICMP-ID haben, in einer warts-Datei auf:

```
scamper -O {{warts}} -o {{path/to/output.warts}} -I "sniff -S  
{{2001:db8:dead:beef::6}} icmp[icmpid] == {{101}}"
```



# sl

Dampflokomotive welche durch das Terminal fährt.

Weitere Informationen: <https://github.com/mtoyoda/sl>.

- Lasse eine Dampflokomotive durch dein Terminal fahren:

```
sl
```

- Der Zug brennt, Menschen schreien:

```
sl -a
```

- Lasse den Zug fliegen:

```
sl -F
```

- Zeige den Zug kleiner an:

```
sl -l
```

- Lasse den Benutzer abbrechen (STRG + C):

```
sl -e
```

# sleep

Verzögert für einen bestimmten Zeitraum.

Weitere Informationen: <https://pubs.opengroup.org/onlinepubs/9699919799/utilities/sleep.html>.

- Verzögere in Sekunden:

```
sleep {{sekunden}}
```

- Verzögere in Minuten:

```
sleep {{minuten}}m
```

- Verzögere in Stunden:

```
sleep {{stunden}}h
```

# sqlite3

Das Kommandozeileninterface für SQLite 3, welches eine eigenständige dateibasierte eingebettete SQL-Engine ist.

Weitere Informationen: <https://sqlite.org>.

- Starte eine interaktive Shell mit einer neuen Datenbank:

```
sqlite3
```

- Öffne eine interaktive Shell mit einer existierenden Datenbank:

```
sqlite3 {{pfad/zu/datenbank.sqlite3}}
```

- Führe ein SQL Statement auf einer existierenden Datenbank aus und beende die Ausführung danach:

```
sqlite3 {{pfad/zu/datenbank.sqlite3}} '{{SELECT * FROM  
einer_tabelle;}}'
```

# ssh-agent

Erstelle einen SSH Agenten-Prozess.

Ein SSH Agent behält die hinzugefügten SSH Schlüssel solange verschlüsselt im Arbeitsspeicher, bis diese entfernt werden oder der Agenten-Prozess beendet wird.

Siehe auch **ssh-add**, um Schlüssel zu verwalten.

Weitere Informationen: <https://man.openbsd.org/ssh-agent>.

- Starte einen SSH Agenten-Prozesses für die aktuelle Shell:

```
eval $(ssh-agent)
```

- Beende den aktuell laufenden SSH Agenten-Prozesses:

```
ssh-agent -k
```

# ssh-copy-id

Installiere den öffentlichen Teil eines SSH Schlüssels in der **authorized\_keys** Datei auf einem externen Server.

Weitere Informationen: <https://manned.org/ssh-copy-id>.

- Kopiere den eigenen öffentlichen SSH Schlüssels zu einem externen Server:

```
ssh-copy-id {{benutzer}}@{{externer_server}}
```

- Kopiere den angegebenen öffentlichen SSH Schlüssels zu einem externen Server:

```
ssh-copy-id -i {{pfad/zu/öffentlichem_schlüssel}}  
{{benutzer}}@{{externer_server}}
```

- Kopiere den angegeben öffentlichen SSH Schlüssels zu einem externen Server unter Angabe eines bestimmten SSH Ports:

```
ssh-copy-id -i {{pfad/zu/öffentlichem_schlüssel}} -p {{port}}  
{{benutzer}}@{{externer_server}}
```

# ssh-keygen

Generiert SSH Schlüssel für Authentifizierung, Passwort-lose Logins und mehr.

Weitere Informationen: <https://man.openbsd.org/ssh-keygen>.

- Erstelle ein SSH Schlüssel-Paar interaktiv:

```
ssh-keygen
```

- Generiere ein ed25519 Schlüssel-Paar mit 32 Schlüssel-Ableitungs-Iterationen und speicher unter einem bestimmten Dateinamen:

```
ssh-keygen -t {{ed25519}} -a {{32}} -f {{~/.ssh/datei}}
```

- Generiere ein 4096 Bit langen RSA Schlüssel-Paar mit der E-Mail im Kommentarfeld:

```
ssh-keygen -t {{rsa}} -b {{4096}} -C "{{kommentar|e-mail}}"
```

- Entferne den Schlüssel eines Servers aus der `known_hosts` Datei (hilfreich wenn ein Server seinen Schlüssel aktualisiert hat und der alte somit nicht mehr gilt):

```
ssh-keygen -R {{externer_server}}
```

- Rufe den Fingerabdruck eines Schlüssels im MD5 Hex Format ab:

```
ssh-keygen -l -E {{md5}} -f {{~/.ssh/datei}}
```

- Ändere das Passwort eines privaten Schlüssels:

```
ssh-keygen -p -f {{~/.ssh/datei}}
```

- Ändern Sie den Typ des Schlüsselformats (z. B. vom OPENSSH-Format in PEM), die Datei wird an Ort und Stelle neu geschrieben:

```
ssh-keygen -p -N "" -m {{PEM}} -f ~/.ssh/{{datei}}
```

# ssh-keyscan

Rufe öffentliche SSH Schlüssel eines externen Servers ab.

Weitere Informationen: <https://man.openbsd.org/ssh-keyscan>.

- Rufe alle öffentlichen SSH Schlüssel eines Servers ab:

```
ssh-keyscan {{server}}
```

- Rufe alle öffentlichen SSH Schlüssel unter einem bestimmten Port ab:

```
ssh-keyscan -p {{port}} {{server}}
```

- Rufe bestimmte SSH Schlüssel-Typen ab:

```
ssh-keyscan -t {{rsa,dsa,ecdsa,ed25519}} {{server}}
```

- Aktualisiere die `known_hosts` SSH Datei mit dem Fingerabdruck eines bestimmten Servers:

```
ssh-keyscan -H {{server}} >> ~/.ssh/known_hosts
```

# ssh

Secure Shell ist ein Protokoll für das sichere einloggen auf einem externen System.

Es kann dafür eingesetzt werden um Befehle auf externen Systemen auszuführen.

Weitere Informationen: <https://man.openbsd.org/ssh>.

- Stelle eine Verbindung zu einem externen Server her:

```
ssh {{benutzer}}@{{externer_server}}
```

- Stelle eine Verbindung zu einem externen Server mit spezifischer Identität her (privater SSH Schlüssel):

```
ssh -i {{pfad/zu/schlüssel_datei}} {{benutzer}}  
@{{externer_server}}
```

- Stelle eine Verbindung zu einem externen Server unter einem spezifischen Port her:

```
ssh {{benutzer}}@{{externer_server}} -p {{2222}}
```

- SSH Tunneln: Leite Ports dynamische Port weiter (SOCKS proxy auf localhost: 1080):

```
ssh -D {{1080}} {{benutzer}}@{{externer_server}}
```

- SSH Tunneln: Leite einen spezifischen Ports (localhost:9999 zu example.org: 80) weiter zusammen mit deaktivierter pseudy-tty Provisionierung für die Ausführung eines Befehls:

```
ssh -L {{9999}}:{{example.org}}:{{80}} -N -T {{benutzer}}  
@{{externer_server}}
```

- SSH Springen: Verbinde über einen Spring-Server zu einem externen Server (Es können auch mehrere Spring-Server über eine Komma-separierte Liste angegeben werden):

```
ssh -J {{benutzer}}@{{spring_server}} {{benutzer}}  
@{{externer_server}}
```

- Agenten Weiterleitung: Leite die eigenen Authentifizierungs-Informationen an den externen Server weiter (siehe man `ssh_config` für mehr Optionen):

```
ssh -A {{benutzer}}@{{externer_server}}
```



# sshfs

Dateisystem Client für SSH.

Weitere Informationen: <https://github.com/libfuse/sshfs>.

- Hänge ein externes Verzeichnis ein:

```
sshfs {{benutzer}}@{{externer_server}}:  
{{externes_verzeichnis}} {{lokales_einhänge_verzeichnis}}
```

- Hänge ein externes Verzeichnis aus:

```
umount {{lokaler_einhänge_verzeichnis}}
```

- Hänge ein externes Verzeichnis unter einem bestimmten Port ein:

```
sshfs {{benutzer}}@{{externer_server}}:  
{{externes_verzeichnis}} -p {{2222}}
```

- Verwende Komprimierung:

```
sshfs {{benutzer}}@{{externer_server}}:  
{{externes_verzeichnis}} -C
```

- Folge symbolischen Links:

```
sshfs -o follow_symlinks {{benutzer}}@{{externer_server}}:  
{{externes_verzeichnis}} {{lokaler_einhänge_verzeichnis}}
```

# sshpass

Stelle SSH Passwörtern bereit.

Weitere Informationen: <https://manned.org/sshpass>.

- Stelle eine Verbindung zu einem externen Server über ein Passwort aus einem Datei-Objekt her (in diesem Fall `stdin`):

```
sshpass -d {{0}} ssh {{benutzer}}@{{server}}
```

- Stelle eine Verbindung zu einem externen Server mit Hilfe eines Passworts bei automatischer Akzeptierung von unbekannten SSH Schlüsseln her:

```
sshpass -p {{passwort}} ssh -o StrictHostKeyChecking=no  
{{benutzer}}@{{server}}
```

- Stelle eine Verbindung zu einem externen Server mit Hilfe eines Passworts aus der ersten Zeile einer Datei bei automatischer Akzeptierung von unbekannten SSH Schlüsseln mit anschließender Ausführung eines Befehls her:

```
sshpass -f {{pfad/zu/datei}} ssh -o StrictHostKeyChecking=no  
{{benutzer}}@{{server}} "{{befehl}}"
```

# steam

Eine Plattform für Videospiele von Valve.

Weitere Informationen: [https://developer.valvesoftware.com/wiki/Command Line Options](https://developer.valvesoftware.com/wiki/Command_Line_Options).

- Starte Steam und gebe Debug-Nachrichten auf die Standardausgabe aus:

```
steam
```

- Starte Steam und aktiviere die eingebaute Debug-Menüoption:

```
steam -console
```

- Aktiviere die Menüoption für die Steam-Konsole und öffne diese in einer aktiven Steam-Instanz:

```
steam steam://open/console
```

- Logge dich in Steam mit den angegebenen Zugangsdaten ein:

```
steam -login {{benutzername}} {{passwort}}
```

- Starte Steam im Big-Picture-Modus:

```
steam -tenfoot
```

- Stoppe Steam:

```
steam -shutdown
```

# steamcmd

Ein Kommandozeilenwerkzeug, um über Steam verfügbare Anwendungen zu verwalten.

Weitere Informationen: <https://manned.org/steamcmd>.

- Installiere und aktualisiere eine Anwendung ohne dich einzuloggen:

```
steamcmd +login {{anonymous}} +app_update {{anwendungs_id}}  
+quit
```

- Installiere oder aktualisiere eine Anwendung unter Angabe deiner Zugangsdaten:

```
steamcmd +login {{benutzername}} +app_update  
{{anwendungs_id}} +quit
```

- Installiere eine Anwendung für eine bestimmte Plattform:

```
steamcmd +@sSteamCmdForcePlatformType {{windows}} +login  
{{benutzername}} +app_update {{anwendungs_id}} validate +quit
```

# stun

Classic STUN-Client.

Weitere Informationen: <https://manned.org/stun.1>.

- Stelle eine STUN-Anfrage:

```
stun {{stun.1und1.de}}
```

- Stelle eine STUN-Anfrage und spezifiziere den Quellport:

```
stun {{stun.1und1.de}} -p {{4302}}
```

# tail

Gib das Ende einer Datei aus.

Weitere Informationen: <https://www.gnu.org/software/coreutils/tail>.

- Zeige die letzten Zeilen einer Datei an:

```
tail --lines {{anzahl_an_zeilen}} {{datei}}
```

- Zeige alle Zeilen einer Datei ab einer bestimmten Zeile an:

```
tail --lines +{{zeile}} {{datei}}
```

- Zeige die letzten Bytes einer Datei an:

```
tail --bytes {{anzahl_an_bytes}} {{datei}}
```

- Lies aus einer Datei, bis Ctrl + C gedrückt wird:

```
tail --follow {{datei}}
```

- Lies aus einer Datei, bis Ctrl + C gedrückt wird, selbst, wenn die Datei nicht zugänglich ist:

```
tail --retry --follow {{datei}}
```

- Zeige die letzten Zeilen einer Datei an und lade alle paar Sekunden neu:

```
tail --lines {{anzahl_an_zeilen}} --sleep-interval  
{{anzahl_an_sekunden}} --follow {{datei}}
```

# tar

Archivierungstool.

Häufig kombiniert mit anderen Methoden zur Komprimierung, wie gzip oder bzip2.

Weitere Informationen: <https://www.gnu.org/software/tar>.

- Erstelle ein Archiv von Dateien:

```
tar cf {{pfad/zu/ziel.tar}} {{pfad/zu/datei1 pfad/zu/datei2 ...}}
```

- Erstelle ein mit gzip komprimiertes Archiv:

```
tar czf {{ziel.tar.gz}} {{pfad/zu/datei1 pfad/zu/datei2 ...}}
```

- Erstelle ein mit gzip komprimiertes Archiv mit relativen Pfaden:

```
tar czf {{pfad/zu/ziel.tar.gz}} --directory={{pfad/zu/verzeichnis}} .
```

- Extrahiere ein (komprimiertes) Archiv in das derzeitige Verzeichnis im ausführlichen Modus:

```
tar xvf {{pfad/zu/quelle.tar[.gz|.bz2|.xz]}}
```

- Extrahiere ein Archiv in ein Verzeichnis:

```
tar xf {{pfad/zu/quelle.tar}} --directory={{pfad/zu/verzeichnis}}
```

- Erstelle ein komprimiertes Archiv und benutze den die Dateierdung des Archivs um die Kompressionsmethode zu wählen:

```
tar caf {{pfad/zu/ziel.tar.xz}} {{pfad/zu/datei1 pfad/zu/datei2 ...}}
```

- Führe die Inhalte eines tar Archivs auf:

```
tar tvf {{pfad/zu/quelle.tar}}
```

- Extrahiere Dateien die mit einem Muster übereinstimmen:

```
tar xf {{pfad/zu/quelle.tar}} --wildcards "{{*.html}}"
```

# termdown

Command-line Countdown-Timer.

Weitere Informationen: <https://github.com/trehn/termdown>.

- Starte Stoppuhr:

```
termdown
```

- Starte einen Countdown von 1 Minute 30 Sekunden:

```
termdown {{1m30s}}
```

- Starte einen Countdown von 1 Minute 30 Sekunden. Nach ablauf der Zeit blinkt das Terminal:

```
termdown {{1m30s}} --blink
```

- Countdown mit Titel über der abgelaufenen Zeit:

```
termdown {{1m30s}} --title "{{Interesting title}}"
```

- Zeige aktuelle Zeit:

```
termdown --time
```



# terraform fmt

Formatierung der Konfiguration gemäß den Konventionen der Terraform-Sprache.

Weitere Informationen: <https://www.terraform.io/docs/commands/fmt.html>.

- Formatieren der Konfiguration im aktuellen Verzeichnis:

```
terraform fmt
```

- Formatieren der Konfiguration im aktuellen Verzeichnis und den Unterverzeichnissen:

```
terraform fmt -recursive
```

- Anzeige der Unterschiede bei Formatierungsänderungen:

```
terraform fmt -diff
```

- Die Dateien mit Formatierungsinkonsistenzen werden nicht auf `stdout` ausgegeben:

```
terraform fmt -list=false
```

# terraform plan

Erzeugen und Anzeigen von Terraform-Ausführungsplänen.

Weitere Informationen: <https://www.terraform.io/docs/cli/commands/plan.html>.

- Erzeugen und Anzeigen des Ausführungsplans im aktuellen Verzeichnis:

```
terraform plan
```

- Einen Plan zur Zerstörung aller derzeit existierenden entfernten Objekte anzeigen:

```
terraform plan -destroy
```

- Anzeigen eines Plans zur Aktualisierung des Terraform-Status und der Ausgabewerte:

```
terraform plan -refresh-only
```

- Werte für Eingabevariablen festlegen:

```
terraform plan -var '{{name1}}={{wert1}}' -var '{{name2}}={{wert2}}'
```

- Anzeigen eines Plans auf eine Teilmenge von Ressourcen:

```
terraform plan -target {{resource_type.resource_name[index]}}
```

- Ausgabe eines Plans als JSON:

```
terraform plan -json
```

- Ausgabe eines Plans in eine separate Datei:

```
terraform plan -no-color > {{pfad/zu/datei}}
```

# terraform

Erstellen und Bereitstellen von Infrastruktur als Code für Cloud-Anbieter.

Weitere Informationen: <https://www.terraform.io/>.

- Initialisieren einer neuen oder bestehenden Terraform-Konfiguration:

```
terraform init
```

- Überprüfen Sie, ob die Konfigurationsdateien syntaktisch gültig sind:

```
terraform validate
```

- Formatierung der Konfiguration gemäß den Konventionen der Terraform-Sprache:

```
terraform fmt
```

- Erzeugen und Anzeigen eines Ausführungsplans:

```
terraform plan
```

- Infrastruktur aufbauen oder ändern:

```
terraform apply
```

- Terraform-verwaltete Infrastruktur zerstören:

```
terraform destroy
```

# tex

Kompiliere eine TeX Quelldatei in ein DVI Dokument.

Weitere Informationen: <https://www.tug.org/begin.html>.

- Kompiliere ein DVI Dokument:

```
tex {{quelldatei.tex}}
```

- Kompiliere ein DVI Dokument und gib ein bestimmtes Output-Verzeichnis an:

```
tex -output-directory={{pfad/zu/verzeichnis}}  
{{quelldatei.tex}}
```

- Kompiliere ein DVI Dokument und stoppe bei jedem Fehler:

```
tex -halt-on-error {{quelldatei.tex}}
```

# texdoc

Suche nach passenden Dokumentationen für (La)TeX Befehle oder Packages.

Weitere Informationen: <https://texdoc.org/index.html>.

- Öffne das erste Suchergebnis im Standard-PDF-Viewer:

```
texdoc {{suche}}
```

- Liste die besten Suchergebnisse auf:

```
texdoc --list {{suche}}
```

- Öffne die vollständige Dokumentation von texdoc:

```
texdoc {{texdoc}}
```

# texliveonfly

Lade fehlende TeX Live Packages während dem Kompilieren einer **.tex** Datei herunter.

Weitere Informationen: <https://ctan.org/pkg/texliveonfly>.

