

# tldr pages book

Simplified and community-driven man pages

*Generated on Wed Nov 6 19:26:06 2024*

Website: <https://tldr.sh>

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

# Android

# am

Manager d'activité Android.

Plus d'informations : <https://developer.android.com/tools/adb#am>.

- Commence une activité spécifique :

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

- Commence une activité et insère de la donnée :

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

- Commence une activité qui correspond à une action et une catégorie spécifique :

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

- Convertis une intention en URI :

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

# bugreport

Affiche un rapport de bug Android.

Cette commande peut être utilisée uniquement depuis **adb shell**.

Plus d'informations : <https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreport>.

- Affiche un rapport de bug d'un appareil Android :

```
bugreport
```

# bugreportz

Génère une archive d'un rapport de bug Android.

Cette commande peut être utilisée uniquement depuis **adb shell**.

Plus d'informations : <https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreportz>.

- Génère une archive à partir d'un rapport de bug d'un appareil Android :

```
bugreportz
```

- Affiche la progression courante d'une opération bugreportz :

```
bugreportz -p
```

- Affiche l'aide :

```
bugreportz -h
```

- Affiche la version de bugreportz :

```
bugreportz -v
```

# cmd

Manager de service Android.

Plus d'informations : <https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/cmd/>.

- Liste tous les services en cours d'exécution :

```
cmd -l
```

- Appelle un service spécifique :

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

- Appelle un service avec des arguments :

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

# dalvikvm

Machine virtuelle Java pour Android.

Plus d'informations : <https://source.android.com/docs/core/runtime>.

- Démarre un programme Java :

```
dalvikvm -classpath {{chemin/vers/fichier.jar}}  
{{nom_de_la_classe}}
```

# dumpsys

Fourni des informations sur les services du système Android.

Cette commande peut être utilisée uniquement depuis **adb shell**.

Plus d'informations : <https://developer.android.com/tools/dumpsys>.

- Récupère un diagnostic pour chaque service système :

```
dumpsys
```

- Récupère un diagnostic pour un service système spécifique :

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

- Liste tous les services dont **dumpsys** peut donner les informations :

```
dumpsys -l
```

- Liste les arguments spécifiques d'un service :

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

- Exclut un service spécifique d'un diagnostic :

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

- Spécifie un temps limite en secondes (10s par défaut) :

```
dumpsys -t {{seconds}}
```



# getprop

Affiche les informations des propriétés système Android.

Plus d'informations : <https://manned.org/getprop>.

- Affiche les informations des propriétés système Android :

```
getprop
```

- Affiche les informations d'une propriété spécifique :

```
getprop {{prop}}
```

- Affiche le niveau API du SDK :

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

- Affiche la version d'Android :

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

- Affiche le modèle de l'appareil Android :

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

- Affiche le status du déblocage OEM :

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

- Affiche l'adresse MAC de la carte Wi-Fi Android :

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

# input

Envoie à un appareil Android des codes événements ou des gestes d'écran tactile.

Cette commande peut être utilisée uniquement depuis **adb shell**.

Plus d'informations : <https://developer.android.com/reference/android/view/KeyEvent.html#constants> 1.

- Envoie un code événement (un seul caractère) à un appareil Android :

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

- Envoie du texte à un appareil Android (%s représentant les espaces) :

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

- Envoie un tapotement (tap) à un appareil Android :

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

- Envoie un mouvement de swipe à un appareil Android :

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

- Envoie un appui long à un appareil Android en utilisant un mouvement de swipe :

```
input swipe {{x_pos}} {{y_pos}} {{x_pos}} {{y_pos}}  
{{duration_in_ms}}
```

# logcat

Exporte une log depuis les messages système.

Plus d'informations : <https://developer.android.com/tools/logcat>.

- Affiche la journalisation système :

```
logcat
```

- Écris la journalisation système dans un fichier :

```
logcat -f {{chemin/vers/fichier}}
```

- Affiche les lignes qui correspondent à une expression régulière :

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

# pkg

Utilitaire de gestion de paquets pour Termux.

Plus d'informations : [https://wiki.termux.com/wiki/Package\\_Management](https://wiki.termux.com/wiki/Package_Management).

- Mets à jour tous les paquets déjà installés :

```
pkg upgrade
```

- Installe un paquet :

```
pkg install {{paquet}}
```

- Désinstalle un paquet :

```
pkg uninstall {{paquet}}
```

- Réinstalle un paquet :

```
pkg reinstall {{paquet}}
```

- Cherche un paquet :

```
pkg search {{paquet}}
```

# pm

Afficher des informations sur les applications d'un appareil Android.

Plus d'informations : <https://developer.android.com/tools/adb#pm>.

- Affiche la liste des applications installées :

```
pm list packages
```

- Affiche une liste de toutes les applications système installées :

```
pm list packages -s
```

- Affiche une liste de toutes les applications tierces :

```
pm list packages -3
```

- Affiche une liste des applications qui correspondent à des mots clés :

```
pm list packages {{mots_clés}}
```

- Affiche le chemin vers l'APK d'une application spécifique :

```
pm path {{application}}
```

# screencap

Prenez une capture d'écran d'un écran mobile.

Cette commande ne peut être utilisée que via **adb shell**.

Plus d'informations : <https://developer.android.com/tools/adb#screencap>.

- Prendre une capture d'écran :

```
screencap {{chemin/vers/fichier}}
```

# settings

Récupère les informations du système d'exploitation Android.

Plus d'informations : <https://adbinstaller.com/commands/adb-shell-settings-5b670d5ee7958178a2955536>.

- Affiche une liste des paramètres de l'espace de nom `global` :

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

- Récupère la valeur d'un paramètre :

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

- Assigne une valeur à un paramètre :

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

- Supprime un paramètre :

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

# wm

Affiche les informations de l'écran d'un appareil Android.

Cette commande peut être utilisée uniquement depuis **adb shell**.

Plus d'informations : <https://adbinstaller.com/commands/adb-shell-wm-5b672b17e7958178a2955538>.

- Affiche la taille physique de l'écran d'un appareil Android :

```
wm size
```

- Affiche la densité physique de l'écran d'un appareil Android :

```
wm density
```



Common

# 2to3

Conversion automatisé de code Python 2 vers 3.

Plus d'informations : <https://manned.org/2to3>.