- Lade fehlende Packages während dem Kompilieren herunter:

```
texliveonfly {{quelldatei.tex}}
```

- Verwende einen bestimmten Compiler (standardmäßig pdf<sub>l</sub>atex):

```
texliveonfly --compiler={{compiler}} {{quelldatei.tex}}
```

- Verwende ein bestimmtes Tex Live bin Verzeichnis:

```
texliveonfly --texlive_bin={{pfad/zu/texlive_bin}}  
{{quelldatei.tex}}
```

# tig

Eine interaktive Kommandozeilenoberfläche für Git.

Weitere Informationen: <https://jonas.github.io/tig/doc/manual.html>.

- Zeige die Commits des aktuellen Branches:

```
tig
```

- Zeige die Commits eines bestimmten Branches:

```
tig {{branch}}
```

- Zeige die Commits von bestimmten Dateien oder Verzeichnissen:

```
tig {{pfad1 pfad2 ...}}
```

- Zeige die Unterschiede zwischen zwei Referenzen (wie z.B. Branches oder Tags):

```
tig {{base_ref}}..{{compared_ref}}
```

- Zeige die Commits von allen Branches und Stashes:

```
tig --all
```

- Zeige alle gespeicherten Stashes:

```
tig stash
```

# tldr

Zeigt kurze Zusammenfassungen (tldr-Seiten) von Kommandozeilen-Befehlen an.

Weitere Informationen: <https://github.com/tldr-pages/tldr/blob/main/CLIENT-SPECIFICATION.md#command-line-interface>.

- Zeige die tldr-Seite für einen Befehl an (Hinweis: So bist du hierher gekommen!):

```
tldr {{befehl}}
```

- Zeige die tldr-Seite für `cd` an und überschreibe die Standardplattform:

```
tldr -p {{android|linux|osx|sunos|windows}} {{cd}}
```

- Zeige die tldr-Seite für einen Unterbefehl:

```
tldr {{git-checkout}}
```

- Aktualisiere die lokalen Seiten (wenn er Client Caching unterstützt):

```
tldr -u
```



# tldr

Dieser Befehl ist ein Alias von **tldr-lint**.

Weitere Informationen: <https://github.com/tldr-pages/tldr-lint>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr tldr-lint
```

# tlmgr arch

Dieser Befehl ist ein Alias von **tlmgr platform**.

Weitere Informationen: <https://www.tug.org/texlive/tlmgr.html>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr tlmgr platform
```

# tlmgr

Verwalte Packages und Konfigurationen einer existierenden TeX Live Installation.

Manche Unterbefehle wie **tlmgr paper** sind separat dokumentiert.

Weitere Informationen: <https://www.tug.org/texlive/tlmgr.html>.

- Installiere ein Package und seine Abhängigkeiten:

```
tlmgr install {{package}}
```

- Entferne ein Package und seine Abhängigkeiten:

```
tlmgr remove {{package}}
```

- Zeige Informationen über ein Package an:

```
tlmgr info {{package}}
```

- Aktualisiere alle Packages:

```
tlmgr update --all
```

- Zeige mögliche Aktualisierungen an, ohne Änderungen vorzunehmen:

```
tlmgr update --list
```

- Starte die grafische Oberfläche von tlmgr:

```
tlmgr gui
```

- Liste alle Tex Live Konfigurationen auf:

```
tlmgr conf
```

# todoman

Dieser Befehl ist ein Alias von **todo**.

Weitere Informationen: <https://todoman.readthedocs.io/>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr todo
```

# traefik

Ein HTTP-Reverse-Proxy und Load-Balancer.

Weitere Informationen: <https://traefik.io>.

- Starte den Server mit der Standardkonfiguration:

```
traefik
```

- Starte den Server mit einer benutzerdefinierten Konfigurationsdatei:

```
traefik --ConfigFile {{konfigurationsdatei.toml}}
```

- Starte den Server mit aktiviertem Cluster-Modus:

```
traefik --cluster
```

- Starte den Server mit dem Web-UI:

```
traefik --web
```

# transmission

Dieser Befehl ist ein Alias von **transmission-daemon**.

Weitere Informationen: <https://transmissionbt.com/>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr transmission-daemon
```

# ugrep

Ultraschnelles Suchtool mit Abfrage-TUI.

Weitere Informationen: <https://github.com/Genivia/ugrep>.

- Starte eine interaktive TUI um rekursiv nach Dateien im aktuellen Verzeichnis zu suchen (Strg-Z für Hilfe):

```
ugrep --query
```

- Suche im aktuellen Verzeichnis rekursiv nach Dateien, die einem bestimmten regulären Ausdruck entsprechen:

```
ugrep "{{suchmuster}}"
```

- Suche in einer Datei oder in allen Dateien in einem bestimmten Verzeichnis und zeige die Zeilennummer jedes Treffers:

```
ugrep --line-number "{{suchmuster}}" {{pfad/zu/
datei_oder_verzeichnis}}
```

- Suche in allen Dateien im aktuellen Verzeichnis rekursiv und zeige den Dateinamen jeder passenden Datei:

```
ugrep --files-with-matches "{{suchmuster}}"
```

- Suche nach einem "fuzzy" regulären Ausdruck mit bis zu 3 zusätzlichen, fehlenden oder nicht übereinstimmenden Zeichen:

```
ugrep --fuzzy={{3}} "{{suchmuster}}"
```

- Suche auch in allen komprimierten Dateien und Zip- und tar-Archive:

```
ugrep --decompress "{{suchmuster}}"
```

- Suche nur in Dateien deren Dateinamen mit einem bestimmten glob-Muster übereinstimmen:

```
ugrep --glob="{{glob_muster}}" "{{suchmuster}}"
```

- Suche nur in C++ Quelldateien (verwende --file-type=list, um mögliche Optionen aufzulisten):

```
ugrep --file-type=cpp "{{suchmuster}}"
```

# unclutter

Versteckt den Mauszeiger.

Weitere Informationen: <https://manned.org/unclutter.1x>.

- Verstecke den Mauszeiger nach 3 Sekunden:

```
unclutter -idle {{3}}
```



# unlzma

Dieser Befehl ist ein Alias von **xz**.

Weitere Informationen: <https://manned.org/unlzma>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr xz
```

# unxz

Dieser Befehl ist ein Alias von **xz**.

Weitere Informationen: <https://manned.org/unxz>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr xz
```

# vi

Dieser Befehl ist ein Alias von **vim**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr vim
```

# vifm

Vifm (VI File Manager) ist ein Kommandozeilen-Dateimanager.

Weitere Informationen: <https://github.com/vifm/vifm>.

- Öffne das aktuelle Verzeichnis:

```
vifm .
```

- Öffne angegebene Verzeichnisse auf der linken oder rechten Seite:

```
vifm {{pfad/zu/verzeichnis1 pfad/zu/verzeichnis2 ...}}
```

# vim

Vim (Vi IMproved), ein Befehlszeilen-Texteditor, bietet mehrere Modi für verschiedene Arten der Textmanipulation an.

Das Drücken von **i** schaltet den Editier-Modus ein. **<Esc>** wechselt in den Befehls-Modus, der die Verwendung von Vim-Befehlen ermöglicht.

Weitere Informationen: <https://www.vim.org>.

- Öffne eine Datei:

```
vim {{pfad/zu/datei}}
```

- Öffne eine Datei an einer bestimmten Zeilennummer:

```
vim +{{zeilennummer}} {{pfad/zu/datei}}
```

- Zeige Vim's Benutzeranleitung:

```
:help<Enter>
```

- Speichere und schließe die aktuelle Datei:

```
:wq<Enter>
```

- Mache die letzte Aktion rückgängig:

```
u
```

- Suche nach einem Muster in der Datei (mit n/N zum nächsten/vorherigen Treffer gehen):

```
/{{suchmuster}}<Enter>
```

- Ersetze einen regulären Ausdruck alle Treffer in einer Datei:

```
:%s/{{regulärer_ausdruck}}/{{neuer_text}}/g<Enter>
```

- Zeige Zeilennummern an:

```
:set nu<Enter>
```

# vimdiff

Öffne zwei oder mehr Dateien in Vim und zeige ihre Unterschiede an.

Siehe auch **vim**.

Weitere Informationen: <https://www.vim.org>.

- Öffne zwei Dateien und zeige ihre Unterschiede an:

```
vimdiff {{pfad/zu/datei_1}} {{pfad/zu/datei_2}}
```

- Bewege den Cursor zum linken|rechten Fenster:

```
<Ctrl> + w {{h|l}}
```

- Springe zum vorigen Unterschied:

```
[c
```

- Springe zum nächsten Unterschied:

```
]c
```

- Kopiere die hervorgehobenen Unterschiede vom anderen in das aktuelle Fenster:

```
do
```

- Kopiere die hervorgehobenen Unterschiede vom aktuellen in das andere Fenster:

```
dp
```

- Aktualisiere die hervorgehobenen Unterschiede und Textfaltungen:

```
:diffupdate
```

- Öffne/Schließe die Textfaltung unter dem Cursor:

```
za
```

# vimtutor

Vim-Tutor zum Erlernen grundlegender Vim-Befehle.

Weitere Informationen: <https://manned.org/vimtutor>.

- Öffne den Vim-Tutor in der gegebenen Sprache (de, en, fr, ...):

```
vimtutor {{sprache}}
```

- Beende den Vim-Tutor:

```
<Esc> :q <Enter>
```

# whoami

Gib den Benutzernamen des aktuellen Benutzers aus.

Weitere Informationen: <https://www.gnu.org/software/coreutils/whoami>.

- Gib den aktiven Benutzernamen aus:

```
whoami
```

- Gib den Benutzernamen nach einer Änderung der Benutzeridentität aus:

```
sudo whoami
```



# xzcat

Dieser Befehl ist ein Alias von **xz**.

Weitere Informationen: <https://manned.org/xzcat>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr xz
```

# yarn

JavaScript und Node.js Paket-Manager Alternative.

Weitere Informationen: <https://yarnpkg.com>.

- Installiere ein Modul global:

```
yarn global add {{modul_name}}
```

- Installiere alle in der `package.json` Datei genannten Dependencies (`install` ist optional):

```
yarn install
```

- Installiere ein Modul und füge es als Dependency der `package.json` Datei hinzu (`- -dev` um es als Dev-Dependency zu installieren):

```
yarn add {{modul_name}}@{{version}}
```

- Deinstalliere ein Modul und entferne es von der `package.json` Datei:

```
yarn remove {{modul_name}}
```

- Erstelle interaktiv eine `package.json` Datei:

```
yarn init
```

- Identifiziere ob ein Modul eine Dependency ist und liste andere Module, die von diesem abhängen:

```
yarn why {{modul_name}}
```

# zoxide

Behält den Überblick über die am häufigsten verwendeten Verzeichnisse.

Verwendet einen Ranking-Algorithmus, um zum besten Treffer zu navigieren.

Weitere Informationen: <https://github.com/ajeetsouza/zoxide>.

- Wechsel zu dem Verzeichnis mit dem höchsten Rang, das "foo" im Namen enthält:

```
zoxide query {{foo}}
```

- Wechsel in das höchstrangige Verzeichnis, das "foo" und danach "bar" enthält:

```
zoxide query {{foo}} {{bar}}
```

- Starte eine interaktive Verzeichnissuche (erfordert `fzf`):

```
zoxide query --interactive
```

- Füge ein Verzeichnis hinzu oder erhöhe seinen Rang:

```
zoxide add {{path/to/directory}}
```

- Entferne interaktiv ein Verzeichnis aus der Datenbank von `zoxide`:

```
zoxide remove {{path/to/directory}} --interactive
```

- Generiere Shell-Konfigurationen für Befehls-Aliase (`z`, `za`, `zi`, `zq`, `zr`) für die angegebene Shell:

```
zoxide init {{bash|fish|zsh}}
```

# zsh

Z SHell.

Mit **bash** und **sh** kompatible Eingabeaufforderung.

Weitere Informationen: <https://www.zsh.org>.

- Starte eine interaktive Eingabeaufforderung:

```
zsh
```

- Führe Parameter als Befehl aus:

```
zsh -c {{befehl}}
```

- Führe Befehle aus einem Skript aus:

```
zsh {{pfad/zu/skript}}
```

- Führe Befehle aus einem Skript aus und schreibe die Befehle in die Konsole:

```
zsh --xtrace {{pfad/zu/skript}}
```

- Starte eine interaktive Eingabeaufforderung, in der jeder Befehl ausgegeben wird, bevor er ausgeführt wird:

```
zsh --verbose
```

- Führe einen Befehl innerhalb von Zsh mit ausgeschalteten Glob-Mustern aus:

```
noglob {{befehl}}
```

Linux

# a2disconf

Deaktiviert eine Apache-Konfigurationsdatei auf Debian-basierten Betriebssystemen.

Weitere Informationen: <https://manpages.debian.org/latest/apache2/a2disconf.8.en.html>.

- Deaktiviere eine Konfigurationsdatei:

```
sudo a2disconf {{konfigurationsdatei}}
```

- Zeige keine Informationsnachrichten an:

```
sudo a2disconf --quiet {{konfigurationsdatei}}
```

# a2dismod

Deaktiviert ein Apache-Modul auf Debian-basierten Betriebssystemen.

Weitere Informationen: <https://manpages.debian.org/latest/apache2/a2dismod.8.en.html>.

- Deaktiviere ein Modul:

```
sudo a2dismod {{modul}}
```

- Zeige keine Informationsnachrichten an:

```
sudo a2dismod --quiet {{modul}}
```

# a2dissite

Deaktiviert einen Apache virtuellen Host auf Debian-basierten Betriebssystemen.

Weitere Informationen: <https://manpages.debian.org/latest/apache2/a2dissite.8.en.html>.

- Deaktiviere einen virtuellen Host:

```
sudo a2dissite {{virtueller_host}}
```

- Zeige keine Informationsnachrichten an:

```
sudo a2dissite --quiet {{virtueller_host}}
```



# a2enconf

Aktiviert eine Apache-Konfigurationsdatei auf Debian-basierten Betriebssystemen.

Weitere Informationen: <https://manpages.debian.org/latest/apache2/a2enconf.8.en.html>.

- Aktiviere eine Konfigurationsdatei:

```
sudo a2enconf {{konfigurationsdatei}}
```

- Zeige keine Informationsnachrichten an:

```
sudo a2enconf --quiet {{konfigurationsdatei}}
```

# a2enmod

Aktiviert ein Apache-Modul auf Debian-basierten Betriebssystemen.

Weitere Informationen: <https://manpages.debian.org/latest/apache2/a2enmod.8.en.html>.

- Aktiviere ein Modul:

```
sudo a2enmod {{modul}}
```

- Zeige keine Informationsnachrichten an:

```
sudo a2enmod --quiet {{modul}}
```

# a2ensite

Aktiviert einen Apache virtuellen Host auf Debian-basierten Betriebssystemen.

Weitere Informationen: <https://manpages.debian.org/latest/apache2/a2ensite.8.en.html>.

- Aktiviere einen virtuellen Host:

```
sudo a2ensite {{virtueller_host}}
```

- Zeige keine Informationsnachrichten an:

```
sudo a2ensite --quiet {{virtueller_host}}
```

# a2query

Zeigt Apache Laufzeitkonfigurationen auf Debian-basierten Betriebssystemen an.

Weitere Informationen: <https://manpages.debian.org/latest/apache2/a2query.html>.

- Zeige aktivierte Apache-Module an:

```
sudo a2query -m
```

- Prüfe, ob ein bestimmtes Modul installiert ist:

```
sudo a2query -m {{modulname}}
```

- Zeige aktivierte virtuelle Hosts an:

```
sudo a2query -s
```

- Zeige das aktuell aktivierte Multi-Processing-Modul an:

```
sudo a2query -M
```

- Zeige die Apache-Versionsnummer an:

```
sudo a2query -v
```

# aa-complain

Setze eine AppArmor-Richtlinie in den Beschwerde-Modus.

Siehe auch: **aa-disable**, **aa-enforce**, **aa-status**.

Weitere Informationen: [https://gitlab.com/apparmor/apparmor/-/wikis/manpage\\_aa-complain.8](https://gitlab.com/apparmor/apparmor/-/wikis/manpage_aa-complain.8).

- Setze eine Richtlinie in den Beschwerde-Modus:

```
sudo aa-complain {{pfad/zum/profil}}
```

- Setze Richtlinien in den Beschwerde-Modus:

```
sudo aa-complain --dir {{pfad/zum/profil}}
```

# abbr

Abkürzungen für die fish shell verwalten.

Die vom Nutzer definierten Abkürzungen werden nach der Eingabe durch die Langversionen ersetzt.

Weitere Informationen: <https://fishshell.com/docs/current/cmds/abbr.html>.

- Neue Abkürzung hinzufügen:

```
abbr --add {{abkürzungsname}} {{befehl}} {{befehlsparameter}}
```

- Vorhandene Abkürzung umbenennen:

```
abbr --rename {{alter_name}} {{neuer_name}}
```

- Vorhandene Abkürzung löschen:

```
abbr --erase {{abkürzungsname}}
```

- Abkürzungen eines anderen Host über SSH importieren:

```
ssh {{host_name}} abbr --show | source
```

# ac

Zeigt an, wie lange Benutzer verbunden waren.

Weitere Informationen: <https://www.gnu.org/software/acct/manual/accounting.html#ac>.

- Zeige, wie viele Stunden der aktuelle Benutzer verbunden war:

```
ac
```

- Zeige, wie viele Stunden jeder Benutzer verbunden war:

```
ac --individual-totals
```

- Zeige, wie viele Stunden ein bestimmter Benutzer verbunden war:

```
ac --individual-totals {{nutzernamen}}
```

- Zeige, wie viele Stunden ein bestimmter Benutzer pro Tag verbunden war:

```
ac --daily-totals --individual-totals {{nutzernamen}}
```

- Zeige zusätzliche Details:

```
ac --compatibility
```

# acpi

Zeigt den Akkustatus oder Temperatur-Informationen an.

Weitere Informationen: <https://sourceforge.net/projects/acpiclient/files/acpiclient/>.

- Zeige Informationen über den Akku an:

```
acpi
```

- Zeige Informationen zur Temperatur an:

```
acpi -t
```

- Zeige Informationen über die Kühlung an:

```
acpi -c
```

- Zeige Temperatur-Informationen in Fahrenheit an:

```
acpi -tf
```

- Zeige alle Informationen an:

```
acpi -V
```

- Extrahiere Informationen von `/proc`, anstatt von `/sys`:

```
acpi -p
```



# add-apt-repository

Verwalte apt-Repository-Definitionen.

Weitere Informationen: <https://manned.org/apt-add-repository>.