- Afficher les changements qui seront effectués sans les effectuer (coup d'essai) :

```
2to3 {{chemin/vers/fichier.py}}
```

- Convertir un fichier Python 2 en Python 3 :

```
2to3 --write {{chemin/vers/fichier.py}}
```

- Convertir des fonctionnalités spécifiques de Python 2 vers Python 3 :

```
2to3 --write {{chemin/vers/fichier.py}} --fix {{raw_input}}  
--fix {{print}}
```

- Convertir toutes les fonctionnalités de Python 2 vers Python 3 sauf exceptions spécifiques :

```
2to3 --write {{chemin/vers/fichier.py}} --nofix {{has_key}}  
--nofix {{isinstance}}
```

- Afficher une liste de toutes les fonctionnalités de langage qui peuvent être converties de Python 2 vers Python 3 :

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

- Convertir tous les fichier Python2 vers Python 3 dans un dossier :

```
2to3 --output-dir {{chemin/vers/dossier_python3}} --write-  
unchanged-files --nobackups {{chemin/vers/dossier_python2}}
```

- Executer 2to3 avec plusieurs fil d'exécution :

```
2to3 --processes {{4}} --output-dir {{chemin/vers/  
dossier_python3}} --write --nobackups --no-diff {{chemin/  
vers/dossier_python2}}
```

# 7z

Un archiveur de fichiers avec un haut taux de compression.

Plus d'informations : <https://manned.org/7z>.

- Compresser un fichier ou un dossier :

```
7z a {{archive.7z}} {{chemin/vers/fichier_ou_dossier}}
```

- Chiffrer une archive existante (en incluant les en-têtes) :

```
7z a {{archive_chiffree.7z}} -p{{motdepasse}} -mhe=on  
{{archive.7z}}
```

- Extraire un fichier 7z existant en conservant l'arborescence des fichiers :

```
7z x {{archive.7z}}
```

- Extraire une archive vers la destination donnée :

```
7z x {{archive.7z}} -o{{chemin/vers/destination}}
```

- Extraire une archive vers la sortie standard :

```
7z x {{archive.7z}} -so
```

- Archiver en utilisant un algorithme de compression particulier :

```
7z a -t{{7z|bzip2|gzip|lzip|tar|zip}} {{archive}} {{chemin/  
vers/fichier_ou_dossier}}
```

- Lister le contenu d'une archive :

```
7z l {{archive.7z}}
```

# 7za

Un archiveur de fichiers avec un haut taux de compression.

Similaire à **7z** sauf qu'il supporte moins de type de fichiers mais il est multi-plateforme.

Plus d'informations : <https://manned.org/7za>.

- Comprime un fichier ou un dossier :

```
7za a {{chemin/vers/archive.7z}} {{chemin/vers/
file_or_directory}}
```

- Chiffre une archive existante (en incluant les en-têtes) :

```
7za a {{chemin/vers/archive_chiffree.7z}} -p{{motdepasse}} -
mhe=on {{chemin/vers/archive.7z}}
```

- Extrait une archive en conservant l'arborescence des fichiers :

```
7za x {{chemin/vers/archive.7z}}
```

- Extrait une archive vers un dossier spécifique :

```
7za x {{chemin/vers/archive.7z}} -o{{chemin/vers/la/sortie}}
```

- Extrait une archive vers sortie standard :

```
7za x {{chemin/vers/archive.7z}} -so
```

- Comprime en utilisant une compression spécifique :

```
7za a -t{{7z|bzip2|gzip|lzip|tar|zip}} {{chemin/vers/archive.
7z}} {{chemin/vers/le/fichier_ou_dossier}}
```

- Liste le contenu d'une archive :

```
7za l {{chemin/vers/archive.7z}}
```

# 7zr

Un archiveur de fichiers avec un haut taux de compression.

Similaire à **7z** sauf qu'il supporte que les fichiers 7z.

Plus d'informations : <https://manned.org/7zr>.

- Comprime un fichier ou un dossier :

```
7zr a {{chemin/vers/archive.7z}} {{chemin/vers/le/  
fichier_ou_dossier}}
```

- Chiffre une archive existante (en incluant les en-têtes) :

```
7zr a {{chemin/vers/archive_chiffree.7z}} -p{{password}} -  
mhe={{on}} {{chemin/vers/archive.7z}}
```

- Extrait une archive en conservant l'arborescence des fichiers :

```
7zr x {{chemin/vers/archive.7z}}
```

- Extrait une archive vers un dossier spécifique :

```
7zr x {{chemin/vers/archive.7z}} -o{{chemin/vers/la/sortie}}
```

- Extrait une archive vers sortie standard :

```
7zr x {{chemin/vers/archive.7z}} -so
```

- Liste le contenu d'une archive :

```
7zr l {{chemin/vers/archive.7z}}
```

# aapt

Android Asset Packaging Tool.

Compile et empaquette les ressources d'une application Android.

Plus d'informations : <https://manned.org/aapt>.

- Liste les fichiers contenus une archive APK :

```
aapt list {{chemin/vers/app.apk}}
```

- Affiche les metadatas d'une application (version, autorisations, etc.) :

```
aapt dump badging {{chemin/vers/app.apk}}
```

- Créé une nouvelle archive APK avec les fichiers venant d'un dossier spécifique :

```
aapt package -F {{chemin/vers/app.apk}} {{chemin/vers/le/  
dossier}}
```

# ab

Outil d'analyse pour serveur Apache HTTP.

Plus d'informations : <https://httpd.apache.org/docs/current/programs/ab.html>.

- Exécute 100 requêtes HTTP GET sur une URL donnée :

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

- Exécute 100 requêtes HTTP GET en parallèle par groupe de 10, sur une URL :

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

- Exécute 100 requêtes HTTP POST sur une URL, en utilisant un contenu JSON depuis un fichier :

```
ab -n 100 -T {{application/json}} -p {{chemin/vers/le/
fichier.json}} {{url}}
```

- Utilise la fonctionnalité HTTP Keep Alive pour exécuter plusieurs requêtes dans la même session HTTP :

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

- Fixe le nombre maximum de secondes d'exécution pour l'analyse :

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

# abduco

Manageur de session dans un terminal.

Plus d'informations : <https://www.brain-dump.org/projects/abduco/>.

- Affiche les sessions :

```
abduco
```

- Rejoint une session, la crée si elle n'existe pas :

```
abduco -A {{nom}} {{bash}}
```

- Rejoint une session avec `dvtm`, la crée si elle n'existe pas :

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

- Quitte la session courante :

```
<Ctrl> + \
```

- Rejoint une session en mode lecture seule :

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



# ac

Imprime les statistiques sur combien de temps les utilisateurs ont été connectés.

Plus d'informations : <https://man.openbsd.org/ac>.

- Imprime combien de temps l'utilisateur actuel a été connecté en heures :

```
ac
```

- Imprime combien de temps les utilisateurs ont été connecté :

```
ac -p
```

- Imprime combien de temps un utilisateur en particulier a été connecté en heures :

```
ac -p {{nom_d_utilisateur}}
```

- Imprime combien de temps un utilisateur particulier a été connecté en heure par jour (avec le total) :

```
ac -dp {{nom_d_utilisateur}}
```

# ack

Un outil de recherche comme grep, optimisé pour les développeurs.

Regardez aussi : **rg**, qui est beaucoup plus rapide.

Plus d'informations : <https://beyondgrep.com/documentation>.

- Recherche des fichiers contenant une chaîne de caractère ou une expression régulière dans le répertoire courant récursivement :

```
ack "{{motif_de_recherche}}"
```

- Recherche pour un motif insensible à la casse :

```
ack --ignore-case "{{motif_de_recherche}}"
```

- Recherche les lignes qui correspondent à un motif, affiche uniquement les textes correspondants et pas le reste de la ligne :

```
ack -o "{{motif_de_recherche}}"
```

- Limite la recherche aux fichiers d'un certain type :

```
ack --type {{ruby}} "{{motif_de_recherche}}"
```

- Exclut un certain type de fichier de la recherche :

```
ack --type no{{ruby}} "{{motif_de_recherche}}"
```

- Compte le nombre total de correspondances :

```
ack --count --no-filename "{{motif_de_recherche}}"
```

- Affiche les noms et le nombre total de correspondances pour chaque fichier :

```
ack --count --files-with-matches "{{motif_de_recherche}}"
```

- Affiche toutes les valeurs qui peuvent être utilisées avec --type :

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

# acme.sh

Script shell implémentant le protocole client ACME, une alternative à **certbot**.

Voir aussi **acme.sh dns**.

Plus d'informations : <https://github.com/acmesh-official/acme.sh>.

- Publie un certificat en utilisant le mode webroot :

```
acme.sh --issue --domain {{exemple.com}} --webroot {{/chemin/vers/webroot}}
```

- Publie un certificat pour plusieurs domaines en utilisant le mode autonome avec le port 80 :

```
acme.sh --issue --standalone --domain {{exemple.com}} --domain {{www.exemple.com}}
```

- Publie un certificat en utilisant le mode autonome TLS avec le port 443 :

```
acme.sh --issue --alpn --domain {{exemple.com}}
```

- Publie un certificat en utilisant une configuration Nginx :

```
acme.sh --issue --nginx --domain {{exemple.com}}
```

- Publie un certificat en utilisant une configuration Apache :

```
acme.sh --issue --apache --domain {{exemple.com}}
```

- Publie un certificat wildcard (\*) en utilisant le mode automatique DNS API :

```
acme.sh --issue --dns {{dns_cf}} --domain {{*.exemple.com}}
```

- Installe les fichiers de certificat dans un dossier spécifique (Utile pour les renouvellements automatiques de certificat) :

```
acme.sh --install-cert -d {{exemple.com}} --key-file {{/chemin/vers/exemple.com.key}} --fullchain-file {{/chemin/vers/exemple.com.cer}} --reloadcmd {{"systemctl force-reload nginx"}}
```

# act

Execute des GitHub Actions en local avec Docker.

Plus d'informations : <https://github.com/nektos/act>.

- [l]iste les jobs disponibles :

```
act -l
```

- Execute l'événement par défaut :

```
act
```

- Execute un événement spécifique :

```
act {{type_d_événement}}
```

- Execute un [j]ob spécifique :

```
act -j {{id_job}}
```

- Ne pas lancer les actions maintenant (e.g un essai) :

```
act -n
```

- Affiche le journal en mode verbeux :

```
act -v
```

- Execute un [W]orkflow en particulier, avec l'événement push :

```
act push -W {{path/to/workflow}}
```

# acyclic

Construit un graphe orienté acyclique en inversant quelques sommets.

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

Plus d'informations : <https://graphviz.org/pdf/acyclic.1.pdf>.

- Construit un graphe orienté acyclique en inversant quelques sommets :

```
acyclic {{chemin/vers/entrée.gv}} > {{chemin/vers/sortie.gv}}
```

- Affiche si un graphe est acyclique, possède une boucle ou est non-dirigé, ne produit pas de graphe en sortie :

```
acyclic -v -n {{chemin/vers/entrée.gv}}
```

- Affiche l'aide d' `acyclic` :

```
acyclic -?
```

# adb install

Android Debug Bridge Install: Pousse des paquets vers une instance d'émulateur Android ou un appareil Android.

Plus d'informations : <https://developer.android.com/tools/adb>.

- Pousse une application Android vers l'émulateur/l'appareil :

```
adb install {{chemin/vers/le/fichier.apk}}
```

- Pousse une application Android vers l'émulateur/l'appareil spécifique via son numéro de série (écrase la variable \$ANDROID\_SERIAL):

```
adb -s {{numero_de_serie}} install {{chemin/vers/le/fichier.apk}}
```

- Réinstalle une application existante, tout en gardant ses données :

```
adb install -r {{chemin/vers/le/fichier.apk}}
```

- Pousse une application Android en autorisant la rétrogradation de version (uniquement pour les paquets debuggable) :

```
adb install -d {{chemin/vers/le/fichier.apk}}
```

- Accorde toutes les permissions listées dans le manifeste de l'application :

```
adb install -g {{chemin/vers/le/fichier.apk}}
```

- Mets à jour rapidement un paquet en mettant à jour uniquement les parties de l'APK qui ont changé :

```
adb install --fastdeploy {{chemin/vers/le/fichier.apk}}
```

# adb logcat

Jeter une log des messages systèmes.

Plus d'informations : <https://developer.android.com/tools/logcat>.

- Affiche les logs systèmes :

```
adb logcat
```

- Affiche les logs qui correspond à une expression régulière :

```
adb logcat -e {{expression_régulière}}
```

- Affiche les logs pour un tag donné, dans un mode spécifique ([V]erbose, [D]ebug, [I]nfo, [W]arning, [E]rror, [F]atal, [S]ilent) :

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

- Affiche les logs pour des applications React Native en mode [V]erbose et mes sous [S]ilence les autres tags :

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

- Affiche les logs avec un niveau de priorité supérieur à [W]arning :

```
adb logcat *:W
```

- Colorie les logs (souvent utilisé avec des filtres) :

```
adb logcat -v color
```

# adb reverse

Android Debug Bridge Reverse: Transfère des connections réseaux depuis une instance d'émulateur Android ou un appareil Android.

Plus d'informations : <https://developer.android.com/tools/adb>.

- Affiche la liste de toutes les connections réseaux qui sont transféré depuis les émulateurs ou les appareils :

```
adb reverse --list
```

- Transfère un port TCP depuis un émulateur ou un appareil vers localhost :

```
adb reverse tcp:{{port_distant}} tcp:{{port_local}}
```

- Supprime une connection socket en cours depuis un émulateur ou un appareil :

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

- Supprime toutes les connections socket depuis tous les émulateurs et appareils :

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



# adb shell

Android Debug Bridge Shell: Exécute une commande shell sur une instance d'émulateur Android ou un appareil Android.