- Füge ein neues apt-Repository hinzu:

```
add-apt-repository {{repository_spec}}
```

- Entferne ein apt-Repository:

```
add-apt-repository --remove {{repository_spec}}
```

- Aktualisiere den Paketcache nach dem Hinzufügen eines Repositories:

```
add-apt-repository --update {{repository_spec}}
```

- Erlaube das Herunterladen von Quellpaketen aus dem Repository:

```
add-apt-repository --enable-source {{repository_spec}}
```

# addpart

Informiert den Linux-Kernel über die Existenz der angegebenen Partition.

Dieser Befehl ist ein einfacher Wrapper um den **add partition** ioctl.

Weitere Informationen: <https://manned.org/addpart>.

- Informiere den Kernel über die Existenz der angegebenen Partition:

```
addpart {{gerät}} {{partition}} {{start}} {{länge}}
```

# addr2line

Konvertiere Adressen von Binärdateien in Dateinamen und Zeilennummern.

Weitere Informationen: <https://manned.org/addr2line>.

- Zeige den Dateinamen und die Zeilennummer des Quellcodes von einer Befehlsadresse einer ausführbaren Datei an:

```
addr2line --exe={{pfad/zur/ausführbaren_datei}} {{adresse}}
```

- Zeige den Funktionsnamen, Dateinamen und Zeilennummer an:

```
addr2line --exe={{pfad/zum/executable}} --functions  
{{adresse}}
```

- Entmangle den Funktionsnamen für C++ Code:

```
addr2line --exe={{pfad/zum/executable}} --functions --  
demangle {{adresse}}
```

# adduser

Tool um Benutzer hinzuzufügen.

Weitere Informationen: <https://manpages.debian.org/latest/adduser/adduser.html>.

- Erstelle einen neuen Benutzer mit einem Standard-Home-Verzeichnis und Aufforderung an den Benutzer, ein Passwort festzulegen:

```
adduser {{benutzername}}
```

- Erstelle einen neuen Benutzer ohne Home-Verzeichnis:

```
adduser --no-create-home {{benutzername}}
```

- Erstelle einen neuen Benutzer mit einem Home-Verzeichnis unter dem angegebenen Pfad:

```
adduser --home {{pfad/zu/home}} {{benutzername}}
```

- Erstelle einen neuen Benutzer, bei dem die angegebene Shell als Anmeldeshell eingestellt ist:

```
adduser --shell {{pfad/zu/shell}} {{benutzername}}
```

- Erstelle einen neuen Benutzer und füge ihn zur angegebenen Gruppe hinzu:

```
adduser --ingroup {{gruppe}} {{benutzername}}
```

# alien

Ein Installations-Paket in ein anderes Format umwandeln.

Weitere Informationen: <https://manned.org/alien>.

- Ein spezifisches Installationspaket in das Debian Format umwandeln (.deb Erweiterung):

```
sudo alien --to-deb {{pfad/zum/paket}}
```

- Ein spezifisches Installationspaket in das Red Hat Format umwandeln (.rpm Erweiterung):

```
sudo alien --to-rpm {{pfad/zum/paket}}
```

- Ein spezifisches Installationspaket in das Slackware Format umwandeln (.tgz Erweiterung):

```
sudo alien --to-tgz {{pfad/zum/paket}}
```

- Ein spezifisches Installationspaket in das Debian Format umwandeln und auf dem System installieren:

```
sudo alien --to-deb --install {{pfad/zum/paket}}
```

# alpine

Ein E-Mail und Usenet Client mit pico/nano-inspirierten Interface.

Unterstützt die meisten modernen IMAP Server.

Weitere Informationen: <https://manned.org/alpine>.

- Öffne Alpine:

```
alpine
```

- Öffne alpine im E-Mail-Editor, um eine E-Mail an eine bestimmte Adresse zu verfassen:

```
alpine {{email@example.net}}
```

- Beende Alpine:

```
q + y
```

# alternatives

Dieser Befehl ist ein Alias von **update-alternatives**.

Weitere Informationen: <https://manned.org/alternatives>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr update-alternatives
```

# amixer

Mixer für den ALSA-Soundkarten-Treiber.

Weitere Informationen: <https://manned.org/amixer>.

- Erhöhe den Gesamtpegel um 10%:

```
amixer -D pulse sset Master {{10%+}}
```

- Verringere den Gesamtpegel um 10%:

```
amixer -D pulse sset Master {{10%-}}
```



# anbox

Führe Android-Anwendungen auf jedem Linux-Betriebssystem aus.

Weitere Informationen: <https://manned.org/anbox>.

- Starte Anbox im App Manager:

```
anbox launch --package={{org.anbox.appmgr}} --  
component={{org.anbox.appmgr.AppViewActivity}}
```

# apache2ctl

Das CLI Tool um den Apache HTTP Web Server zu administrieren.

Dieser Befehl wird mit Debian-basierten Betriebssystemen geliefert, für RHEL siehe **httpd**.

Weitere Informationen: <https://manpages.debian.org/latest/apache2/apache2ctl.8.en.html>.

- Starte den Apache daemon. Gibt einen Fehler aus, wenn er bereits läuft:

```
sudo apache2ctl start
```

- Stoppe den Apache Daemon:

```
sudo apache2ctl stop
```

- Starte den Apache Daemon neu:

```
sudo apache2ctl restart
```

- Überprüfe die Syntax einer Konfigurationsdatei:

```
sudo apache2ctl -t
```

- Liste alle geladenen Module auf:

```
sudo apache2ctl -M
```

# apk

Alpine Linux-Paketverwaltungstool.

Weitere Informationen: [https://wiki.alpinelinux.org/wiki/Alpine\\_Linux\\_package\\_management](https://wiki.alpinelinux.org/wiki/Alpine_Linux_package_management).

- Aktualisiere die Indizes von allen externen Repositories:

```
apk update
```

- Installiere ein neues Paket:

```
apk add {{paket}}
```

- Entferne ein Paket:

```
apk del {{paket}}
```

- Repariere oder aktualisiere ein Paket, ohne die Hauptabhängigkeiten zu ändern:

```
apk fix {{paket}}
```

- Suche Pakete mit einem Schlüsselwort:

```
apk search {{schlüsselwort}}
```

- Erhalte Informationen über ein bestimmtes Paket:

```
apk info {{paket}}
```

# aplay

Command-line Musik Player für den ALSA-Soundkarten-Treiber.

Weitere Informationen: <https://manned.org/aplay>.

- Spiele eine bestimmte Datei (Abtastrate, Bittiefe, etc. werden automatisch für das Dateiformat erkannt):

```
aplay {{pfad/zu/datei}}
```

- Spiele die ersten 10 Sekunden einer bestimmten Datei mit 2500 Hz:

```
aplay --duration={{10}} --rate={{2500}} {{pfad/zu/datei}}
```

- Spiele die rohe Datei mit 22050 Hz, mono, 8-bit, als Mu-Lw .au Datei:

```
aplay --channels={{1}} --file-type {{raw}} --rate={{22050}}  
--format={{mu_law}} {{pfad/zu/datei}}
```

# apport-bug

Reiche einen Fehlerbericht über Ubuntu ein.

Weitere Informationen: <https://wiki.ubuntu.com/Apport>.

- Reiche einen Fehlerbericht über das gesamte System ein:

```
apport-bug
```

- Reiche einen Fehlerbericht über ein bestimmtes Paket ein:

```
apport-bug {{paket}}
```

- Reiche einen Fehlerbericht über eine bestimmte ausführbare Datei ein:

```
apport-bug {{pfad/zum/executable}}
```

- Reiche einen Fehlerbericht über einen bestimmten Prozess ein:

```
apport-bug {{PID}}
```

# apt-add-repository

Editiere die Repository-Listen.

Weitere Informationen: <https://manpages.debian.org/latest/software-properties-common/apt-add-repository.1.html>.

- Füge ein neues Repository hinzu:

```
apt-add-repository {{repository_spec}}
```

- Entferne ein Repository:

```
apt-add-repository --remove {{repository}}
```

- Aktualisiere den Cache nachdem das Repository hinzugefügt wurde:

```
apt-add-repository --update {{repository}}
```

- Aktiviere Source-Pakete:

```
apt-add-repository --enable-source {{repository}}
```

# apt-cache

Debian und Ubuntu-Paketsuche.

Weitere Informationen: <https://manpages.debian.org/latest/apt/apt-cache.8.html>.

- Suche nach einem Paket in deinen aktuellen Paketquellen:

```
apt-cache search {{suchbegriff}}
```

- Zeige die Paketinformationen zu einem Paket:

```
apt-cache show {{paket}}
```

- Überprüfe ob ein Paket installiert und up to date ist:

```
apt-cache policy {{paket}}
```

- Zeige die Abhängigkeiten eines Pakets:

```
apt-cache depends {{paket}}
```

- Zeige Pakete die von einem bestimmten Paket abhängen:

```
apt-cache rdepends {{paket}}
```

# apt-file

Suche nach Dateien in apt-Paketen, auch in den nicht-installierten.

Weitere Informationen: <https://manpages.debian.org/latest/apt-file/apt-file.1.html>.

- Aktualisiere die Metadatenbank:

```
sudo apt update
```

- Suche nach Paketen, die die/den spezifizierten Pfad/Datei enthalten:

```
apt-file {{search|find}} {{pfad/zur/datei}}
```

- Liste die Inhalte eines bestimmten Pakets auf:

```
apt-file {{show|list}} {{paketname}}
```

- Suche nach Paketen auf die die Regular Expression zutrifft:

```
apt-file {{search|find}} --regexp {{regular_expression}}
```



# apt-get

Debian und Ubuntu Paket Management Tool.

Suche mit **apt-cache** nach Paketen.

Weitere Informationen: <https://manpages.debian.org/latest/apt/apt-get.8.html>.

- Aktualisiere die Liste der Paketquellen (es wird empfohlen diesen Befehl zu Beginn auszuführen):

```
apt-get update
```

- Installiere ein Paket oder aktualisiere es zur neuesten Version:

```
apt-get install {{paket}}
```

- Entferne ein Paket:

```
apt-get remove {{paket}}
```

- Entferne ein Paket und die dazugehörigen Konfigurationen:

```
apt-get purge {{paket}}
```

- Aktualisiere alle Pakete auf die neueste Version:

```
apt-get upgrade
```

- Reinige das Repository:

```
apt-get autoclean
```

- Entferne alle Pakete, die nicht mehr benötigt werden:

```
apt-get autoremove
```

- Aktualisiere alle Pakete (wie `upgrade`), aber entfernt alle obsoleten Pakete:

```
apt-get dist-upgrade
```

# apt-key

Schlüssel-Management-Tool für den APT-Paket-Manager auf Debian und Ubuntu.

Notiz: **apt-key** ist veraltet (außer für **apt-key del** in Maintainerskripten).

Weitere Informationen: <https://manpages.debian.org/latest/apt/apt-key.8.html>.

- Liste alle vertrauten Schlüssel auf:

```
apt-key list
```

- Füge einen Schlüssel hinzu:

```
apt-key add {{public_key_file.asc}}
```

- Lösche einen Schlüssel:

```
apt-key del {{key_id}}
```

- Füge einen Remote-Schlüssel hinzu:

```
wget -q0 - {{https://host.tld/filename.key}} | apt-key add -
```

- Füge einen Schlüssel von einem Schlüsselservers hinzu nur mit der Schlüssel-ID:

```
apt-key adv --keyserver {{pgp.mit.edu}} --recv {{KEYID}}
```

# apt-mark

Tool um den Status eines installierten Paketes zu verändern.

Weitere Informationen: <https://manpages.debian.org/latest/apt/apt-mark.8.html>.

- Markiere ein Paket als automatisch installiert:

```
sudo apt-mark auto {{paketname}}
```

- Halte ein Paket auf seiner aktuellen Version und verhindere dass es aktualisiert wird:

```
sudo apt-mark hold {{paketname}}
```

- Erlaube dass ein Paket wieder aktualisiert werden darf:

```
sudo apt-mark unhold {{paketname}}
```

- Zeige manuell installierte Pakete:

```
apt-mark showmanual
```

- Zeige gehaltene Pakete die nicht aktualisiert werden dürfen:

```
apt-mark showhold
```

# apt

Debian und Ubuntu Paket Management Tool.

Empfohlene Alternative zu **apt-get** seit Ubuntu 16.04.

Weitere Informationen: <https://manpages.debian.org/latest/apt/apt.8.html>.

- Aktualisiere die Liste der Paketquellen (es wird empfohlen, diesen Befehl zu Beginn auszuführen):

```
sudo apt update
```

- Suche nach einem Paket:

```
apt search {{paket}}
```

- Zeige Informationen über ein Paket:

```
apt show {{paket}}
```

- Installiere ein Paket oder aktualisiere es zur neusten Version:

```
sudo apt install {{paket}}
```

- Entferne ein Paket:

```
sudo apt remove {{paket}}
```

- Aktualisiere alle installierten Pakete auf die neueste Version:

```
sudo apt upgrade
```

- Liste alle Pakete auf:

```
apt list
```

- Liste alle installierten Pakete auf:

```
apt list --installed
```

# aptitude

Debian und Ubuntu Paket Management Tool.

Weitere Informationen: <https://manpages.debian.org/latest/aptitude/aptitude.8.html>.

- Synchronisiere die Paketliste und verfügbaren Versionen. Dieser Command sollte zuerst ausgeführt werden bevor weitere aptitude Commands ausgeführt werden:

```
aptitude update
```

- Installiere ein neues Paket und seine Abhängigkeiten:

```
aptitude install {{paket}}
```

- Suche nach einem Paket:

```
aptitude search {{paket}}
```

- Suche nach einem installierten Paket (?installed ist ein aptitude Suchbegriff):

```
aptitude search '?installed ({{paket}})'
```

- Entferne ein Paket und alle Abhängigkeiten:

```
aptitude remove {{paket}}
```

- Aktualisiere installierte Pakete auf die neusten Versionen:

```
aptitude upgrade
```

- Aktualisiere installierte Pakete (wie `aptitude upgrade`), inklusive obsoleter Pakete und installiere zusätzliche Pakete um die neuen Paket-Abhängigkeiten zu erfüllen:

```
aptitude full-upgrade
```

- Friere ein installiertes Paket ein und verhindere, dass es automatisch aktualisiert wird:

```
aptitude hold '?installed({{paket}})'
```

# arch-chroot

Erweiterter **chroot**-Befehl zur Unterstützung des Arch-Linux-Installationsprozesses.

Weitere Informationen: <https://man.archlinux.org/man/arch-chroot.8>.

- Starte eine interaktive Shell (Standardmäßig Bash) in einem neuen Root-Verzeichnis:

```
arch-chroot {{pfad/zu/neuem/root}}
```

- Spezifiziere den Benutzer (nicht der jetzige Benutzer) der die Shell ausführt:

```
arch-chroot -u {{anderer_benutzer}} {{pfad/zu/neuem/root}}
```

- Führe einen benutzerdefinierten Befehl (anstelle des Standardbefehls Bash) im neuen Root-Verzeichnis aus:

```
arch-chroot {{pfad/zu/neuem/root}} {{befehl}}  
{{befehlsparameter}}
```

- Gib die Shell an, die nicht die Standard-Shell Bash ist (in diesem Fall sollte das Paket **zsh** auf dem Zielsystem installiert worden sein):

```
arch-chroot {{pfad/zu/neuem/root}} {{zsh}}
```

# archey

Simples Tool um System-Informationen stylisch zu präsentieren.

Weitere Informationen: <https://lclarkmichalek.github.io/archey3/>.

- Zeige System-Informationen:

```
archey
```

# archinstall

Geführte Arch Linux-Installation.

Weitere Informationen: <https://archinstall.readthedocs.io>.

- Starte den interaktiven Installer:

```
archinstall
```

- Starte einen voreingestellten Installer:

```
archinstall {{minimal|unattended}}
```



# archlinux-java

Ein Helferskript, das Funktionen für Java-Umgebungen bereitstellt.

Weitere Informationen: [https://wiki.archlinux.org/title/Java#Switching\\_between\\_JVM](https://wiki.archlinux.org/title/Java#Switching_between_JVM).

- Liste installierte Java-Umgebungen:

```
archlinux-java status
```

- Setze die default Java-Umgebung:

```
archlinux-java set {{java_environment}}
```

- Entferne die default Java-Umgebung:

```
archlinux-java unset
```

- Setze die default Java-Umgebung automatisch:

```
archlinux-java fix
```

# arecord

Sound Recorder für den ALSA-Soundkarten-Treiber.

Weitere Informationen: <https://manned.org/arecord>.

- Nehme einen Schnipsel in CD-Qualität auf (beende die Aufnahme mit CTRL-C):

```
arecord -vv --format=cd {{pfad/zur/datei.wav}}
```

- Nehme einen Schnipsel in CD-Qualität auf mit einer festen Länge von 10 Sekunden:

```
arecord -vv --format=cd --duration={{10}} {{pfad/zur/datei.wav}}
```

- Nehme einen Schnipsel auf und speichere es als MP3 (beende die Aufnahme mit CTRL-C):

```
arecord -vv --format=cd --file-type raw | lame -r - {{path/to/file.mp3}}
```

- Liste alle Soundkarten und digitalen Ausgabe Geräte:

```
arecord --list-devices
```

- Benutze das interaktive Interface (z.B. Space oder Enter für Play oder Pause):

```
arecord --interactive
```

# arithmetic

Quiz über simple arithmetische Probleme.

Weitere Informationen: <https://manpages.debian.org/latest/bsdgames/arithmetic.6.en.html>.

- Starte ein arithmetisches Quiz:

```
arithmetic
```

- Spezifiziere einen oder mehr arithmetische [O]peratoren um Probleme mit ihnen zu bekommen:

```
arithmetic -o {{+|-|x|/}}
```

- Gib eine Reichweite. Additions- und Multiplikationsprobleme werden Zahlen zwischen 0 und der gegebenen Reichweite enthalten. Subtraktionen und Divisionen werden Zahlen zwischen -1 und der gegebenen Reichweite enthalten:

```
arithmetic -r {{7}}
```

# ark

KDE-Archivierungstool.

Weitere Informationen: <https://docs.kde.org/stable5/en/ark/ark/>.

- Extrahiere ein Archiv ins aktuelle Verzeichnis:

```
ark --batch {{pfad/zu/archiv}}
```

- Verändere das Verzeichnis in das extrahiert wird:

```
ark --batch --destination {{pfad/zu/verzeichnis}} {{pfad/zu/archiv}}
```

- Erstelle ein Archiv wenn es nicht existiert und füge Dateien hinzu:

```
ark --add-to {{pfad/zu/archiv}} {{pfad/zu/datei1 pfad/zu/datei2 ...}}
```

# as

Portabler GNU assembler.

Hauptsächlich beabsichtigt um output von **gcc** für **ld** vorzubereiten.

Weitere Informationen: <https://manned.org/as>.

- Assemble eine Datei und schreibe den Output in eine in **a.out**:

```
as {{datei.s}}
```

- Assemble den Output einer gegebenen Datei:

```
as {{datei.s}} -o {{out.o}}
```

- Generiere den Output schneller indem Leerzeichen und Kommentare nicht verarbeitet werden. (Sollte nur für vertrauenswürdige Compiler benutzt werden):

```
as -f {{datei.s}}
```

- Inkludiere einen gegebenen Pfad in der Liste von Verzeichnissen für die Suche nach Dateien:

```
as -I {{pfad/zum/verzeichnis}} {{datei.s}}
```

# ascii

Zeige ASCII-Aliase von Buchstaben.

Weitere Informationen: <http://www.catb.org/~esr/ascii/>.

- Zeige ASCII-Aliase eines Buchstabens:

```
ascii {{a}}
```

- Zeige ASCII-Aliase im kurzen, skriptfreundlichen Modus:

```
ascii -t {{a}}
```

- Zeige ASCII-Aliase von mehreren Buchstaben:

```
ascii -s {{tldr}}
```

- Zeige die ASCII-Tabelle im Dezimalsystem:

```
ascii -d
```

- Zeige die ASCII-Tabelle im Hexadezimalsystem:

```
ascii -x
```

- Zeige die ASCII-Tabelle im Oktalsystem:

```
ascii -o
```

- Zeige die ASCII-Tabelle im Binärsystem:

```
ascii -b
```

- Zeige eine Übersicht über alle Optionen und die komplette ASCII-Tabelle:

```
ascii
```

# asciiart

Konvertiere Bilder zu ASCII.

Weitere Informationen: <https://github.com/nodanaonlyzuul/asciiart>.

- Lese ein Bild aus einer Datei und zeige es als ASCII:

```
asciiart {{pfad/zur/datei.jpg}}
```

- Lese ein Bild aus einer URL und zeige es als ASCII:

```
asciiart {{www.example.com/bild.jpg}}
```

- Wähle die Breite der Ausgabe (standardmäßig 100):

```
asciiart --width {{50}} {{pfad/zum/bild.jpg}}
```

- Zeige die Ausgabe in Farbe:

```
asciiart --color {{pfad/zum/bild.jpg}}
```

- Wähle das Ausgabeformat (standardmäßig text):

```
asciiart --format {{text|html}} {{pfad/zum/bild.jpg}}
```

- Invertiere die Zeichentabelle:

```
asciiart --invert-chars {{pfad/zum/bild.jpg}}
```

# aspell

Interaktiver Korrekturleser.

Weitere Informationen: <http://aspell.net/>.

- Überprüfe eine einzelne Datei auf Fehler:

```
aspell check {{pfad/zur/datei}}
```

- Liste falsch geschriebene Worte von Standard Input:

```
cat {{file}} | aspell list
```

- Zeige verfügbare Wörterbuchsprachen:

```
aspell dicts
```

- Nutze `aspell` mit einem anderen Wörterbuch (nimmt 2-Zeichen-Locale laut ISO 639 Sprach Code):

```
aspell --lang={{cs}}
```

- Zeige alle falsch geschriebenen Wörter von Standard Input und ignoriere alle Wörter in einer persönlichen Wortliste:

```
cat {{file}} | aspell --personal={{persönliche-wort-  
liste.pws}} list
```



# batcat

Dieser Befehl ist ein Alias von **bat**.

Weitere Informationen: <https://github.com/sharkdp/bat>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr bat
```

# beep

Ein Befehl, um den PC-Lautsprecher zu steuern.

Weitere Informationen: <https://github.com/spkr-beep/beep>.

- Gib einen Ton aus:

```
beep
```

- Gib einen Ton mehrmals aus:

```
beep -r {{wiederholungen}}
```

- Gib einen Ton mit einer bestimmten Frequenz (Hz) und Länge (Millisekunden) aus:

```
beep -f {{frequenz}} -l {{länge}}
```

- Spiele jede neue Frequenz und Länge als einen eigenen Ton:

```
beep -f {{frequenz}} -l {{länge}} -n -f {{frequenz}} -l {{länge}}
```

- Spiele die C-Dur-Tonleiter:

```
beep -f {{262}} -n -f {{294}} -n -f {{330}} -n -f {{349}} -n  
-f {{392}} -n -f {{440}} -n -f {{494}} -n -f {{523}}
```

# bspwm

Dieser Befehl ist ein Alias von **bsp**c.

Weitere Informationen: <https://github.com/baskerville/bspwm>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr bspc
```

# cal

Zeigt einen Kalender an, in dem der aktuelle Tag markiert ist.

Weitere Informationen: <https://manned.org/cal>.

- Zeige den Kalender für den aktuellen Monat an:

```
cal
```

- Zeige den vorherigen, aktuellen und nächsten Monat an:

```
cal -3
```

- Verwende Montag als ersten Tag der Woche:

```
cal --monday
```

- Zeige den Kalender für ein bestimmtes Jahr an (4 Ziffern):

```
cal {{jahr}}
```

- Zeige den Kalender für einen bestimmten Monat eines Jahres an:

```
cal {{monat}} {{jahr}}
```

# CC

Dieser Befehl ist ein Alias von **gcc**.

Weitere Informationen: <https://gcc.gnu.org>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr gcc
```

# certbot

The Let's Encrypt Agent zum automatischen Erhalten und Erneuern von TLS-Zertifikaten.

Nachfolger von **letsencrypt**.

Weitere Informationen: <https://certbot.eff.org/docs/using.html>.

- Beziehe ein neues Zertifikat über die webroot-Autorisierung, aber ohne dieses automatisch zu installieren:

```
sudo certbot certonly --webroot --webroot-path {{pfad/zu/webroot}} --domain {{subdomain.beispiel.de}}
```

- Beziehe ein neues Zertifikat über die nginx-Autorisierung und automatische Installation des neuen Zertifikats:

```
sudo certbot --nginx --domain {{subdomain.beispiel.de}}
```

- Beziehe ein neues Zertifikat über die apache-Autorisierung und automatische Installation des neuen Zertifikats:

```
sudo certbot --apache --domain {{subdomain.beispiel.de}}
```

- Erneuere alle Let's Encrypt Zertifikate die in 30 Tagen oder weniger auslaufen (nicht vergessen alle Server, die diese nutzen, neu zu starten):

```
sudo certbot renew
```

- Simuliere die Zertifikatserneuerung, ohne diese zu speichern:

```
sudo certbot --webroot --webroot-path {{pfad/zu/webroot}} --domain {{subdomain.beispiel.de}} --dry-run
```

- Beziehe ein Test-Zertifikat:

```
sudo certbot --webroot --webroot-path {{pfad/zu/webroot}} --domain {{subdomain.beispiel.de}} --test-cert
```

# cdisk

Ein Programm zur Verwaltung von Partitionstabellen mittels einer Curses-basierten UI.

Weitere Informationen: <https://manned.org/cdisk>.

- Öffne das Partitionierungsinterface für eine bestimmte Festplatte:

```
cdisk {{/dev/sdX}}
```

- Erzeuge und bearbeite eine neue Partitionierungstabelle für eine bestimmte Festplatte:

```
cdisk --zero {{/dev/sdX}}
```

# cgroups

Dieser Befehl ist ein Alias von **cgclassify**.

Weitere Informationen: <https://www.kernel.org/doc/Documentation/cgroup-v2.txt>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr cgclassify
```



# cp

Kopiere Dateien und Verzeichnisse.

Weitere Informationen: <https://www.gnu.org/software/coreutils/cp>.

- Kopiere eine Datei an einen anderen Ort:

```
cp {{pfad/zur/ausgangs_datei.ext}} {{pfad/zur/ziel_datei.ext}}
```

- Kopiere eine Datei in ein anderes Verzeichnis und behalte den Dateinamen bei:

```
cp {{pfad/zur/ausgang_datei.ext}} {{pfad/zum/ziel_verzeichnis}}
```

- Kopiere die Inhalte eines Verzeichnisses rekursiv zu einem neuen Ort (wenn das Ziel existiert, wird das Verzeichnis ins bestehende Ziel Verzeichnis kopiert):

```
cp -r {{pfad/zum/ausgangs_verzeichnis}} {{pfad/zum/ziel_verzeichnis}}
```

- Kopiere ein Verzeichnis rekursiv im ausführlichen Modus (zeigt die Dateien die kopiert werden):

```
cp -vr {{pfad/zum/ausgangs_verzeichnis}} {{pfad/zum/ziel_verzeichnis}}
```

- Kopiere text Dateien zu einem anderen Ort im interaktiven Modus (fragt die Nutzer:in bevor eine Datei überschrieben wird):

```
cp -i {{*.txt}} {{pfad/zum/ziel_verzeichnis}}
```

- Folge symbolischen Verzeichnislinks vorm Kopieren:

```
cp -L {{link}} {{pfad/zum/ziel_verzeichnis}}
```

- Benutze den vollen Pfad der Ausgangsdateien und erstelle alle fehlenden Verzeichnisse beim Kopieren:

```
cp --parents {{quelle/pfad/zur/datei}} {{pfad/zur/ziel_datei}}
```

# cuyo

Tetris-ähnliches Spiel.

Weitere Informationen: <https://www.karimmi.de/cuyo/>.

- Starte ein neues Spiel:

`cuyo`

- Verschiebe die Teile horizontal:

`{{A|D|Pfeiltaste Links|Pfeiltaste Rechts}}`

- Drehe die Teile:

`{{W|Pfeiltaste nach oben}}`

- Lasse die Teile schnell fallen:

`{{S|Pfeiltaste nach unten}}`

# dnf

Paketmanagement Tool für RHEL, Fedora, und CentOS (ersetzt yum).

Weitere Informationen: <https://dnf.readthedocs.io>.

- Aktualisiere alle Pakete auf die neueste Version:

```
sudo dnf upgrade
```

- Suche nach Paketen:

```
dnf search {{schlüsselwort}}
```

- Zeige Daten über ein bestimmtes Paket an:

```
dnf info {{paket}}
```

- Installiere ein neues Paket:

```
sudo dnf install {{paket}}
```

- Entferne ein Paket:

```
sudo dnf remove {{paket}}
```

- Liste alle Pakete auf:

```
dnf list --installed
```

- Zeige welches Paket eine Datei besitzt:

```
dnf provides {{pfad/zu/datei}}
```

# dpkg

Debian Paketmanager.

Manche Unterbefehle wie **dpkg deb** sind separat dokumentiert.

Weitere Informationen: <https://manpages.debian.org/latest/dpkg/dpkg.html>.

- Installiere ein Paket:

```
dpkg -i {{pfad/zu/datei.deb}}
```

- Entferne ein Paket:

```
dpkg -r {{paketname}}
```

- Liste installierte Pakete auf:

```
dpkg -l {{muster}}
```

- Liste die Inhalte eines Pakets auf:

```
dpkg -L {{paketname}}
```

- Liste die Inhalte einer lokalen Paketdatei auf:

```
dpkg -c {{pfad/zu/datei.deb}}
```

- Finde heraus welche Pakete eine Datei besitzen:

```
dpkg -S {{dateiname}}
```

# esa snap

Sentinel Application Platform (SNAP) für die Prozessierung von Satellitendaten der Europäischen Raumfahrtagentur (ESA).

Weitere Informationen: <http://step.esa.int/main/download/snap-download/>.

- Zeige alle Updates an:

```
snap --nosplash --nogui --modules --list --refresh
```

- Zeige Hilfe an:

```
snap --help
```

# httpie

Ein benutzerfreundliches HTTP-Tool.

Weitere Informationen: <https://github.com/httpie/httpie>.

- Sende eine GET-Anfrage (Standardmethode ohne Anfragedaten):

```
http {{https://example.com}}
```

- Sende eine POST-Anfrage (Standardmethode mit Anfragedaten):

```
http {{https://example.com}} {{hello=World}}
```

- Sende eine POST-Anfrage mit einer Datei als Eingabe:

```
http {{https://example.com}} < {{file.json}}
```

- Sende eine PUT-Anfrage mit einem bestimmten JSON-Body:

```
http PUT {{https://example.com/todos/7}} {{hello=world}}
```

- Sende eine DELETE-Anfrage mit einem bestimmten Anfrage-Header:

```
http DELETE {{https://example.com/todos/7}} {{API-Key:foo}}
```

- Zeige den gesamten HTTP-Austausch (sowohl Anfrage als auch Antwort):

```
http -v {{https://example.com}}
```

- Lade eine Datei herunter:

```
http --download {{https://example.com}}
```

- Folge Umleitungen und zeige Zwischenanfragen und -antworten:

```
http --follow --all {{https://example.com}}
```

# ip address

IP Adressen Management Unterbefehl.

Weitere Informationen: <https://manned.org/ip-address>.

- Zeige Netzwerk-Interfaces mit ihren Adressen:

```
ip address
```

- Zeige nur die aktiven Netzwerk-Interfaces:

```
ip address show up
```

- Zeige Informationen über ein bestimmtes Interface:

```
ip address show dev {{eth0}}
```

- Füge eine Adresse zu einem Interface hinzu:

```
ip address add {{ip_adresse}} dev {{eth0}}
```

- Entferne eine Adresse von einem Interface:

```
ip address delete {{ip_adresse}} dev {{eth0}}
```

- Entfernt alle IP Adressen in einem speziellen Bereich von einem Interface:

```
ip address flush dev {{eth0}} scope {{global|host|link}}
```

# ip route list

Dieser Befehl ist ein Alias von **ip route show**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr ip-route-show
```



# ip

Zeige und manipulierte routing, Geräte, Policy routing und Tunnel.

Weitere Informationen: <https://www.manned.org/ip.8>.

- Zeige Interfaces mit detaillierten Informationen:

```
ip address
```

- Zeige Interfaces mit kurzen Netzwerkinformationen:

```
ip -brief address
```

- Zeige Interfaces mit kurzen link layer Informationen:

```
ip -brief link
```

- Zeige die Routing Tabelle:

```
ip route
```

- Zeige Nachbarn (ARP Tabelle):

```
ip neighbour
```

- Schalte ein bestimmtes Interface ein oder aus:

```
ip link set {{interface}} {{up|down}}
```

- Entferne oder füge eine IP zu einem Interface hinzu:

```
ip addr add/del {{ip}}/{{mask}} dev {{interface}}
```

- Füge eine Standard Route hinzu:

```
ip route add default via {{ip}} dev {{interface}}
```

# ipcalc

Einfache Operationen und Berechnungen mit IP-Adressen und Netzwerken durchführen.

Weitere Informationen: <https://manned.org/ipcalc>.

- Zeige Informationen über eine Adresse oder ein Netzwerk mit einer bestimmten Subnetzmaske an:

```
ipcalc {{1.2.3.4}} {{255.255.255.0}}
```

- Zeige Informationen über eine Adresse oder ein Netzwerk in CIDR-Notation an:

```
ipcalc {{1.2.3.4}}/{{24}}
```

- Zeige die Broadcast-Adresse einer Adresse oder eines Netzwerks an:

```
ipcalc -b {{1.2.3.4}}/{{30}}
```

- Zeige die Netzwerkadresse der angegebenen IP-Adresse und Netzmaske an:

```
ipcalc -n {{1.2.3.4}}/{{24}}
```

- Zeige geografische Informationen zu einer bestimmten IP-Adresse an:

```
ipcalc -g {{1.2.3.4}}
```

# jhead

Manipulation der Zeitstempel und EXIF Daten in Bilddateien.

Weitere Informationen: <https://www.sentex.net/~mwandel/jhead/usage.html>.

- Zeige alle EXIF Daten eines JPEG Bildes:

```
jhead {{pfad/zu/bild.jpg}}
```

- Setze die Dateizeit auf die EXIF Zeit (d.h. die Dateizeit wird geändert):

```
jhead -ft {{pfad/zu/bild.jpg}}
```

- Setze die EXIF Zeit auf die Dateizeit (d.h. die EXIF Zeit wird geändert):

```
jhead -dsft {{pfad/zu/bild.jpg}}
```

- Benenne alle Dateinamen der JPEG Bilder nach EXIF Datum/Zeit um (Format: Jahr\_Monat\_Tag-Stunde\_Minute\_Sekunde.jpg):

```
jhead -n%Y_%m_%d-%H_%M_%S *.jpg
```

- Rotiere alle Bilder im aktuellen Ordner verlustfrei (Basierend auf dem EXIF Orientierungstag):

```
jhead -autorot *.jpg
```

- Korrigiere den EXIF Zeitstempel (Format: Stunden:Minuten:Sekunden). Im Beispiel wird jedes Bild eine Stunde in die Vergangenheit geschoben (z.B. Zeitumstellung vergessen):

```
jhead -ta-1:00:00 *.jpg
```

- Entferne alle Metadaten einschließlich der Vorschaubilder aus der JPEG Datei:

```
jhead -purejpg {{pfad/zu/bild.jpg}}
```

# lastb

Zeigt eine Liste der zuletzt angemeldeten Benutzer an.

Weitere Informationen: <https://manned.org/lastb>.

- Zeige eine Liste aller zuletzt angemeldeten Benutzer an:

```
sudo lastb
```

- Zeige eine Liste aller zuletzt angemeldeten Benutzer seit einem bestimmten Zeitpunkt an:

```
sudo lastb --since {{YYYY-MM-DD}}
```

- Zeige eine Liste aller zuletzt angemeldeten Benutzer bis zu einem bestimmten Zeitpunkt an:

```
sudo lastb --until {{YYYY-MM-DD}}
```

- Zeige eine Liste aller angemeldeten Benutzer zu einem bestimmten Zeitpunkt an:

```
sudo lastb --present {{hh:mm}}
```

- Zeige eine Liste aller zuletzt angemeldeten Benutzer und übersetze die IP zu einem Hostnamen:

```
sudo lastb --dns
```

# lastlog

Zeigt den letzten Login aller Benutzer oder eines bestimmten Benutzers an.

Weitere Informationen: <https://manned.org/lastlog>.

- Zeige den letzten Login aller Benutzer an:

```
lastlog
```

- Zeige den lastlog Datensatz des angegebenen Benutzers an:

```
lastlog --user {{benutzername}}
```

- Zeige Datensätze älter als 7 Tage an:

```
lastlog --before 7
```

- Zeige Datensätze jünger als 3 Tage an:

```
lastlog --time 3
```

# ldconfig

Symbolische Verknüpfungen und Zwischenspeicher für Abhängigkeiten von gemeinsam genutzten Bibliotheken konfigurieren.