Plus d'informations : <https://developer.android.com/tools/adb>.

- Démarre une session shell interactive sur l'émulateur/l'appareil :

```
adb shell
```

- Récupère toutes les propriétés depuis un émulateur ou un appareil :

```
adb shell getprop
```

- Remet toutes les permissions courante à leurs valeurs par défaut :

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

- Révoque une permission dangereuse pour une application :

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

- Déclenche un événement clé :

```
adb shell input keyevent {{code}}
```

- Nettoie les données d'une application sur un émulateur ou un appareil :

```
adb shell pm clear {{paquet}}
```

- Démarre une activité sur un émulateur ou un appareil :

```
adb shell am start -n {{paquet}}/{{activité}}
```

- Démarre une activité maison depuis un émulateur ou un appareil :

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

# adb

Android Debug Bridge: Communiquer avec une instance d'émulateur Android ou un appareil Android.

Certaines commandes comme **adb shell** ont leur propre documentation.

Plus d'informations : <https://developer.android.com/tools/adb>.

- Vérifie si le processus du serveur adb est en fonctionnement et le démarre :

```
adb start-server
```

- Arrête le processus du serveur adb :

```
adb kill-server
```

- Démarre un shell distant sur l'émulateur/l'appareil ciblé :

```
adb shell
```

- Pousse une application Android vers l'émulateur/l'appareil :

```
adb install -r {{chemin/vers/le/fichier.apk}}
```

- Copie un fichier/dossier depuis l'appareil ciblé :

```
adb pull {{chemin/vers/le/fichier_ou_dossier_de_l'appareil}}  
{{chemin/vers/le/dossier_de_destination_local}}
```

- Copie un fichier/dossier vers l'appareil ciblé :

```
adb push {{chemin/vers/le/fichier_ou_dossier_local}}  
{{chemin/vers/le/dossier_de_destination_de_l'appareil}}
```

- Récupère une liste des appareils connectés :

```
adb devices
```

# AdGuardHome

Un logiciel réseau pour bloquer les pubs et les traqueurs.

Plus d'informations : <https://github.com/AdguardTeam/AdGuardHome>.

- Lance AdGuard Home :

```
AdGuardHome
```

- Lance AdGuard Home avec une configuration spécifique :

```
AdGuardHome --config {{chemin/vers/AdGuardHome.yaml}}
```

- Configure le répertoire de travail où les données seront stockées :

```
AdGuardHome --work-dir {{chemin/vers/répertoire}}
```

- Installe ou désinstalle AdGuard Home comme un service :

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

- Démarre le service AdGuard Home :

```
AdGuardHome --service start
```

- Recharge la configuration pour le service AdGuard Home :

```
AdGuardHome --service reload
```

- Éteint ou redémarre le service AdGuard Home :

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

# adscript

Compilateur pour fichiers Adscript.

Plus d'informations : <https://github.com/Amplus2/Adscript>.

- Compile un fichier vers un fichier objet :

```
adscript --output {{chemin/vers/fichier.o}} {{chemin/vers/
fichier_source.adscript}}
```

- Compile et lie un fichier vers un exécutable autonome :

```
adscript --executable --output {{chemin/vers/fichier}}
{{chemin/vers/fichier_source.adscript}}
```

- Compile un fichier vers LLVM IR à la place du code machine natif :

```
adscript --llvm-ir --output {{chemin/vers/fichier.ll}}
{{chemin/vers/fichier_source.adscript}}
```

- Fait une compilation croisée d'un fichier vers un fichier objet pour une architecture CPU ou un système d'exploitation distant :

```
adscript --target-triple {{i386-linux-elf}} --output
{{chemin/vers/fichier.o}} {{chemin/vers/
fichier_source.adscript}}
```

# afconvert

Convertir entre les formats de fichiers AFF et brut.

Plus d'informations : <https://manned.org/afconvert.1>.

- Utiliser une extension spécifique (par défaut: aff) :

```
afconvert -a {{extension}} {{chemin/vers/fichier_source}}  
{{chemin/vers/fichier_de_sortie1 chemin/vers/  
fichier_de_sortie2 ...}}
```

- Utiliser un niveau de compression spécifique (par défaut: 7) :

```
afconvert -X{{0..7}} {{chemin/vers/fichier_source}} {{chemin/  
vers/fichier_de_sortie1 chemin/vers/fichier_de_sortie2 ...}}
```

# ag

The Silver Searcher. Comme **ack**, mais inspire à être plus rapide.

Plus d'informations : [https://github.com/ggreer/the\\_silver\\_searcher](https://github.com/ggreer/the_silver_searcher).

- Trouve les fichiers qui contiennent "foo", et affiche les lignes correspondantes dans le contexte courant :

```
ag {{foo}}
```

- Trouve les fichiers qui contiennent "foo" dans un dossier spécifique :

```
ag {{foo}} {{chelin/vers/dossier}}
```

- Trouve les fichiers qui contiennent "foo", mais affiche les nom de fichier uniquement :

```
ag -l {{foo}}
```

- Trouve les fichiers qui contiennent "FOO" en étant insensible à la casse et affiche que le premier résultat plutôt que la ligne entière :

```
ag -i -o {{F00}}
```

- Trouve "foo" dans les fichiers avec un nom contenant "bar" :

```
ag {{foo}} -G {{bar}}
```

- Trouve des fichiers dont le contenu correspond à une expression régulière :

```
ag '{{^ba(r|z)$}}'
```

- Trouve les fichiers avec un nom contenant "foo" :

```
ag -g {{foo}}
```

# agate

Un serveur simple pour le protocole réseau Gemini.

Plus d'informations : <https://github.com/mbrubeck/agate>.

- Exécute et génère une clé privée et un certificat :

```
agate --content {{chemin/vers/contenu/}} --addr {[::]:1965}  
--addr {{0.0.0.0:1965}} --hostname {{exemple.com}} --lang  
{{fr-FR}}
```

- Lance le serveur :

```
agate {{chemin/vers/fichier}}
```

- Affiche l'aide :

```
agate -h
```

# age

Un outil de cryptage de fichiers simple, moderne et sécurisé.

Plus d'informations : <https://github.com/FiloSottile/age>.