Weitere Informationen: <https://manned.org/ldconfig>.

- Aktualisiere symbolische Verknüpfungen und erstelle den Zwischenspeicher neu (wird normalerweise ausgeführt, wenn eine neue Bibliothek installiert wird):

```
sudo ldconfig
```

- Aktualisiere die symbolischen Verknüpfungen für ein bestimmtes Verzeichnis:

```
sudo ldconfig -n {{pfad/zu/verzeichnis}}
```

- Gib die Bibliotheken im Zwischenspeicher aus und prüfe ob eine bestimmte Bibliothek vorhanden ist:

```
ldconfig -p | grep {{bibliotheksname}}
```

# ldd

Zeigt Abhängigkeiten von dynamischen Bibliotheken an.

Weitere Informationen: <https://manned.org/ldd>.

- Zeige Abhängigkeiten von dynamischen Bibliotheken einer Binärdatei an:

```
ldd {{pfad/zu/binärdatei}}
```

- Zeige ungenutzte direkte Abhängigkeiten an:

```
ldd -u {{pfad/zu/binärdatei}}
```

# line

Liest eine einzelne Eingabezeile.

Weitere Informationen: <https://manned.org/line.1>.

- Lies eine Eingabezeile:

```
line
```



# locate

Zum schnellen Finden von Dateinamen.

Weitere Informationen: <https://manned.org/locate>.

- Suche nach Dateien in der Datenbank. Hinweis: Die Datenbank wird periodisch aktualisiert (für gewöhnlich täglich oder wöchentlich):

```
locate {{muster}}
```

- Suche nach Dateien mit dem exakten Dateinamen. (Ein Muster ohne Platzhalterzeichen wird als `*muster*` interpretiert):

```
locate '*/{{{dateiname}}}'
```

- Aktualisiere die Datenbank. Dies ist nötig, falls kürzlich hinzugefügte Dateien gefunden werden sollen:

```
sudo updatedb
```

# logsave

Speichert die Ausgabe eines Befehls in eine Logdatei.

Weitere Informationen: <https://manned.org/logsave>.

- Führe einen Befehl mit angegebenen Argumenten aus und speichere die Ausgabe in eine Logdatei:

```
logsave {{pfad/zu/logdatei}} {{befehl}}
```

- Übernimm die Eingabe der Standardeingabe und speichere diese in eine Logdatei:

```
logsave {{logdatei}} -
```

- Hänge die Ausgabe an eine Logdatei an, anstatt deren aktuellen Inhalt zu ersetzen:

```
logsave -a {{logfile}} {{befehl}}
```

- Zeige die ausführliche Ausgabe an:

```
logsave -v {{logfile}} {{befehl}}
```

# logwatch

Fasst viele verschiedene Logs für gängige Dienste (z.B. Apache, pam\_unix, sshd, usw.) in einem einzelnen Bericht zusammen.

Weitere Informationen: <https://manned.org/logwatch>.

- Analysiere Logs für einen Zeitraum mit einer bestimmten Detailtiefe:

```
logwatch --range {{yesterday|today|all|help}} --detail {{low|medium|others}}
```

- Beschränke den Bericht auf Informationen zu einem ausgewählten Dienst:

```
logwatch --range {{all}} --service {{apache|pam_unix|etc}}
```

# lsmod

Zeigt den Status von Linux-Kernel-Modulen an.

Siehe auch **modprobe**, welches Kernel-Module lädt.

Weitere Informationen: <https://manned.org/lsmod>.

- Liste alle aktuell geladenen Kernel-Module auf:

```
lsmod
```

# megadl

Dieser Befehl ist ein Alias von **megatools-dl**.

Weitere Informationen: <https://megatools.megous.com/man/megatools-dl.html>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr megatools-dl
```

# ncal

Dieser Befehl ist ein Alias von **cal**.

Weitere Informationen: <https://manned.org/ncal>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr cal
```

# nixos-container

Startet NixOS Container basierend auf Linux Containern.

Weitere Informationen: <https://nixos.org/manual/nixos/stable/#ch-containers>.

- Gibt eine Liste der gestarteten Container aus:

```
sudo nixos-container list
```

- Erstelle einen NixOS Container mit einer spezifischen Konfigurations-Datei:

```
sudo nixos-container create {{container_name}} --config-file  
{{nix_config_file_path}}
```

- Starte, stoppe, terminiere oder zerstöre den angegebenen Container:

```
sudo nixos-container {{start|stop|terminate|destroy|status}}  
{{container_name}}
```

- Führe ein Kommando in einem laufenden Container aus:

```
sudo nixos-container run {{container_name}} -- {{command}}  
{{command_arguments}}
```

- Aktualisiere eine Containerkonfiguration:

```
sudo $EDITOR /var/lib/container/{{container_name}}/etc/nixos/  
configuration.nix && sudo nixos-container update  
{{container_name}}
```

- Starte eine interaktive Shell innerhalb eines laufenden Containers:

```
sudo nixos-container root-login {{container_name}}
```

# nixos-option

Prüfe eine NixOS Konfiguration.

Weitere Informationen: <https://nixos.org/manual/nixos/stable/index.html#sec-modularity>.

- Liste alle Unterschlüssel eines angegebenen Options-Schlüssels:

```
nixos-option {{option_key}}
```

- Liste aktuelle Boot-Kernelmodule:

```
nixos-option boot.kernelModules
```

- Liste Authorisierungsschlüssel für einen spezifischen Benutzer:

```
nixos-option users.users.{{username}}.openssh.authorizedKeys.  
{{keyFiles|keys}}
```

- Liste alle Remote-Builder-Maschinen:

```
nixos-option nix.buildMachines
```

- Liste alle Unterschlüssel eines angegebenen Options-Schlüssels innerhalb einer angegebenen Konfigurations-Datei:

```
NIXOS_CONFIG={{path_to_configuration.nix}} nixos-option  
{{option_key}}
```

- Zeige rekursiv alle Werte eines Benutzers:

```
nixos-option -r users.users.{{user}}
```



# nixos-rebuild

Rekonfiguriere eine NixOS-Maschine.

Weitere Informationen: <https://nixos.org/nixos/manual/#sec-changing-config>.

- Erstelle und wechsele zu einer neuen Konfiguration und nutze diese künftig als Standardkonfiguration:

```
sudo nixos-rebuild switch
```

- Gib der neu erstellten Standardkonfiguration einen Namen:

```
sudo nixos-rebuild switch -p {{name}}
```

- Erstelle und wechsele zu einer neuen Konfiguration, nutze diese künftig als Standardkonfiguration und installiere Updates:

```
sudo nixos-rebuild switch --upgrade
```

- Setze Änderungen der Konfiguration zurück und wechsele zur vorhergehenden Konfiguration:

```
sudo nixos-rebuild switch --rollback
```

- Erstelle eine neue Konfiguration und starte diese zukünftig direkt ohne sofort zu wechseln:

```
sudo nixos-rebuild boot
```

- Erstelle und wechsele direkt zu einer neuen Konfiguration, ändere den Standard-Start-Eintrag nicht (dieses Kommando ist für Testzwecke gedacht):

```
sudo nixos-rebuild test
```

- Erstelle die Konfiguration und öffne diese in einer virtuellen Maschine:

```
sudo nixos-rebuild build-vm
```

# nologin

Alternative Shell, die verhindert, dass sich ein Benutzer einloggt.

Weitere Informationen: <https://manned.org/nologin.5>.

- Setze die Login-Shell eines Benutzers auf `nologin`, um zu verhindern, dass der Benutzer sich anmeldet:

```
chsh -s {{user}} nologin
```

- Passe die Nachricht für Benutzer mit Login-Shell `nologin` an:

```
echo "{{nachricht}}" > /etc/nologin.txt
```

# nordvpn

Kommandozeilen-Schnittstelle für NordVPN.

Weitere Informationen: <https://nordvpn.com/download/linux/>.

- Interaktiv bei einem NordVPN-Konto anmelden:

```
nordvpn login
```

- Zeige den Verbindungsstatus an:

```
nordvpn status
```

- Stelle eine Verbindung zum nächsten NordVPN-Server her:

```
nordvpn connect
```

- Liste alle verfügbaren Länder auf:

```
nordvpn countries
```

- Stelle eine Verbindung zu einem NordVPN-Server in einem bestimmten Land her:

```
nordvpn connect {{Germany}}
```

- Stelle eine Verbindung zu einem NordVPN-Server in einem bestimmten Land und einer bestimmten Stadt her:

```
nordvpn connect {{Germany}} {{Berlin}}
```

- Aktiviere die autoconnect-Option:

```
nordvpn set autoconnect on
```

# numlockx

Steuere den Num-Lock-Tasten-Status in X11-Sitzungen.

Weitere Informationen: <http://www.mike-devlin.com/linux/README-numlockx.htm>.

- Zeige den aktuellen Status der Num-Lock-Taste an:

```
numlockx status
```

- Schalte die Num-Lock-Taste ein:

```
numlockx on
```

- Schalte die Num-Lock-Taste aus:

```
numlockx off
```

- Schalte die Num-Lock-Taste um:

```
numlockx toggle
```

# pacman --database

Mit der Arch Linux Paketdatenbank arbeiten.

Verschiedene Attribute von installierten Paketen bearbeiten.

Siehe auch: **pacman**.

Weitere Informationen: <https://man.archlinux.org/man/pacman.8>.

- Markiere ein Paket als implizit installiert:

```
sudo pacman --database --asdeps {{paketname}}
```

- Markiere ein Paket als explizit installiert:

```
sudo pacman --database --asexplicit {{paketname}}
```

- Überprüfe, dass alle Paketabhängigkeiten installiert sind:

```
pacman --database --check
```

- Überprüfe in den Repositorien, dass alle angegebenen Abhängigkeiten verfügbar sind:

```
pacman --database --check --check
```

- Zeige nur Fehlermeldungen:

```
pacman --database --check --quiet
```

- Zeige Hilfe an:

```
pacman --database --help
```

# pacman --deptest

Überprüfe alle angegebenen Abhängigkeiten und gib eine Liste von Abhängigkeiten zurück, welche auf dem System nicht erfüllt sind.

Siehe auch: **pacman**.

Weitere Informationen: <https://man.archlinux.org/man/pacman.8>.

- Zeige Paketnamen von Abhängigkeiten an, welche nicht installiert sind:

```
pacman --deptest {{paket1 paket2 ...}}
```

- Überprüfe ob ein installiertes Paket eine Minimalversion erfüllt:

```
pacman --deptest "{{bash>=5}}"
```

- Überprüfe ob eine neuere version eines Paketes installiert ist:

```
pacman --deptest "{{bash>5}}"
```

- Zeige Hilfe an:

```
pacman --deptest --help
```

# pacman --files

Arch Linux Paketverwaltungs-Werkzeug.

Siehe auch: **pacman**, **pkgfile**.

Weitere Informationen: <https://man.archlinux.org/man/pacman.8>.

- Aktualisiere die Paketdatenbank:

```
sudo pacman --files --refresh
```

- Finde das Paket, welches eine bestimmte Datei besitzt:

```
pacman --files {{dateiname}}
```

- Finde das Paket, welches eine bestimmte Datei besitzt, mittels eines regulären Ausdrucks:

```
pacman --files --regex '{{suchmuster}}'
```

- Liste nur Paketnamen auf:

```
pacman --files --quiet {{dateiname}}
```

- Liste die Dateien auf welche einem bestimmten Paket gehören:

```
pacman --files --list {{paketname}}
```

- Zeige Hilfe an:

```
pacman --files --help
```

# pacman --query

Arch Linux Paketverwaltungs-Werkzeug.

Siehe auch: **pacman**.

Weitere Informationen: <https://man.archlinux.org/man/pacman.8>.

- Liste alle installierten Pakete und dessen Versionen auf:

```
pacman --query
```

- Liste alle ausdrücklich installierten Pakete und dessen Versionen auf:

```
pacman --query --explicit
```

- Finde heraus welches Paket eine Datei besitzt:

```
pacman --query --owns {{dateiname}}
```

- Zeige Informationen über ein installiertes Paket an:

```
pacman --query --info {{paketname}}
```

- Liste alle Dateien auf welche einem Paket gehören:

```
pacman --query --list {{paketname}}
```

- Liste verwaiste Pakete auf (Pakete welche als Abhängigkeit installiert wurden, aber von keinem Paket benötigt werden):

```
pacman --query --unrequired --deps --quiet
```

- Liste installierte Pakete auf welche nicht in den Repositorien gefunden werden können:

```
pacman --query --foreign
```

- Liste veraltete Pakete auf:

```
pacman --query --upgrades
```



# pacman --remove

Arch Linux Paketverwaltungs-Werkzeug.

Siehe auch: **pacman**.

Weitere Informationen: <https://man.archlinux.org/man/pacman.8>.

- Entferne ein Paket und dessen Abhängigkeiten:

```
sudo pacman --remove --recursive {{paketname}}
```

- Entferne ein Paket sowie alle Abhängigkeiten und Konfigurationsdateien:

```
sudo pacman --remove --recursive --nosave {{paketname}}
```

- Entferne ein Paket ohne Bestätigungsaufforderung:

```
sudo pacman --remove --noconfirm {{paketname}}
```

- Entferne verwaiste Pakete (Pakete welche als Abhängigkeit installiert wurden, aber von keinem Paket benötigt werden):

```
sudo pacman --remove --recursive --nosave $(pacman --query --unrequired --deps --quiet)
```

- Entferne ein Paket und alle Pakete die davon abhängig sind:

```
sudo pacman --remove --cascade {{paketname}}
```

- Liste Pakete auf, welche betroffen sein würden (entfernt keine Pakete):

```
pacman --remove --print {{paketname}}
```

- Zeige Hilfe für diesen Unterbefehl an:

```
pacman --remove --help
```

# pacman --sync

Arch Linux Paketverwaltungs-Werkzeug.

Siehe auch: **pacman**.

Weitere Informationen: <https://man.archlinux.org/man/pacman.8>.

- Installiere ein neues Paket:

```
sudo pacman --sync {{paketname}}
```

- Synchronisiere und aktualisiere alle Pakete (füge `--downloadonly` hinzu um die Pakete nur herunterzuladen und nicht zu aktualisieren):

```
sudo pacman --sync --refresh --sysupgrade
```

- Aktualisiere alle Pakete und installiere ein neues ohne Bestätigungsaufforderung:

```
sudo pacman --sync --refresh --sysupgrade --noconfirm  
{{paketname}}
```

- Suche in der Paketdatenbank mit einem regulären Ausdruck oder Schlüsselwort:

```
pacman --sync --search "{{suchmuster}}"
```

- Zeige Informationen über ein Paket an:

```
pacman --sync --info {{paketname}}
```

- Überschreibe widersprüchliche Dateien während einer Paketaktualisierung:

```
sudo pacman --sync --refresh --sysupgrade --overwrite {{pfad/  
zur/datei}}
```

- Synchronisiere und aktualisiere alle Pakete, ignoriere aber ein bestimmtes Paket (kann mehr als einmal angegeben werden):

```
sudo pacman --sync --refresh --sysupgrade --ignore  
{{paketname}}
```

- Entferne nicht installierte Pakete und ungenutzte Repositorys vom Cache (nutze zwei `--clean` Operationen um alle Pakete aufzuräumen):

```
sudo pacman --sync --clean
```

# pacman --upgrade

Arch Linux Paketverwaltungs-Werkzeug.

Siehe auch: **pacman**.

Weitere Informationen: <https://man.archlinux.org/man/pacman.8>.

- Installiere ein oder mehrere Pakete von Dateien:

```
sudo pacman --upgrade {{pfad/zu/paket1.pkg.tar.zst}} {{pfad/zu/paket2.pkg.tar.zst}}
```

- Installiere ein Paket ohne Bestätigungsaufforderung:

```
sudo pacman --upgrade --noconfirm {{pfad/zu/paket.pkg.tar.zst}}
```

- Überschreibe widersprüchliche Dateien während einer Paketinstallation:

```
sudo pacman --upgrade --overwrite {{pfad/zu/datei}} {{pfad/zu/paket.pkg.tar.zst}}
```

- Installiere ein Paket und überspringe die Überprüfung von Abhängigkeitsversionen:

```
sudo pacman --upgrade --nodeps {{pfad/zu/paket.pkg.tar.zst}}
```

- Liste Pakete auf, welche betroffen sein würden (installiert keine Pakete):

```
pacman --upgrade --print {{pfad/zu/paket.pkg.tar.zst}}
```

- Zeige Hilfe an:

```
pacman --upgrade --help
```

# pacman

Arch Linux Paket Management Tool.

Siehe auch: **pacman-database**, **pacman-deptest**, **pacman-files**, **pacman-key**, **pacman-mirrors**, **pacman-query**, **pacman-remove**, **pacman-sync**, **pacman-upgrade**.

Weitere Informationen: <https://man.archlinux.org/man/pacman.8>.

- Synchronisiere und aktualisiere alle Pakete:

```
sudo pacman -Syu
```

- Installiere ein neues Paket:

```
sudo pacman -S {{paket}}
```

- Entferne ein Paket und dessen Abhängigkeiten:

```
sudo pacman -Rs {{paket}}
```

- Liste alle installierten Pakete und dessen Versionen auf:

```
pacman -Q
```

- Liste alle ausdrücklich installierten Pakete und dessen Versionen auf:

```
pacman -Qe
```

- Zeige verwaiste Pakete an, welche als Abhängigkeiten installiert wurden, aber nicht mehr von anderen Paketen benötigt werden:

```
pacman -Qtdq
```

- Leere den gesamten pacman Cache:

```
sudo pacman -Scc
```

# paru

Ein AUR-Helfer und pacman-Wrapper.

Weitere Informationen: <https://github.com/Morganamilo/paru>.

- Interaktiv nach einem Paket suchen und es installieren:

```
paru {{paketname_oder_suchbegriff}}
```

- Alle Pakete synchronisieren und aktualisieren:

```
paru
```

- AUR-Pakete aktualisieren:

```
paru -Sua
```

- Informationen über ein Paket abrufen:

```
paru -Si {{paketname}}
```

- Herunterladen von PKGBUILD und anderen Paket-Quelldateien aus dem AUR oder dem ABS:

```
paru --getpkgbuild {{paketname}}
```

- Anzeigen der PKGBUILD-Datei eines Pakets:

```
paru --getpkgbuild --print {{paketname}}
```

# qm create

Erzeugung einer virtuellen Maschine per QEMU/KVM Virtual Machine Manager.

Weitere Informationen: <https://pve.proxmox.com/pve-docs/qm.1.html>.

- Erzeuge eine virtuelle Maschine:

```
qm create {{100}}
```

- Erzeuge eine virtuelle Maschine und starte sie unmittelbar danach im Anschluss:

```
qm create {{100}} --start 1
```

- Lege den Typ des Betriebssystems auf der virtuellen Maschine fest:

```
qm create {{100}} --ostype {{win10}}
```

- Ersetze eine bestehende virtuelle Maschine (setzt deren Archivierung voraus):

```
qm create {{100}} --archive {{pfad/zu/backup_file.tar}} --  
force 1
```

- Lege ein Skript fest, welches automatisch abhängig vom Zustand der virtuellen Maschine ausgelöst werden soll:

```
qm create {{100}} --hookscript {{pfad/zu/skript.pl}}
```

# qm start

Starte eine virtuelle Maschine per QEMU/KVM Virtual Machine Manager.

Weitere Informationen: <https://pve.proxmox.com/pve-docs/qm.1.html>.

- Starte eine bestimmte virtuelle Maschine:

```
qm start {{100}}
```

- Lege den QEMU Maschinentyp fest (etwa den zu emulierenden Prozessor):

```
qm start {{100}} --machine {{q35}}
```

- Starte eine bestimmte virtuelle Maschine mit einem Timeout (Zeitabschaltung) nach 60 Sekunden:

```
qm start {{100}} --timeout {{60}}
```

# qmrestore

Wiederherstellung von QemuServer vmdump Backups.

Weitere Informationen: <https://pve.proxmox.com/pve-docs/qmrestore.1.html>.

- Wiederherstellung einer virtuellen Maschine mittels Backupdatei auf dem ursprünglichen Speicher:

```
qmrestore {{pfad/zu/vmdump-qemu-100.vma.lzo}} {{100}}
```

- Überschreibung einer bestehenden virtuellen Maschine auf dem ursprünglichen Speicher:

```
qmrestore {{pfad/zu/vmdump-qemu-100.vma.lzo}} {{100}} --force  
true
```

- Wiederherstellung einer virtuellen Maschine auf einem bestimmten Speicher:

```
qmrestore {{pfad/zu/vmdump-qemu-100.vma.lzo}} {{100}} --  
storage {{local}}
```

- Sofortiger Start einer virtuellen Maschine bei gleichzeitiger Wiederherstellung im Hintergrund (nur bei Proxmox Backup Server):

```
qmrestore {{pfad/zu/vmdump-qemu-100.vma.lzo}} {{100}} --live-  
restore true
```



# see

Alias für die Ansicht von **run-mailcap**.

Ein Alias für die Aktion print eines **run-mailcap**.

Weitere Informationen: <https://manned.org/see>.

- Die Aktion see kann verwendet werden, um eine beliebige Datei (in der Regel ein Bild) im Standard-Mailcap-Explorer anzuzeigen:

```
see {{dateiname}}
```

- Verwenden mit run-mailcap:

```
run-mailcap --action=view {{dateiname}}
```

# sysctl

Laufzeit-Kernelparameter auflisten und ändern.

Weitere Informationen: <https://manned.org/sysctl.8>.

- Liste alle verfügbaren Kernelparameter mit ihren Werten auf:

```
sysctl -a
```

- Setze einen veränderbaren Kernelparameter:

```
sysctl -w {{sektion.tunable}}={{wert}}
```

- Frage aktuell geöffnete Datei-Handler ab:

```
sysctl fs.file-nr
```

- Frage die maximale Anzahl geöffneter Dateien ab:

```
sysctl fs.file-max
```

- Übernimm Änderungen aus der `/etc/sysctl.conf` Datei:

```
sysctl -p
```

# ubuntu-bug

Dieser Befehl ist ein Alias von **apport-bug**.

Weitere Informationen: <https://manned.org/ubuntu-bug>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr apport-bug
```

# watch

Führe einen Befehl wiederholt aus und überwache die Ausgabe im Vollbildmodus.

Weitere Informationen: <https://manned.org/watch>.

- Überwache die Dateien im aktuellen Verzeichnis:

```
watch {{ls}}
```

- Überwache verfügbaren Festplatten-Speicherplatz und hebe die Änderungen hervor:

```
watch -d {{df}}
```

- Überwache "node"-Prozesse und aktualisiere alle 3 Sekunden:

```
watch -n {{3}} "{{ps aux | grep node}}"
```

- Überwache, ob sich der Festplatten-Speicherplatz ändert und höre mit der Überwachung auf, sobald eine Änderung auftritt:

```
watch -g {{df}}
```

# xrandr

Setzt die Auflösung, Orientierung und/oder Reflektion eines Bildschirmausgangs.

Weitere Informationen: <https://www.x.org/releases/current/doc/man/man1/xrandr.1.xhtml>.

- Zeige den momentanen Systemzustand an (erkannte Bildschirme, Auflösungen, ...):

```
xrandr --query
```

- Deaktiviere nicht mehr verbundene Ausgangsgeräte und aktiviere verbundene Ausgänge mit Standardeinstellungen:

```
xrandr --auto
```

- Ändere die Auflösung und Bildfrequenz von DisplayPort 1 zu 1920x1080, 60Hz:

```
xrandr --output {{DP1}} --mode {{1920x1080}} --rate {{60}}
```

- Setze die Auflösung von HDMI auf 1280x1024 und platziere HDMI1 rechts von DP1:

```
xrandr --output {{HDMI2}} --mode {{1280x1024}} --right-of {{DP1}}
```

- Deaktiviere den Ausgang von VGA1:

```
xrandr --output {{VGA1}} --off
```

- Setze die Bildschirmhelligkeit von LVDS1 auf 50%:

```
xrandr --output {{LVDS1}} --brightness {{0.5}}
```

# yay

Yet Another Yogurt: Ein Programm für Arch Linux um Pakete vom Arch User Repository zu installieren.

Siehe auch **pacman**.

Weitere Informationen: <https://github.com/jguer/yay>.

- Suche und installiere Pakete von den Repositorys und dem AUR interaktiv:

```
yay {{paketname|suchbegriff}}
```

- Synchronisiere und aktualisiere alle Pakete von den Repositorys und dem AUR:

```
yay
```

- Synchronisiere und aktualisiere nur AUR-Pakete:

```
yay -Sua
```

- Installiere ein neues Paket von den Repositorys und dem AUR:

```
yay -S {{paketname}}
```

- Entferne ein Paket sowie alle Abhängigkeiten und Konfigurationsdateien:

```
yay -Rns {{paketname}}
```

- Suche in der Paketdatenbank nach einem Schlüsselwort in den Repositorys und dem AUR:

```
yay -Ss {{schlüsselwort}}
```

- Entferne verwaiste Pakete (als Abhängigkeit installiert, aber von keinem Paket benötigt):

```
yay -Yc
```

- Zeige Statistiken für installierte Pakete sowie die Gesundheit des Systems an:

```
yay -Ps
```

# zathura

Ein vim-artiger, modaler Dokumentenbetrachter mit integrierter Kommandozeile.

Benötigt ein installiertes Backend (poppler, Postscript oder DjVu).

Weitere Informationen: <https://pwmt.org/projects/zathura/>.

- Öffne eine Datei:

```
zathura {{pfad/zur/datei}}
```

- Navigiere nach links/oben/unten/rechts:

```
{{H|J|K|L|Pfeiltasten}}
```

- Rotiere:

```
r
```

- Invertiere die Farben:

```
<Strg> + R
```

- Durchsuche den Text nach einem gegebenen String:

```
/{{string}}
```

- Erstelle/lösche Lesezeichen:

```
::{{bmark|bdelete}} {{name_des_lesezeichens}}
```

- Liste alle Lesezeichen auf:

```
:blist
```

# zypper

SUSE & openSUSE Package-Management-Werkzeug.

Weitere Informationen: [https://en.opensuse.org/SDB:Zypper\\_manual](https://en.opensuse.org/SDB:Zypper_manual).

- Synchronisiere die Liste von Paketen und verfügbaren Versionen:

```
zypper refresh
```

- Installiere ein neues Paket:

```
zypper install {{paket}}
```

- Entferne ein Paket:

```
zypper remove {{paket}}
```

- Aktualisiere installierte Pakete zur neuesten verfügbaren Version:

```
zypper update
```

- Suche Paket nach einem bestimmten Schema:

```
zypper search {{schema}}
```

- Zeige Informationen bezüglich der konfigurierten Repositories:

```
zypper repos --sort-by-priority
```



Osx

# aa

Dieser Befehl ist ein Alias von **yaa**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr yaa
```

# caffeinate

Hindert den Mac daran, in den Schlaf-Modus zu gehen.

Weitere Informationen: <https://keith.github.io/xcode-man-pages/caffeinate.8.html>.

- Halte den Mac für 1 Stunde (3600 Sekunden) wach:

```
caffeinate -u -t {{3600}}
```

- Halte den Mac wach, bis ein bestimmter Befehl abgeschlossen ist:

```
caffeinate -s {{befehl}}
```

- Halte den Mac wach, bis `caffeinate` durch Cmd-C beendet wird:

```
caffeinate -i
```

# diskutil

Dienstprogramm zur Verwaltung lokaler Festplatten und Volumes.

Weitere Informationen: <https://keith.github.io/xcode-man-pages/diskutil.8.html>.

- Auflisten aller aktuell verfügbaren Festplatten, Partitionen und gemounteten Volumes:

```
diskutil list
```

- Reparieren der Dateisystem-Datenstrukturen eines Volumes:

```
diskutil repairVolume {{/dev/diskX}}
```

- Einen Datenträger aushängen:

```
diskutil unmountDisk {{/dev/diskX}}
```

- Eine CD/DVD auswerfen (zuerst aushängen):

```
diskutil eject {{/dev/disk1}}
```

# g[

Dieser Befehl ist ein Alias von **-p linux** [.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux [
```

# gawk

Dieser Befehl ist ein Alias von **-p linux awk**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux awk
```

# gb2sum

Dieser Befehl ist ein Alias von **-p linux b2sum**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux b2sum
```

# gbase32

Dieser Befehl ist ein Alias von **-p linux base32**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux base32
```



# gbase64

Dieser Befehl ist ein Alias von **-p linux base64**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux base64
```

# gbasename

Dieser Befehl ist ein Alias von **-p linux basename**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux basename
```

# gbasenc

Dieser Befehl ist ein Alias von **-p linux basenc**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux basenc
```

# gcat

Dieser Befehl ist ein Alias von **-p linux cat**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux cat
```

# gchcon

Dieser Befehl ist ein Alias von **-p linux chcon**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux chcon
```

# gchgrp

Dieser Befehl ist ein Alias von **-p linux chgrp**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux chgrp
```

# gchmod

Dieser Befehl ist ein Alias von **-p linux chmod**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux chmod
```

# gchown

Dieser Befehl ist ein Alias von **-p linux chown**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux chown
```



# gchroot

Dieser Befehl ist ein Alias von **-p linux chroot**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux chroot
```

# gcksum

Dieser Befehl ist ein Alias von **-p linux cksum**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux cksum
```

# gcomm

Dieser Befehl ist ein Alias von **-p linux comm**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux comm
```

# gcp

Dieser Befehl ist ein Alias von **-p linux cp**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux cp
```

# gcsplit

Dieser Befehl ist ein Alias von **-p linux csplit**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux csplit
```

# gcut

Dieser Befehl ist ein Alias von **-p linux cut**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux cut
```

# gdate

Dieser Befehl ist ein Alias von **-p linux date**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux date
```

# gdd

Dieser Befehl ist ein Alias von **-p linux dd**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux dd
```



# gdf

Dieser Befehl ist ein Alias von **-p linux df**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux df
```

# gdir

Dieser Befehl ist ein Alias von **-p linux dir**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux dir
```

# gdircolors

Dieser Befehl ist ein Alias von **-p linux dircolors**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux dircolors
```

# gdirname

Dieser Befehl ist ein Alias von **-p linux dirname**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux dirname
```

# gdnsdomainname

Dieser Befehl ist ein Alias von **-p linux dnsdomainname**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux dnsdomainname
```

# gecho

Dieser Befehl ist ein Alias von **-p linux echo**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux echo
```

# ged

Dieser Befehl ist ein Alias von **-p linux ed**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux ed
```

# gegrep

Dieser Befehl ist ein Alias von **-p linux egrep**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux egrep
```



# genv

Dieser Befehl ist ein Alias von **-p linux env**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux env
```

# gexpand

Dieser Befehl ist ein Alias von **-p linux expand**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux expand
```

# gexpr

Dieser Befehl ist ein Alias von **-p linux expr**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux expr
```

# gfactor

Dieser Befehl ist ein Alias von **-p linux factor**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux factor
```

# gfalse

Dieser Befehl ist ein Alias von **-p linux false**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux false
```

# gfgrep

Dieser Befehl ist ein Alias von **-p linux fgrep**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux fgrep
```

# gfind

Dieser Befehl ist ein Alias von **-p linux find**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux find
```

# gfmt

Dieser Befehl ist ein Alias von **-p linux fmt**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux fmt
```



# gfold

Dieser Befehl ist ein Alias von **-p linux fold**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux fold
```

# gftp

Dieser Befehl ist ein Alias von **-p linux ftp**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux ftp
```

# ggrep

Dieser Befehl ist ein Alias von **-p linux grep**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux grep
```

# ggroups

Dieser Befehl ist ein Alias von **-p linux groups**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux groups
```

# ghead

Dieser Befehl ist ein Alias von **-p linux head**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux head
```

# ghostid

Dieser Befehl ist ein Alias von **-p linux ghostid**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux ghostid
```

# ghostname

Dieser Befehl ist ein Alias von **-p linux hostname**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux hostname
```

# gid

Dieser Befehl ist ein Alias von **-p linux id**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux id
```



# gifconfig

Dieser Befehl ist ein Alias von **-p linux ifconfig**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux ifconfig
```

# gindent

Dieser Befehl ist ein Alias von **-p linux indent**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux indent
```

# ginstall

Dieser Befehl ist ein Alias von **-p linux install**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux install
```

# gjoin

Dieser Befehl ist ein Alias von **-p linux join**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux join
```

# gkill

Dieser Befehl ist ein Alias von **-p linux kill**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux kill
```

# glibtool

Dieser Befehl ist ein Alias von **-p linux libtool**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux libtool
```

# glibtoolize

Dieser Befehl ist ein Alias von **-p linux libtoolize**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux libtoolize
```

# glink

Dieser Befehl ist ein Alias von **-p linux link**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux link
```



# gln

Dieser Befehl ist ein Alias von **-p linux ln**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux ln
```

# glocate

Dieser Befehl ist ein Alias von **-p linux locate**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux locate
```

# glogger

Dieser Befehl ist ein Alias von **-p linux logger**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux logger
```

# glogname

Dieser Befehl ist ein Alias von **-p linux logname**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux logname
```

# gls

Dieser Befehl ist ein Alias von **-p linux ls**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux ls
```

# gmake

Dieser Befehl ist ein Alias von **-p linux make**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux make
```

# gmd5sum

Dieser Befehl ist ein Alias von **-p linux md5sum**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux md5sum
```

# gmkdir

Dieser Befehl ist ein Alias von **-p linux mkdir**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux mkdir
```



# gmkfifo

Dieser Befehl ist ein Alias von **-p linux mkfifo**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux mkfifo
```

# gmknod

Dieser Befehl ist ein Alias von **-p linux mknod**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux mknod
```

# gmkttemp

Dieser Befehl ist ein Alias von **-p linux mktemp**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux mktemp
```

# gmv

Dieser Befehl ist ein Alias von **-p linux mv**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux mv
```

# gnice

Dieser Befehl ist ein Alias von **-p linux nice**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux nice
```

# gnl

Dieser Befehl ist ein Alias von **-p linux nl**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux nl
```

# gnohup

Dieser Befehl ist ein Alias von **-p linux nohup**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux nohup
```

# gnproc

Dieser Befehl ist ein Alias von **-p linux nproc**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux nproc
```



# gnumfmt

Dieser Befehl ist ein Alias von **-p linux numfmt**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux numfmt
```

# god

Dieser Befehl ist ein Alias von **-p linux od**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux od
```

# gpaste

Dieser Befehl ist ein Alias von **-p linux paste**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux paste
```

# gpathchk

Dieser Befehl ist ein Alias von **-p linux pathchk**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux pathchk
```

# gping

Dieser Befehl ist ein Alias von **-p linux ping**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux ping
```

# gping6

Dieser Befehl ist ein Alias von **-p linux ping6**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux ping6
```

# gpinky

Dieser Befehl ist ein Alias von **-p linux pinky**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux pinky
```

# gpr

Dieser Befehl ist ein Alias von **-p linux pr**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux pr
```



# gprintenv

Dieser Befehl ist ein Alias von **-p linux printenv**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux printenv
```

# gprintf

Dieser Befehl ist ein Alias von **-p linux printf**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux printf
```

# gptx

Dieser Befehl ist ein Alias von **-p linux ptx**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux ptx
```

# gpwd

Dieser Befehl ist ein Alias von **-p linux pwd**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux pwd
```

# grcp

Dieser Befehl ist ein Alias von **-p linux rcp**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux rcp
```

# greadlink

Dieser Befehl ist ein Alias von **-p linux readlink**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux readlink
```

# grealpath

Dieser Befehl ist ein Alias von **-p linux realpath**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux realpath
```

# grexec

Dieser Befehl ist ein Alias von **-p linux rexec**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux rexec
```



# grlogin

Dieser Befehl ist ein Alias von **-p linux rlogin**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux rlogin
```

# grm

Dieser Befehl ist ein Alias von **-p linux rm**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux rm
```

# grmdir

Dieser Befehl ist ein Alias von **-p linux rmdir**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux rmdir
```

# grsh

Dieser Befehl ist ein Alias von **-p linux rsh**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux rsh
```

# gruncon

Dieser Befehl ist ein Alias von **-p linux runcon**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux runcon
```

# gsed

Dieser Befehl ist ein Alias von **-p linux sed**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux sed
```

# gseq

Dieser Befehl ist ein Alias von **-p linux seq**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux seq
```

# gsha1sum

Dieser Befehl ist ein Alias von **-p linux sha1sum**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux sha1sum
```



# gsha224sum

Dieser Befehl ist ein Alias von **-p linux sha224sum**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux sha224sum
```

# gsha256sum

Dieser Befehl ist ein Alias von **-p linux sha256sum**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux sha256sum
```