- Générez un fichier crypté qui peut être décrypté avec une mot de passe :

```
age --passphrase --output {{chemin/vers/fichier_crypté}}  
{{chemin/vers/fichier_non_crypté}}
```

- Générer une paire de clés, en enregistrant la clé privée dans un fichier non crypté et en imprimant la clé publique sur stdout:

```
age-keygen --output {{chemin/vers/fichier}}
```

- Cryptage d'un fichier avec une ou plusieurs clés publiques qui sont entrées comme des littéraux :

```
age --recipient {{clé_publicue_1}} --recipient  
{{clé_publicue_2}} {{chemin/vers/fichier_non_crypté}} --  
output {{chemin/vers/fichier_crypté}}
```

- Cryptez un fichier avec une ou plusieurs clés publiques spécifiées dans un fichier de destinataires :

```
age --recipients-file {{chemin/vers/fichier_destinataire}}  
{{chemin/vers/fichier_non_crypté}} --output {{chemin/vers/  
fichier_crypté}}
```

- Déchiffrer un fichier avec un mot de passe :

```
age --decrypt --output {{chemin/vers/fichier_décrypté}}  
{{chemin/vers/fichier_crypté}}
```

- Decrypt a file with a private key file :

```
age --decrypt --identity {{chemin/vers/fichier_clé_privée}}  
--output {{chemin/vers/fichier_décrypté}} {{chemin/vers/  
fichier_crypté}}
```



# airmon-ng

Active le mode surveillance sur les appareils sans fils.

Plus d'informations : <https://www.aircrack-ng.org/doku.php?id=airmon-ng>.

- Liste les appareils sans fils et leurs statuts :

```
sudo airmon-ng
```

- Allume le mode surveillance sur un appareil spécifique :

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

- Tue les processus nuisibles qui utilisent les appareils sans fils :

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

- Éteint le mode surveillance pour une interface réseau spécifique :

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

# airpaste

Partage des messages et des fichiers sur le même réseau en utilisant mDNS.

Plus d'informations : <https://github.com/mafintosh/airpaste>.

- Attend un message et l'affiche une fois reçu :

```
airpaste
```

- Envoie un message :

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

- Envoie un fichier :

```
airpaste < {{chemin/vers/fichier}}
```

- Recevoir un fichier :

```
airpaste > {{chemin/vers/fichier}}
```

- Crée ou rejoins un canal :

```
airpaste {{nom_du_canal}}
```

# airshare

Transférer des données entre deux machines dans un réseau local.

Plus d'informations : <https://airshare.rtf.d.io/en/latest/cli.html>.

- Partager des fichiers ou des dossiers :

```
airshare {{code}} {{chemin/vers/fichier_ou_dossier1 chemin/vers/fichier_ou_dossier2 ...}}
```

- Recevoir un fichier :

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

- Héberger un serveur de réception (utilisez ceci pour pouvoir télécharger des fichiers via l'interface web) :

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

- Envoyer des fichiers ou des dossiers a un serveur de reception :

```
airshare --upload {{code}} {{chemin/vers/fichier_ou_dossier1 chemin/vers/fichier_ou_dossier2 ...}}
```

- Envoyer les fichiers dont les chemins ont été copiés dans le presse-papiers :

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

- Recevoir un fichier et le copier dans le presse-papier :

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

# ajson

Exécute un JSONPath sur un objet JSON.

Plus d'informations : <https://github.com/spyzhov/ajson>.

- Lis un JSON depuis un fichier et exécute une expression JSONPath spécifique :

```
ajson '{{$.json[?(@.path)]}}' {{chemin/vers/fichier.json}}
```

- Lis un JSON depuis l'entrée standard et exécute une expression JSONPath spécifique :

```
cat {{chemin/vers/fichier.json}} | ajson '{{$.json[?(@.path)]}}'
```

- Lis un JSON depuis une URL et évalue une expression JSONPath spécifique :

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

- Lis un JSON simple et calcule une valeur :

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

# alacritty

Emulateur de terminal propulsé par GPU, Multi-plateforme.

Plus d'informations : <https://github.com/alacritty/alacritty>.

- Ouvre une nouvelle fenêtre Alacritty :

```
alacritty
```

- Lance dans un dossier spécifique :

```
alacritty --working-directory {{chemin/vers/dossier}}
```

- Lance une commande dans une nouvelle fenêtre Alacritty :

```
alacritty -e {{commande}}
```

- Utilise un autre fichier de configuration (le fichier par défaut étant `$XDG_CONFIG_HOME/alacritty/alacritty.toml`):

```
alacritty --config-file {{chemin/vers/config.toml}}
```

- Lance avec la mise à jour en live dès que la configuration est modifiée ( peu également être activé par défaut dans `alacritty.toml`):

```
alacritty --live-config-reload --config-file {{chemin/vers/config.toml}}
```

# alex

Un outil qui corrige les phrases insensibles et inconsidérées (en Anglais uniquement).

Il vous aide à trouver un genre, une polarité, une ethnie, un blasphème, ou autre inégalité en lisant un texte en anglais.

Plus d'informations : <https://github.com/get-alex/alex>.

- Analyser un texte depuis l'entrée standard :

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

- Analyser tous les fichiers dans le dossier courant :

```
alex
```

- Analyser un fichier spécifique :

```
alex {{fichiertexte.md}}
```

- Analyser tous les fichiers Markdown sauf `exemple.md` :

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

# alias

Crée des alias -- mots qui sont remplacés par une commande textuelle.

Les alias expirent avec la session courante shell, sauf s'il a été défini dans le fichier de configuration shell, par exemple `~/.bashrc`.

Plus d'informations : <https://tldp.org/LDP/abs/html/aliases.html>.

- Liste tous les alias :

```
alias
```

- Crée un alias générique :

```
alias {{mot}}="{{commande}}"
```

- Affiche la commande associée à un alias donné :

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

- Enlève un alias :

```
unalias {{mot}}
```

- Transforme `rm` en une commande interactive :

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

- Crée `la` comme un raccourci de `ls -a` :

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

# amass enum

Trouve les sous-domaines d'un domaine.

Plus d'informations : [https://github.com/owasp-amass/amass/blob/master/doc/user\\_guide.md#the-enum-subcommand](https://github.com/owasp-amass/amass/blob/master/doc/user_guide.md#the-enum-subcommand).

- Trouve les sous-domaines d'un domaine passivement :

```
amass enum -passive -d {{nom_de_domaine}}
```

- Trouve les sous-domaines d'un domaine et les verifie activement en essayant de résoudre les sous-domaines trouvés :

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

- Fait recherche par force brute pour les sous-domaines :

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

- Sauvegarde les résultats vers un fichier texte :

```
amass enum -o {{fichier_de_sortie}} -d {{nom_de_domaine}}
```

- Sauvegarde les résultats dans une base de données :

```
amass enum -o {{fichier_de_sortie}} -dir {{chemin/vers/la_base_de_données}}
```



# amass intel

Collecte des renseignements open source sur une organisation comme les noms de domaines racine et les ASNs.

Plus d'informations : [https://github.com/owasp-amass/amass/blob/master/doc/user\\_guide.md#the-intel-subcommand](https://github.com/owasp-amass/amass/blob/master/doc/user_guide.md#the-intel-subcommand).