# gsha384sum

Dieser Befehl ist ein Alias von **-p linux sha384sum**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux sha384sum
```

# gsha512sum

Dieser Befehl ist ein Alias von **-p linux sha512sum**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux sha512sum
```

# gshred

Dieser Befehl ist ein Alias von **-p linux shred**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux shred
```

# gshuf

Dieser Befehl ist ein Alias von **-p linux shuf**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux shuf
```

# gsleep

Dieser Befehl ist ein Alias von **-p linux sleep**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux sleep
```

# gsort

Dieser Befehl ist ein Alias von **-p linux sort**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux sort
```



# gsplit

Dieser Befehl ist ein Alias von **-p linux split**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux split
```

# gstat

Dieser Befehl ist ein Alias von **-p linux stat**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux stat
```

# gstdbuf

Dieser Befehl ist ein Alias von **-p linux stdbuf**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux stdbuf
```

# gstty

Dieser Befehl ist ein Alias von **-p linux stty**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux stty
```

# gsum

Dieser Befehl ist ein Alias von **-p linux sum**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux sum
```

# gsync

Dieser Befehl ist ein Alias von **-p linux sync**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux sync
```

# gtac

Dieser Befehl ist ein Alias von **-p linux tac**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux tac
```

# gtail

Dieser Befehl ist ein Alias von **-p linux tail**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux tail
```



# gtalk

Dieser Befehl ist ein Alias von **-p linux talk**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux talk
```

# gtar

Dieser Befehl ist ein Alias von **-p linux tar**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux tar
```

# gtee

Dieser Befehl ist ein Alias von **-p linux tee**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux tee
```

# gtelnet

Dieser Befehl ist ein Alias von **-p linux telnet**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux telnet
```

# gtest

Dieser Befehl ist ein Alias von **-p linux test**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux test
```

# gtftp

Dieser Befehl ist ein Alias von **-p linux tftp**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux tftp
```

# gtime

Dieser Befehl ist ein Alias von **-p linux time**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux time
```

# gtimeout

Dieser Befehl ist ein Alias von **-p linux timeout**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux timeout
```



# gtouch

Dieser Befehl ist ein Alias von **-p linux touch**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux touch
```

# gtr

Dieser Befehl ist ein Alias von **-p linux tr**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux tr
```

# gtracroute

Dieser Befehl ist ein Alias von **-p linux traceroute**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux traceroute
```

# gtrue

Dieser Befehl ist ein Alias von **-p linux true**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux true
```

# gtruncate

Dieser Befehl ist ein Alias von **-p linux truncate**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux truncate
```

# gtsort

Dieser Befehl ist ein Alias von **-p linux tsort**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux tsort
```

# gtty

Dieser Befehl ist ein Alias von **-p linux tty**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux tty
```

# guname

Dieser Befehl ist ein Alias von **-p linux uname**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux uname
```



# gunexpand

Dieser Befehl ist ein Alias von **-p linux unexpand**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux unexpand
```

# guniq

Dieser Befehl ist ein Alias von **-p linux uniq**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux uniq
```

# gunits

Dieser Befehl ist ein Alias von **-p linux units**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux units
```

# gunlink

Dieser Befehl ist ein Alias von **-p linux unlink**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux unlink
```

# gupdatedb

Dieser Befehl ist ein Alias von **-p linux updatedb**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux updatedb
```

# guptime

Dieser Befehl ist ein Alias von **-p linux uptime**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux uptime
```

# gusers

Dieser Befehl ist ein Alias von **-p linux users**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux users
```

# gvdir

Dieser Befehl ist ein Alias von **-p linux vdir**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux vdir
```



# gwc

Dieser Befehl ist ein Alias von **-p linux wc**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux wc
```

# gwhich

Dieser Befehl ist ein Alias von **-p linux which**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux which
```

# gwho

Dieser Befehl ist ein Alias von **-p linux who**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux who
```

# gwhoami

Dieser Befehl ist ein Alias von **-p linux whoami**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux whoami
```

# gwhois

Dieser Befehl ist ein Alias von **-p linux whois**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux whois
```

# gxargs

Dieser Befehl ist ein Alias von **-p linux xargs**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux xargs
```

# gyes

Dieser Befehl ist ein Alias von **-p linux yes**.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr -p linux yes
```

# httpie

Ein benutzerfreundliches HTTP-Tool.

Weitere Informationen: <https://github.com/httpie/httpie>.

- Sende eine GET-Anfrage (Standardmethode ohne Anfragedaten):

```
http {{https://example.com}}
```

- Sende eine POST-Anfrage (Standardmethode mit Anfragedaten):

```
http {{https://example.com}} {{hello=World}}
```

- Sende eine POST-Anfrage mit einer Datei als Eingabe:

```
http {{https://example.com}} < {{file.json}}
```

- Sende eine PUT-Anfrage mit einem bestimmten JSON-Body:

```
http PUT {{https://example.com/todos/7}} {{hello=world}}
```

- Sende eine DELETE-Anfrage mit einem bestimmten Anfrage-Header:

```
http DELETE {{https://example.com/todos/7}} {{API-Key:foo}}
```

- Zeige den gesamten HTTP-Austausch (sowohl Anfrage als auch Antwort):

```
http -v {{https://example.com}}
```

- Lade eine Datei herunter:

```
http --download {{https://example.com}}
```

- Folge Umleitungen und zeige Zwischenanfragen und -antworten:

```
http --follow --all {{https://example.com}}
```



# launchd

Dieser Befehl ist ein Alias von **launchctl**.

Weitere Informationen: <https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/BPSystemStartup/Chapters/Introduction.html>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr launchctl
```

# md5

Berechne kryptografische MD5-Prüfsummen.

Weitere Informationen: <https://keith.github.io/xcode-man-pages/md5.1.html>.

- Berechne die MD5-Prüfsumme einer Datei und gib sie mit Dateinamen auf der Konsole aus:

```
md5 {{pfad/zu/datei}}
```

- Berechne die MD5-Prüfsumme mehrerer Dateien und gib sie mit Dateinamen auf der Konsole aus:

```
md5 {{pfad/zu/datei1 pfad/zu/datei2 ...}}
```

- Gib nur die MD5-Prüfsumme auf der Konsole aus:

```
md5 -q {{pfad/zu/datei}}
```

- Berechne die MD5-Prüfsumme für den gegebenen Text:

```
md5 -s "{{text}}"
```

# open

Öffne Dateien, Verzeichnisse und Anwendungen.

Weitere Informationen: <https://keith.github.io/xcode-man-pages/open.1.html>.

- Öffne eine Datei in der zugehörigen Anwendung:

```
open {{pfad/zu/datei}}
```

- Führe eine grafische macOS-Anwendung aus:

```
open -a {{anwendung}}
```

- Führe eine grafische macOS-Anwendung basierend auf der Bundle-Kennung aus (siehe `osascript` für eine einfache Möglichkeit, diese zu identifizieren):

```
open -b {{com.domain.anwendung}}
```

- Öffne das aktuelle Verzeichnis im Finder:

```
open .
```

- Zeige eine Datei im Finder an:

```
open -R {{pfad/zu/datei}}
```

- Öffne alle Dateien einer bestimmten Erweiterung im aktuellen Verzeichnis mit der zugehörigen Anwendung:

```
open {{*.ext}}
```

# pbpaste

Sende den Inhalt der Zwischenablage zum Standardoutput.

Weitere Informationen: <https://keith.github.io/xcode-man-pages/pbpaste.1.html>.

- Schreibe den Inhalt der Zwischenablage in eine Datei:

```
pbpaste > {{datei}}
```

- Benutze die Zwischenablage als Eingabe für andere Kommandos:

```
pbpaste | grep foo
```

# tmutil

Dienstprogramm zum Verwalten von Time Machine-Backups. Die meisten Befehle erfordern Root-Rechte.

Weitere Informationen: <https://keith.github.io/xcode-man-pages/tmutil.8.html>.

- Setze ein HFS+ Laufwerk als Backupziel:

```
sudo tmutil setdestination {{pfad/zu/einhänge_punkt}}
```

- Setze eine APF-Freigabe oder SMB-Freigabe als Backupziel:

```
sudo tmutil setdestination "{{protocol://  
benutzer[:password]@host/share}}"
```

- Hänge das angegebene Ziel an die Liste der Backupziele an:

```
sudo tmutil setdestination -a {{ziel}}
```

- Aktiviere automatische Backups:

```
sudo tmutil enable
```

- Deaktiviere automatische Backups:

```
sudo tmutil disable
```

- Starte ein Backup im Hintergrund, falls nicht bereits eines läuft:

```
sudo tmutil startbackup
```

- Starte ein Backup im Vordergrund, falls nicht bereits eines läuft:

```
sudo tmutil startbackup -b
```

- Stoppe ein laufendes Backup:

```
sudo tmutil stopbackup
```

# xcode-select

Wechsel zwischen verschiedenen Xcode Versionen und den enthaltenen Entwicklertools.

Wird auch verwendet, um den Pfad zu Xcode zu aktualisieren, wenn dieser sich nach einer Installation geändert hat.

Weitere Informationen: <https://developer.apple.com/library/archive/technotes/tn2339/index.html>.

- Installiere die Xcode Entwicklertools:

```
xcode-select --install
```

- Wähle einen bestimmten Pfad als aktives Entwicklerverzeichnis aus:

```
xcode-select --switch {{pfad/zu/Xcode.app/Contents/  
Developer}}
```

- Wähle eine Xcode Version aus und ändere das aktive Entwicklerverzeichnis dahin:

```
xcode-select --switch {{pfad/zu/Xcode.app}}
```

- Gib das derzeit aktive Entwicklerverzeichnis aus:

```
xcode-select --print-path
```

- Verwerfe alle vom Benutzer angegebenen Entwicklerverzeichnisse (fortan wird der Standardsuchmechanismus verwendet, um diese zu finden):

```
sudo xcode-select --reset
```

Windows

# cd

Zeige den Namen des aktuellen Arbeitsverzeichnisses an oder ändere dieses.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/cd>.

- Wechsle zu einem Verzeichnis im selben Laufwerk:

```
cd {{pfad/zu/verzeichnis}}
```

- Zeige den Namen des aktuellen Verzeichnisses an:

```
cd
```

- Wechsle zum übergeordneten Verzeichnis des aktuellen Verzeichnisses:

```
cd ..
```

- Wechsle in ein Verzeichnis auf einem anderen Laufwerk:

```
cd {{pfad/zu/verzeichnis}} /d
```



# choco apikey

Verwalte die API-Schlüssel für die Quellen von Chocolatey.

Weitere Informationen: <https://chocolatey.org/docs/commands-apikey>.

- Gib eine Liste von Quellen und ihren API-Schlüsseln aus:

```
choco apikey
```

- Zeige eine bestimmte Quelle und ihren API-Schlüssel an:

```
choco apikey --source "{{quell_url}}"
```

- Setze den API-Schlüssel für eine Quelle:

```
choco apikey --source "{{quell_url}}" --key  
"{{api_schluessel}}"
```

- Entferne den API-Schlüssel einer Quelle:

```
choco apikey --source "{{quell_url}}" --remove
```

# choco feature

Interagiere mit Funktionen, die das Verhalten von Chocolatey verändern.

Weitere Informationen: <https://chocolatey.org/docs/commands-feature>.

- Gib eine Liste von verfügbaren Funktionen aus:

```
choco feature list
```

- Aktiviere eine Funktion:

```
choco feature enable --name {{name}}
```

- Deaktiviere eine Funktion:

```
choco feature disable --name {{name}}
```

# choco info

Zeige ausführliche Informationen über ein Chocolatey-Paket an.

Weitere Informationen: <https://chocolatey.org/docs/commands-info>.

- Zeige Informationen über ein bestimmtes Paket an:

```
choco info {{paket}}
```

- Zeige Informationen über ein bestimmtes lokales Paket an:

```
choco info {{paket}} --local-only
```

- Gib eine eigene Quelle an, von der Paket-Informationen abgerufen werden:

```
choco info {{paket}} --source {{quell_url|alias}}
```

- Gib einen Benutzernamen und ein Passwort für die Authentifizierung an:

```
choco info {{paket}} --user {{benutzername}} --password  
{{passwort}}
```

# choco install

Installiere ein oder mehrere Pakete mit Chocolatey.

Weitere Informationen: <https://chocolatey.org/docs/commands-install>.

- Installiere ein oder mehrere Pakete, deren Namen mit Leerzeichen getrennt sind:

```
choco install {{paket1 paket2 ...}}
```

- Installiere Pakete aus einer Konfigurationsdatei:

```
choco install {{pfad/zu/pakete.config}}
```

- Installiere Pakete aus einer nuspec- oder nupkg-Datei:

```
choco install {{pfad/zu/datei}}
```

- Installiere eine bestimmte Version eines Pakets:

```
choco install {{paket}} --version {{version}}
```

- Erlaube die gleichzeitige Installation mehrerer Versionen eines Pakets:

```
choco install {{paket}} --allow-multiple
```

- Stimme allen Fragen automatisch zu:

```
choco install {{paket}} --yes
```

- Gib eine eigene Quelle an, von der Paket-Informationen abgerufen werden:

```
choco install {{paket}} --source {{quell_url|alias}}
```

- Gib einen Benutzernamen und ein Passwort für die Authentifizierung an:

```
choco install {{paket}} --user {{benutzername}} --password {{password}}
```

# choco list

Zeige mit Chocolatey eine Liste von Paketen an.

Weitere Informationen: <https://chocolatey.org/docs/commands-list>.

- Zeige alle verfügbaren Pakete an:

```
choco list
```

- Zeige alle lokal installierten Pakete an:

```
choco list --local-only
```

- Zeige eine Liste einschließlich der lokalen Windows-Programme an:

```
choco list --include-programs
```

- Zeige nur zugelassene Pakete an:

```
choco list --approved-only
```

- Gib eine eigene Quelle an, von der Paket-Informationen abgerufen werden:

```
choco list --source {{quell_url|alias}}
```

- Gib einen Benutzernamen und ein Passwort für die Authentifizierung an:

```
choco list --user {{benutzername}} --password {{passwort}}
```

# choco new

Erstelle neue Paket-Beschreibungs-Dateien mit Chocolatey.

Weitere Informationen: <https://chocolatey.org/docs/commands-new>.

- Erstelle ein neues Grundgerüst für ein Paket:

```
choco new {{paket_name}}
```

- Erstelle ein neues Paket mit einer bestimmten Version:

```
choco new {{paket_name}} --version {{version}}
```

- Erstelle ein neues Paket mit einem bestimmten Betreuer\*innen-Namen:

```
choco new {{paket_name}} --maintainer {{betreuer*innen_name}}
```

- Erstelle ein neues Paket in einem bestimmten Ausgabe-Verzeichnis:

```
choco new {{paket_name}} --output-directory {{pfad/zu/verzeichnis}}
```

- Erstelle ein neues Paket mit verschiedenen URLs für die 32-Bit und 64-Bit Installationsroutinen:

```
choco new {{paket_name}} url="{{url}}" url64="{{url64}}"
```

# choco outdated

Überprüfe mit Chocolatey, ob Pakete veraltet sind.

Weitere Informationen: <https://chocolatey.org/docs/commands-outdated>.

- Zeige eine Liste von veralteten Paketen im Tabellen-Format:

```
choco outdated
```

- Ignoriere angeheftete Pakete in der Ausgabe:

```
choco outdated --ignore-pinned
```

- Gib eine eigene Quelle an, mit der die Aktualität der Pakete überprüft wird:

```
choco outdated --source {{source_url|alias}}
```

- Gib einen Benutzernamen und ein Passwort für die Authentifizierung an:

```
choco outdated --user {{benutzername}} --password  
{{password}}
```

# choco pack

Verpacke eine NuGet-Spezifikation in eine nupkg-Datei.

Weitere Informationen: <https://chocolatey.org/docs/commands-pack>.

- Verpacke eine NuGet-Spezifikation in eine nupkg-Datei:

```
choco pack {{pfad/zu/spezifikation}}
```

- Verpacke eine NuGet-Spezifikation in eine nupkg-Datei und bestimme die Version der ausgegebenen Datei:

```
choco pack {{pfad/zu/spezifikation}} --version {{version}}
```

- Verpacke eine NuGet-Spezifikation in eine nupkg-Datei, welche in einem bestimmten Verzeichnis gespeichert wird:

```
choco pack {{pfad/zu/spezifikation}} --output-directory  
{{pfad/zu/ausgabe_verzeichnis}}
```



# choco pin

Hefte ein Chocolatey-Paket bei einer bestimmten Version an.

Angeheftete Pakete werden nicht weiter aktualisiert.

Weitere Informationen: <https://chocolatey.org/docs/commands-pin>.

- Zeige eine Liste der angehefteten Pakete und ihrer Versionen an:

```
choco pin list
```

- Hefte ein Paket in der installierten Version an:

```
choco pin add --name {{paket}}
```

- Hefte ein Paket in einer bestimmten Version an:

```
choco pin add --name {{paket}} --version {{version}}
```

- Entferne die Anheftung für ein bestimmtes Paket:

```
choco pin remove --name {{paket}}
```

# choco search

Suche mit Chocolatey nach einem lokal oder im Internet verfügbaren Paket.

Weitere Informationen: <https://chocolatey.org/docs/commands-search>.

- Suche nach einem Paket:

```
choco search {{suchabfrage}}
```

- Suche nur lokal nach einem Paket:

```
choco search {{suchabfrage}} --local-only
```

- Suche nur nach genauen Übereinstimmungen:

```
choco search {{suchabfrage}} --exact
```

- Stimme allen Fragen automatisch zu:

```
choco search {{suchabfrage}} --yes
```

- Gib eine eigene Quelle an, welche nach Paketen durchsucht wird:

```
choco search {{suchabfrage}} --source {{quell_url|alias}}
```

- Gib einen Benutzernamen und ein Passwort für die Authentifizierung an:

```
choco search {{suchabfrage}} --user {{benutzername}} --  
password {{passwort}}
```

# choco source

Verwalte die Paketquellen mit Chocolatey.

Weitere Informationen: <https://chocolatey.org/docs/commands-source>.