- Recherche les domaines racines inclus dans une plage d'adresse IP :

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

- Utilise les méthodes de reconnaissance active :

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

- Recherche les noms de domaines racines reliés à un domaine :

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

- Recherche les ASNs qui correspondent à une organisation :

```
amass intel -org {{nom_de_l_organisation}}
```

- Recherche les domaines racines qui correspondent à un ASN :

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

- Sauvegarde les résultats dans un fichier texte :

```
amass intel -o {{fichier_de_sortie}} -whois -d {{nom_de_domaine}}
```

# amass

Outil de Cartographie des Attaques de Surface et découverte de ressource.

Certaines commandes comme **amass intel** ont leur propre documentation.

Plus d'informations : <https://github.com/owasp-amass/amass>.

- Exécute une sous-commande Amass :

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

- Affiche l'aide général :

```
amass -help
```

- Affiche l'aide sur une sous-commande de Amass :

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

- Affiche la version :

```
amass -version
```

# androguard

Outil d'ingénierie inverse pour les applications Android. Écrit en Python.

Plus d'informations : <https://github.com/androguard/androguard>.

- Affiche le manifest d'application Android :

```
androguard axml {{chemin/vers/app.apk}}
```

- Affiche les métadonnées de l'application (version et ID d'application) :

```
androguard apkid {{chemin/vers/app.apk}}
```

- Décompile le code Java de l'application :

```
androguard decompile {{chemin/vers/app.apk}} --output  
{{chemin/vers/dossier}}
```

# ani-cli

Une CLI pour chercher et regarder des animés.

Plus d'informations : <https://github.com/pystardust/ani-cli>.

- Cherche un anime par nom :

```
ani-cli "{{nom_de_l_animé}}"
```

- Télécharge l'épisode :

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

- Utilise VLC comme lecteur de video :

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

- Spécifie un épisode à regarder :

```
ani-cli -e {{numéro_de_l_épisode}} "{{nom_de_l_animé}}"
```

- Continue de regarder l'animé depuis l'historique :

```
ani-cli -c
```

- Met à jour ani-cli :

```
ani-cli -U
```

# anki

Programme de cartes mémo-techniques, puissant et intelligent.

Plus d'informations : <https://docs.ankiweb.net>.

- Lancer `anki`:

```
anki
```

- Lancer `anki` avec un profil spécifique :

```
anki -p {{nom_de_profile}}
```

- Lancer `anki` dans une langue spécifique :

```
anki -l {{langue}}
```

- Lancer `anki` depuis un dossier spécifique au lieu de celui par défaut (`~/Anki`):

```
anki -b {{chemin/vers/dossier}}
```

# ansible-doc

Affiche les informations des modules installés dans les librairies Ansible.

Affiche une liste concise des plugins et leurs description courte.

Plus d'informations : <https://docs.ansible.com/ansible/latest/cli/ansible-doc.html>.

- Liste les plugins action (modules) disponibles :

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

- Liste les plugins disponible d'un certain type :

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

- Affiche les informations sur un plugin action (module) spécifique :

```
ansible-doc {{nom_du_plugin}}
```

- Affiche les informations sur un plugin avec un certain type :

```
ansible-doc --type {{type_de_plugin}} {{nom_du_plugin}}
```

- Affiche le raccourci playbook d'un plugin action (module) :

```
ansible-doc --snippet {{nom_du_plugin}}
```

- Affiche les informations sur un plugin action (module) en JSON :

```
ansible-doc --json {{nom_du_plugin}}
```

# ansible-galaxy

Crée et gère les rôles Ansible.

Plus d'informations : <https://docs.ansible.com/ansible/latest/cli/ansible-galaxy.html>.

- Installe un rôle :

```
ansible-galaxy install {{nom_d_utilisateur}}.{{nom_du_rôle}}
```

- Enlève un rôle :

```
ansible-galaxy remove {{nom_d_utilisateur}}.{{nom_du_rôle}}
```

- Liste les rôles installés :

```
ansible-galaxy list
```

- Recherche pour un rôle donné :

```
ansible-galaxy search {{nom_du_rôle}}
```

- Crée un nouveau rôle :

```
ansible-galaxy init {{nom_du_rôle}}
```

- Récupère les informations sur le rôle d'un utilisateur :

```
ansible-galaxy role info {{nom_d_utilisateur}}.  
{{nom_du_rôle}}
```

- Récupère les informations d'une collection :

```
ansible-galaxy collection info {{nom_d_utilisateur}}.  
{{nom_de_la_collection}}
```

# ansible-inventory

Display or dump an Ansible inventory.

Voir aussi : **ansible**.

Plus d'informations : <https://docs.ansible.com/ansible/latest/cli/ansible-inventory.html>.

- Affiche l'inventaire par défaut :

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

- Affiche un inventaire spécifique :

```
ansible-inventory --list --inventory {{chemin/vers/  
fichier_ou_script_ou_dossier}}
```

- Affiche l'inventaire par défaut en YAML :

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

- Sauvegarde l'inventaire par défaut dans un fichier :

```
ansible-inventory --list --output {{chemin/vers/fichier}}
```



# ansible-playbook

Exécute les tâches définies dans le playbook sur les machines distantes via SSH.

Plus d'informations : <https://docs.ansible.com/ansible/latest/cli/ansible-playbook.html>.

- Exécute les tâches d'un playbook :

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

- Exécute les tâches d'un playbook avec fichier d'inventaire spécifié :

```
ansible-playbook {{playbook}} -i {{fichier_d_inventaire}}
```

- Exécute les tâches d'un playbook avec des variables supplémentaires définies via la ligne de commande :

```
ansible-playbook {{playbook}} -e "{{variable1}}={{valeur1}}  
{{variable2}}={{valeur2}}"
```

- Exécute les tâches d'un playbook avec des variables supplémentaires définies dans un fichier JSON :

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

- Exécute les tâches d'un playbook pour certain tags :

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

- Exécute les tâches d'un playbook en démarrant depuis une certaine tache :

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

# ansible-pull

Récupère les playbook ansible depuis un dépôt VCS et exécute les en local.

Plus d'informations : <https://docs.ansible.com/ansible/latest/cli/ansible-pull.html>.

- Récupère le playbook depuis un VCS et exécute le fichier par défaut local.yaml du playbook :

```
ansible-pull -U {{url_du_dépôt}}
```

- Récupère le playbook depuis un VCS et exécute un playbook spécifique :

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

- Récupère un playbook depuis un VCS sur une branche spécifique et exécute ce dernier :

```
ansible-pull -U {{url_du_dépôt}} -C {{branche}} {{playbook}}
```

- Récupère un playbook depuis un VCS, spécifie les fichiers hôtes et exécute un playbook spécifique :

```
ansible-pull -U {{url_du_dépôt}} -i {{fichier_hôtes}}  
{{playbook}}
```

# ansible-vault

Chiffre & déchiffre des valeurs, des structures de données et des fichiers dans un projet Ansible.

Plus d'informations : [https://docs.ansible.com/ansible/latest/user\\_guide/vault.html#id17](https://docs.ansible.com/ansible/latest/user_guide/vault.html#id17).

- Crée un nouveau fichier vault chiffré avec une invite à rentrer un mot passe :

```
ansible-vault create {{fichier_vault}}
```

- Crée un nouveau fichier vault chiffré avec un fichier clé vault pour le chiffrer :

```
ansible-vault create --vault-password-file  
{{fichier_de_mot_de_passe}} {{fichier_vault}}
```

- Chiffre un fichier existant avec un fichier de mot de passe optionnel :

```
ansible-vault encrypt --vault-password-file  
{{fichier_de_mot_de_passe}} {{fichier_vault}}
```

- Chiffre un texte avec le format de chiffage pour textes d'Ansible, en affichant une invite interactif :

```
ansible-vault encrypt_string
```

- Affiche un fichier chiffré, en utilisant un fichier de mot de passe pour le déchiffrer :

```
ansible-vault view --vault-password-file  
{{fichier_de_mot_de_passe}} {{fichier_vault}}
```

- Remplace le fichier de mot de passe d'un fichier vault déjà chiffré par un autre :

```
ansible-vault rekey --vault-password-file  
{{ancien_fichier_de_mot_de_passe}} --new-vault-password-file  
{{nouveau_fichier_de_mot_de_passe}} {{fichier_vault}}
```

# ansible

Gestionnaire de groupes d'ordinateurs à distance depuis SSH. (Utiliser le fichier `/etc/ansible/hosts` pour ajouter de nouveaux groupes/hôtes).

Certaines commandes comme **ansible galaxy** ont leur propre documentation.

Plus d'informations : <https://www.ansible.com/>.

- Lister les hôtes appartenant à un groupe :

```
ansible {{groupe}} --list-hosts
```

- Ping d'un groupe d'hôtes en invoquant le [m]odule "ping" :

```
ansible {{groupe}} -m ping
```

- Afficher des informations sur un groupe d'hôtes en invoquant le [m]odule "setup" :

```
ansible {{groupe}} -m setup
```

- Exécuter une commande sur un groupe d'hôtes en invoquant le [m]odule "command" avec en paramètre (a) cette commande :

```
ansible {{groupe}} -m command -a '{{ma_commande}}'
```

- Exécuter une commande avec des droits administrateur :

```
ansible {{groupe}} --become --ask-become-pass -m command -a '{{ma_commande}}'
```

- Exécuter une commande en utilisant un fichier d'inventaire personnalisé :

```
ansible {{groupe}} -i {{fichier_d'inventaire}} -m command -a '{{ma_commande}}'
```

- Lister les groupes d'un inventaire :

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

# ansiweather

Un script Shell pour afficher les conditions météo actuelles dans votre terminal.

Plus d'informations : <https://github.com/fcambus/ansiweather>.

- Affiche une prévision avec le système métrique pour les 7 prochains jours dans la ville de Paris, France :

```
ansiweather -u metric -f 7 -l {{Paris,FR}}
```

- Affiche une prévision avec des symboles et les données d'ensoleillement pour votre position actuelle :

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

- Affiche une prévision avec les données sur le vent et l'humidité pour votre position actuelle :

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

# ant

Apache Ant.

Outil pour construire et gérer les projets basés sur du Java.

Plus d'informations : <https://ant.apache.org>.

- Construit un projet avec le fichier de construction `build.xml` :

```
ant
```

- Construit un projet en utilisant un autre fichier que `build.xml` :

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

- Affiche les informations sur les cibles possibles pour ce projet :

```
ant -p
```

- Affiche les informations de débogage :

```
ant -d
```

- Exécute toutes les cibles qui ne dépendent pas d'une ou plusieurs cibles en erreur :

```
ant -k
```

# antibody

Le gestionnaire de modules shell "le plus rapide".

Plus d'informations : <https://getantibody.github.io>.

- Regroupe tous les modules pour un chargement statique :

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

- Mets à jour tous les modules :

```
antibody update
```

- Liste tous les modules installés :

```
antibody list
```

# anytopnm

Convertis n'importe quel type de fichier image vers un format d'image commun.

Plus d'informations : <https://netpbm.sourceforge.net/doc/anytopnm.html>.

- Convertis une image d'entrée au format PBM, PGM ou PPM, peut importe le type d'entrée :

```
anytopnm {{chemin/vers/entrée}} > {{chemin/vers/sortie.pnm}}
```

- Afficher la version :

```
anytopnm -version
```



# apg

Crée arbitrairement les mots de passe aléatoires et complexes.

Plus d'informations : <https://manned.org/apg>.

- Crée des mots de passe aléatoires (la longueur par défaut est 8) :

```
apg
```

- Crée un mot de passe avec au moins 1 symbole (S), 1 Nombre (N), 1 Majuscule (C), 1 Minuscule (L) :

```
apg -M SNCL
```

- Crée un mot de passe avec 16 caractères :

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

- Crée un mot de passe avec une longueur maximum de 16 :

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

- Crée un mot de passe qui n'apparaît pas dans le dictionnaire (le fichier de dictionnaire doit être donné) :

```
apg -r {{chemin/vers/fichier_dictionnaire}}
```

# apktool

Outil de Rétro-ingénierie pour les fichier APK.

Plus d'informations : <https://ibotpeaches.github.io/Apktool/>.

- Décode un fichier APK :

```
apktool d {{fichier.apk}}
```

- Construit un fichier APK depuis un dossier :

```
apktool b {{chemin/vers/un/dossier}}
```

- Installe et enregistre le framework :

```
apktool if {{framework.apk}}
```

# apm

Gestionnaire de paquets pour Atom.

Voir **atom**.

Plus d'informations : <https://github.com/atom/apm>.

- Installe un paquet depuis <http://atom.io/packages> ou un thème depuis <http://atom.io/themes> :

```
apm install {{nom_du_paquet}}
```

- Supprime un paquet ou un thème :

```
apm remove {{nom_du_paquet}}
```

- Mets à jour un paquet ou un thème :

```
apm upgrade {{nom_du_paquet}}
```

# apropos

Recherche dans les pages de manuel, par exemple pour trouver une nouvelle commande.

Plus d'informations : <https://manned.org/apropos>.