- Gib alle momentan verfügbaren Quellen aus:

```
choco source list
```

- Füge eine neue Paketquelle hinzu:

```
choco source add --name {{name}} --source {{url}}
```

- Füge eine neue Paketquelle mit Zugangsdaten hinzu:

```
choco source add --name {{name}} --source {{url}} --user  
{{benutzername}} --password {{password}}
```

- Füge eine neue Paketquelle mit Client-Zertifikat hinzu:

```
choco source add --name {{name}} --source {{url}} --cert  
{{pfad/zu/zertifikat}}
```

- Aktiviere eine Paketquelle:

```
choco source enable --name {{name}}
```

- Deaktiviere eine Paketquelle:

```
choco source disable --name {{name}}
```

- Entferne eine Paketquelle:

```
choco source remove --name {{name}}
```

# choco uninstall

Deinstalliere mit Chocolatey ein oder mehrere Pakete.

Weitere Informationen: <https://chocolatey.org/docs/commands-uninstall>.

- Deinstalliere ein oder mehrere Pakete, deren Namen mit Leerzeichen getrennt sind:

```
choco uninstall {{paket(e)}}
```

- Deinstalliere eine bestimmte Version eines Paketes:

```
choco uninstall {{paket}} --version {{version}}
```

- Stimme allen Fragen automatisch zu:

```
choco uninstall {{paket}} --yes
```

- Deinstalliere auch alle Abhängigkeiten:

```
choco uninstall {{paket}} --remove-dependencies
```

- Deinstalliere alle Pakete:

```
choco uninstall all
```

# choco upgrade

Aktualisiere mit Chocolatey ein oder mehrere Pakete.

Weitere Informationen: <https://chocolatey.org/docs/commands-upgrade>.

- Aktualisiere ein oder mehrere Pakete, deren Namen mit Leerzeichen getrennt sind:

```
choco upgrade {{paket(e)}}
```

- Aktualisiere auf eine bestimmte Version des Pakets:

```
choco upgrade {{paket}} --version {{version}}
```

- Aktualisiere alle Pakete:

```
choco upgrade all
```

- Aktualisiere alle außer den angegebenen, durch Kommas getrennten Paketen:

```
choco upgrade all --except "{{paket(e)}}"
```

- Stimme allen Fragen automatisch zu:

```
choco upgrade {{paket}} --yes
```

- Gib eine eigene Quelle an, von der Pakete aktualisiert werden:

```
choco upgrade {{paket}} --source {{quell_url|alias}}
```

- Gib einen Benutzernamen und ein Passwort für die Authentifizierung an:

```
choco upgrade {{paket}} --user {{benutzername}} --password {{password}}
```

# choco

Ein Kommandozeilenwerkzeug für die Chocolatey Paketverwaltung.

Manche Unterbefehle wie **choco install** sind separat dokumentiert.

Weitere Informationen: <https://chocolatey.org>.

- Führe einen Chocolatey-Befehl aus:

```
choco {{befehl}}
```

- Zeige die allgemeine Hilfe an:

```
choco -?
```

- Zeige die Hilfe für einen bestimmten Befehl an:

```
choco {{befehl}} -?
```

- Überprüfe die Version von Chocolatey:

```
choco --version
```

# chrome

Dieser Befehl ist ein Alias von **chromium**.

Weitere Informationen: <https://chrome.google.com>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr chromium
```

# cinst

Dieser Befehl ist ein Alias von **choco install**.

Weitere Informationen: <https://docs.chocolatey.org/en-us/choco/commands/install>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr choco install
```



# cipher

Zeigt oder Verändert die Verschlüsselung von Verzeichnissen und Dateien auf NTFS-Laufwerken.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/cipher>.

- Informationen über die Verschlüsselung einer bestimmten Datei oder eines Verzeichnisses anzeigen lassen:

```
cipher /c:{{pfad/zu/datei_oder_verzeichnis}}
```

- Verschlüsse eine Datei oder ein Verzeichnis (nachträglich hinzugefügte Dateien werden ebenfalls verschlüsselt, da das Verzeichnis markiert ist):

```
cipher /e:{{pfad/zu/datei_oder_verzeichnis}}
```

- Entschlüsse eine Datei oder ein Verzeichnis:

```
cipher /d:{{pfad/zu/datei_oder_verzeichnis}}
```

- Entferne eine Datei oder ein Verzeichnis sicher:

```
cipher /w:{{pfad/zu/datei_oder_verzeichnis}}
```

# clip

Kopieren von Inhalten der Befehlsausgabe in die Windows Zwischenablage.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/clip>.

- Kopiere die Ausgabe eines Befehls in die Windows Zwischenablage:

```
{{dir}} | clip
```

- Kopiere den Inhalt einer Datei in die Windows Zwischenablage:

```
clip < {{pfad/zu/datei.txt}}
```

- Kopiere Text mit einem nachfolgenden Zeilenumbruch in die Windows Zwischenablage:

```
echo {{irgendein text}} | clip
```

- Kopiere Text ohne nachfolgenden Zeilenumbruch in die Windows Zwischenablage:

```
echo | set /p="irgendein text" | clip
```

# clist

Dieser Befehl ist ein Alias von **choco list**.

Weitere Informationen: <https://docs.chocolatey.org/en-us/choco/commands/list>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr choco list
```

# cmd

Auch Windows-Eingabeaufforderung genannt, der Windows-Befehlsinterpreter.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/cmd>.

- Starten einer interaktiven Shell-Sitzung:

```
cmd
```

- Ausführen eines Befehls (Command):

```
cmd /c {{echo Hallo Welt}}
```

- Ausführen eines Skripts:

```
cmd {{Pfad\zur\Datei.bat}}
```

- Ausführen eines Befehls und anschließendes Aufrufen einer interaktiven Shell:

```
cmd /k {{echo Hallo Welt}}
```

- Starten einer interaktiven Shell-Sitzung, bei der `echo` in der Befehlsausgabe deaktiviert ist:

```
cmd /q
```

- Starten einer interaktiven Shell-Sitzung mit aktivierter oder deaktivierter verzögerter Erweiterung der Umgebungsvariablen:

```
cmd /v:{{on|off}}
```

- Starten einer interaktiven Shell-Sitzung mit aktivierten oder deaktivierten Befehleerweiterungen:

```
cmd /e:{{on|off}}
```

- Starten einer interaktiven Shell-Sitzung mit Unicode-Kodierung:

```
cmd /u
```

# color

Setze die Vordergrund- und Hintergrundfarben der Kommandozeile.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/color>.

- Setze die Kommandozeilenfarben auf die Standardwerte:

```
color
```

- Liste alle verfügbaren Farben und detaillierte Informationen auf:

```
color /?
```

- Setze die Vordergrund- und Hintergrundfarbe der Kommandozeilen auf eine spezifische Farbe:

```
color {{vordergrundfarbe_code}}{{hintergrundfarbe_code}}
```

# cpush

Dieser Befehl ist ein Alias von **choco-push**.

Weitere Informationen: <https://docs.chocolatey.org/en-us/create/commands/push>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr choco-push
```

# cuninst

Dieser Befehl ist ein Alias von **choco uninstall**.

Weitere Informationen: <https://docs.chocolatey.org/en-us/choco/commands/uninstall>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr choco uninstall
```

# curl

In PowerShell kann dieser Befehl ein Alias von **Invoke-WebRequest** sein, wenn das Originalprogramm **curl** (<https://curl.se>) nicht ordnungsgemäß installiert wurde.

Weitere Informationen: <https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/invoke-webrequest>.

- Überprüfen Sie, ob `curl` ordnungsgemäß installiert ist, indem Sie sich die Versionsnummer ausgeben lassen. Wenn nachfolgender Befehl einen Fehler ausgibt, hat PowerShell diesen Befehl möglicherweise durch `Invoke-WebRequest` ersetzt:

```
curl --version
```

- Schaue dir hier die Dokumentation für den ursprünglichen `curl`-Befehl an:

```
tldr curl -p common
```

- Schaue dir hier die Dokumentation für den PowerShell-Befehl `Invoke-WebRequest` an:

```
tldr invoke-webrequest
```



# del

Lösche eine oder mehrere Dateien.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/del>.

- Lösche eine oder mehrere durch Leerzeichen getrennte Dateien oder Dateimuster:

```
del {{dateimuster}}
```

- Fordere vor dem Löschen jeder Datei zur Bestätigung auf:

```
del {{dateimuster}} /p
```

- Erzwingen das Löschen von schreibgeschützten Dateien:

```
del {{dateimuster}} /f
```

- Lösche alle Dateien, die dem Muster entsprechen, rekursiv in allen Unterordnern:

```
del {{dateimuster}} /s
```

- Zeige keine Eingabeaufforderung, wenn Dateien basierend auf einem globalen Platzhalter gelöscht werden sollen:

```
del {{dateimuster}} /q
```

- Zeige Hilfe an und liste verfügbare Attribute auf:

```
del /?
```

- Lösche Dateien mit den gegebenen Attributen:

```
del {{dateimuster}} /a {{attribut}}
```

# dir

Zeige den Inhalt von Verzeichnissen an.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/dir>.

- Zeige den Inhalt des aktuellen Verzeichnisses an:

```
dir
```

- Zeige den Inhalt des angegebenen Verzeichnisses an:

```
dir {{pfad/zu/verzeichnis}}
```

- Zeige den Inhalt, inklusive versteckter Inhalte, des aktuellen Verzeichnisses an:

```
dir /A
```

- Zeige den Inhalt, inklusive versteckte Inhalte, des angegebenen Verzeichnisses an:

```
dir {{pfad/zu/verzeichnis}} /A
```

# exit

Beendet die aktuelle CMD-Instanz oder die aktuelle Batch-Datei.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/exit>.

- Beenden der aktuellen CMD-Instanz:

```
exit
```

- Beenden des aktuellen Batch-Skripts:

```
exit /b
```

- Beenden mit einem bestimmten Exit-Code:

```
exit {{exit_code}}
```

# ftp

Interaktiver Dateitransfer zwischen einem lokalen und einem entfernten FTP-Server.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/ftp>.

- Verbinden mit einem entfernten FTP-Server:

```
ftp {{host}}
```

- Anmelden als anonymer Benutzer:

```
ftp -A {{host}}
```

- Deaktivieren der automatische Anmeldung bei der ersten Verbindung:

```
ftp -n {{host}}
```

- Ausführen einer Datei, die eine Liste von FTP-Befehlen enthält:

```
ftp -s:{{Pfad/zur/Datei}} {{host}}
```

- Herunterladen von mehreren Dateien (globaler Ausdruck):

```
mget {{*.png}}
```

- Hochladen von mehreren Dateien (globaler Ausdruck):

```
mput {{*.zip}}
```

- Löschen mehrerer Dateien auf dem entfernten Server:

```
mdelete {{*.txt}}
```

- Ausführliche Hilfe anzeigen:

```
ftp --help
```

# ipconfig

Zeige die Netzwerkkonfiguration von Windows an und verwalte diese.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/ipconfig>.

- Zeige eine Liste der Netzwerkadapter an:

```
ipconfig
```

- Zeige eine detaillierte Liste der Netzwerkadapter an:

```
ipconfig /all
```

- Erneuere die IP-Adressen für einen Netzwerkadapter:

```
ipconfig /renew {{Adapter}}
```

- Gib die IP-Adressen für einen Netzwerkadapter frei:

```
ipconfig /release {{Adapter}}
```

- Zeige den lokalen DNS-Cache an:

```
ipconfig /displaydns
```

- Entferne alle Einträge aus dem lokalen DNS-Cache:

```
ipconfig /flushdns
```

# iwr

Dieser Befehl ist ein Alias von **invoke-webrequest**.

Weitere Informationen: <https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/invoke-webrequest>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr invoke-webrequest
```

# pwsh where

Dieser Befehl ist ein Alias von **Where-Object**.

Weitere Informationen: <https://learn.microsoft.com/powershell/module/microsoft.powershell.core/where-object>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr Where-Object
```

# rd

Dieser Befehl ist ein Alias von **rmdir**.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/rd>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr rmdir
```



# robocopy

Robustes Kopieren von Dateien und Ordnern.

Standardmäßig werden Dateien nur kopiert, wenn die Quell- und Zieldatei ein unterschiedliches Änderungsdatum oder eine unterschiedliche Dateigröße haben.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/robocopy>.

- Alle .jpg und .bmp Dateien aus dem einen Verzeichnis in ein anderes Verzeichnis kopieren:

```
robocopy {{pfad/zur/quelle}} {{pfad/zum/ziel}} {{*.jpg}}  
{{*.bmp}}
```

- Alle Dateien und Unterverzeichnisse kopieren, einschließlich der leeren Verzeichnisse:

```
robocopy {{pfad/zur/quelle}} {{pfad/zum/ziel}} /E
```

- Ein Verzeichnis spiegeln/synchronisieren. Dabei wird Alles, was nicht in der Quelle vorhanden ist, gelöscht sowie alle Attribute und Berechtigungen übertragen:

```
robocopy {{pfad/zur/quelle}} {{pfad/zum/ziel}} /MIR /COPYALL
```

- Alle Dateien und Unterverzeichnisse kopieren, ausgenommen der Quelldateien, die älter sind als die vorhandenen Zieldateien:

```
robocopy {{pfad/zur/quelle}} {{pfad/zum/ziel}} /E /X0
```

- Gibt alle Dateien aus, die 50 MB und größer sind, anstatt sie zu kopieren:

```
robocopy {{pfad/zur/quelle}} {{pfad/zum/ziel}} /MIN:  
{{52428800}} /L
```

- Erlaubt das Fortsetzen des Vorgangs bei Netzwerkverlust, begrenzt die Anzahl an Versuchen auf 5 und die Wartezeit zwischen Versuchen auf 15 Sekunden:

```
robocopy {{pfad/zur/quelle}} {{pfad/zum/ziel}} /Z /R:5 /W:15
```

- Gibt detaillierte Nutzungshinweise aus:

```
robocopy /?
```

# scoop bucket

Verwalte "Eimer": Git-Repositories, welche Dateien enthalten, die beschreiben, wie Scoop Programme installiert werden.

Kennt Scoop nicht die URL eines Eimers, so muss diese angegeben werden.

Weitere Informationen: <https://github.com/lukesampson/scoop/wiki/Buckets>.

- Liste alle Eimer auf, die gerade aktiv sind:

```
scoop bucket list
```

- Liste alle bekannten Eimer auf:

```
scoop bucket known
```

- Aktiviere einen bekannten Eimer:

```
scoop bucket add {{name}}
```

- Aktiviere einen unbekannten Eimer durch die Angabe eines Namens und einer Git-Repository-URL:

```
scoop bucket add {{name}} {{https://beispiel.de/  
repository.git}}
```

- Deaktiviere einen Eimer:

```
scoop bucket rm {{name}}
```

# scoop

Ein Kommandozeilenwerkzeug, um Windows-Programme (hier bezeichnet als Pakete) zu installieren.

Weitere Informationen: <https://scoop.sh>.

- Installiere ein Paket:

```
scoop install {{paket}}
```

- Entferne ein Paket:

```
scoop uninstall {{paket}}
```

- Aktualisiere alle installierten Pakete:

```
scoop update --all
```

- Zeige alle installierten Pakete an:

```
scoop list
```

- Zeige Informationen über ein bestimmtes Paket an:

```
scoop info {{paket}}
```

- Suche nach einem Paket:

```
scoop search {{paket}}
```

- Entferne die alten Versionen aller Pakete und lösche den Download-Zwischenspeicher:

```
scoop cleanup --cache --all
```

# sls

Dieser Befehl ist ein Alias von **Select-String**.

Weitere Informationen: <https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/select-string>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr select-string
```

# wget

Dieser Befehl ist ein Alias von `wget -p common`.

Weitere Informationen: <https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/invoke-webrequest>.

- Zeige die Dokumentation für den originalen Befehl an:

```
tldr wget -p common
```

# winget

Windows Package Manager CLI.

Weitere Informationen: <https://learn.microsoft.com/windows/package-manager/winget>.

- Installiere ein Paket:

```
winget install {{paket}}
```

- Zeige Informationen über ein Paket an:

```
winget show {{paket}}
```

- Suche ein Paket:

```
winget search {{paket}}
```

# wsl

Verwalte das Windows Subsystem für Linux von der Kommandozeile.

Weitere Informationen: <https://learn.microsoft.com/windows/wsl/reference>.

- Starte eine Linux-Shell (in der Standard-Distribution):

```
wsl {{shell_befehl}}
```

- Führe einen Linux-Befehl aus, ohne eine Shell zu benutzen:

```
wsl --exec {{befehl}} {{befehl_argumente}}
```

- Gib eine bestimmte Distribution an:

```
wsl --distribution {{distribution}} {{shell_befehl}}
```

- Liste verfügbare Distributionen auf:

```
wsl --list
```

- Exportiere eine Distribution in eine .tar Datei:

```
wsl --export {{distribution}} {{pfad/zu/datei.tar}}
```

- Importiere eine Distribution von einer .tar Datei:

```
wsl --import {{distribution}} {{pfad/zu/
installations_verzeichnis}} {{pfad/zu/datei.tar}}
```

- Ändere die WSL-Version einer bestimmten Distribution:

```
wsl --set-version {{distribution}} {{version}}
```

- Fahre das Windows Subsystem für Linux herunter:

```
wsl --shutdown
```

# xcopy

Kopieren von Dateien und Verzeichnisbäumen.

Weitere Informationen: <https://learn.microsoft.com/windows-server/administration/windows-commands/xcopy>.

- Kopiere Datei(en) an den angegebenen Zielort:

```
xcopy {{pfad/zu/datei_oder_verzeichnis}} {{pfad/zu/ziel}}
```

- Liste die zu kopierenden Dateien vor dem Kopieren auf:

```
xcopy {{pfad/zu/datei_oder_verzeichnis}} {{pfad/zu/ziel}} /p
```

- Kopiere nur die Verzeichnisstruktur ohne Dateien:

```
xcopy {{pfad/zu/datei_oder_verzeichnis}} {{pfad/zu/ziel}} /t
```

- Kopiere leere Verzeichnisse mit:

```
xcopy {{pfad/zu/datei_oder_verzeichnis}} {{pfad/zu/ziel}} /e
```

- Behalte die Quell-Zugriffsrichtlinien (ACL) im Ziel Verzeichnis bei:

```
xcopy {{pfad/zu/datei_oder_verzeichnis}} {{pfad/zu/ziel}} /o
```

- Setze den Vorgang nach Unterbrechung der Netzwerkverbindung fort:

```
xcopy {{pfad/zu/datei_oder_verzeichnis}} {{pfad/zu/ziel}} /z
```

- Überschreibe bereits vorhandene Zieldateien automatisch:

```
xcopy {{pfad/zu/datei_oder_verzeichnis}} {{pfad/zu/ziel}} /y
```

- Zeige die detaillierte Hilfe an:

```
xcopy /?
```