- Recherche par mot clé :

```
apropos {{expression_reguliere}}
```

- Recherche sans limiter la sortie à la largeur du terminal :

```
apropos -l {{expression_reguliere}}
```

- Recherche les pages qui contiennent toutes les expressions données (fonction ET) :

```
apropos {{expression_reguliere_1}} -a  
{{expression_reguliere_2}} -a {{expression_reguliere_3}}
```

# ar

Crée, modifie et extrait depuis des archives (**.a**, **.so**, **.o**).

Plus d'informations : <https://manned.org/ar>.

- E[x]trais tous les éléments depuis une archive :

```
ar x {{chemin/vers/archive.a}}
```

- Lis[t]e tous les éléments depuis une archive :

```
ar t {{chemin/vers/archive.ar}}
```

- [r]emplace ou ajoute des fichiers à une archive :

```
ar r {{chemin/vers/archive.deb}} {{chemin/vers/debian-binary  
chemin/vers/control.tar.gz chemin/vers/data.tar.xz ...}}
```

- In[s]ère un fichier d'indexation (équivalent à `ranlib`) :

```
ar s {{chemin/vers/archive.a}}
```

- Crée une archive avec des fichiers et un fichier d'indexation qui l'accompagne :

```
ar rs {{chemin/vers/archive.a}} {{chemin/vers/fichier1.o  
chemin/vers/fichier2.o ...}}
```

# arc

Arcanist: Une CLI pour Phabricator.

Plus d'informations : <https://secure.phabricator.com/book/phabricator/article/arcanist/>.

- Envoie les changements à un différentiel pour relecture :

```
arc diff
```

- Affiche les informations de la révision en suspens :

```
arc list
```

- Mets à jour les messages de commit Git après relecture :

```
arc amend
```

- Pousse les changements Git :

```
arc land
```

# arch

Affiche le nom de l'architecture système.

Voir aussi **uname**.

Plus d'informations : <https://www.gnu.org/software/coreutils/arch>.

- Affiche l'architecture système :

arch

# arduino-builder

Un outil en ligne de commande pour compiler des croquis arduino.

AVERTISSEMENT DE DÉPRÉCIATION: Cet outil a été retiré au profit de **arduino**.

Plus d'informations : <https://github.com/arduino/arduino-builder>.

- Construis un croquis :

```
arduino-builder -compile {{chemin/vers/croquis.ino}}
```

- Spécifie the niveau de débogage (1 à 10, 5 par défaut) :

```
arduino-builder -debug-level {{niveau}}
```

- Spécifie un dossier de construction :

```
arduino-builder -build-path {{chemin/vers/dossier/de/  
construction}}
```

- Utilise un fichier d'option de construction, au lieu de spécifier -hardware, -tools, etc. Manuellement à chaque fois :

```
arduino-builder -build-options-file {{chemin/vers/  
construction.options.json}}
```

- Active le mode verbeux :

```
arduino-builder -verbose {{true}}
```



# arduino

Arduino Studio - Environnement de Développement Intégré pour la plateforme Arduino.

Plus d'informations : <https://github.com/arduino/Arduino/blob/master/build/shared/manpage.adoc>.

- Construis un croquis :

```
arduino --verify {{chemin/vers/croquis.ino}}
```

- Construis et téléverse un croquis :

```
arduino --upload {{chemin/vers/croquis.ino}}
```

- Construis et téléverse un croquis vers un Arduino Nano avec un CPU Atmega328p, connecté sur le port /dev/ttyACM0 :

```
arduino --board {{arduino:avr:nano:cpu=atmega328p}} --port {{/dev/ttyACM0}} --upload {{chemin/vers/croquis.ino}}
```

- Configure la préférence nom à une valeur valeur :

```
arduino --pref {{nom}}={{valeur}}
```

- Construis un croquis, mets le résultat de ce dernier dans un dossier, et réutilise n'importe quelles versions précédentes dans ce dossier :

```
arduino --pref build.path={{chemin/vers/dossier/de/construction}} --verify {{chemin/vers/croquis.ino}}
```

- Sauvegarde toutes préférences (modifiées) dans un fichier preferences.txt :

```
arduino --save-prefs
```

# argocd

Interface en ligne de commande pour contrôler un serveur Argo CD.

Certaines sous-commandes comme **app** ont leur propre documentation d'utilisation.

Plus d'informations : <https://argo-cd.readthedocs.io/en/stable/user-guide/commands/argocd/>.

- Se connecter à un serveur Argo CD :

```
argocd login --insecure --username {{utilisateur}} --password  
{{mot_de_passe}} {{serveur_argocd:port}}
```

- Liste des applications :

```
argocd app list
```

# aria2

Cette commande est un alias de **aria2c**.

- Voir la documentation pour la commande :

```
tldr aria2c
```

# aria2c

Utilitaire de téléchargement.

Supporte HTTP(S), FTP, SFTP, BitTorrent, et Metalink.

Plus d'informations : <https://aria2.github.io>.

- Télécharge depuis une URI vers un fichier :

```
aria2c "{{url}}"
```

- Télécharge un fichier via l'url spécifié en choisissant le nom de ce dernier :

```
aria2c --out {{nom_de_fichier}} "{{url}}"
```

- Télécharge plusieurs fichiers (différents) en parallèle :

```
aria2c --force-sequential {{false}} "{{url1 url2 ...}}"
```

- Télécharge depuis plusieurs sources avec chaque URI pointant vers le même fichier :

```
aria2c "{{url1 url2 ...}}"
```

- Télécharge les URIs listées dans un fichier avec un nombre limité de téléchargements en parallèle :

```
aria2c --input-file={{nom_de_fichier}} --max-concurrent-downloads={{nombre_de_téléchargements}}
```

- Télécharge avec plusieurs connections :

```
aria2c --split {{nombre_de_connections}} "{{url}}"
```

- Téléchargement FTP avec nom d'utilisateur et mot de passe :

```
aria2c --ftp-user={{nom_d_utilisateur}} --ftp-passwd={{mot_de_passe}} "{{url}}"
```

- Limite la vitesse de téléchargement en octets/s :

```
aria2c --max-download-limit={{vitesse}} "{{url}}"
```

# arp-scan

Envoie des paquets ARP à des hôtes (spécifié via des adresses IP ou des noms de domaines) pour scanner le réseau local.

Plus d'informations : <https://github.com/royhills/arp-scan>.

- Scanne le réseau local actuel :

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

- Scanne un réseau IP pour un masque de bits donné :

```
arp-scan {{192.168.1.1}}/{{24}}
```

- Scanne un réseau IP dans une plage IP :

```
arp-scan {{127.0.0.0}}-{{127.0.0.31}}
```

- Scanne un réseau IP pour un masque de sous-réseaux donné :

```
arp-scan {{10.0.0.0}}:{{255.255.255.0}}
```

# arp

Affiche et manipule votre cache système ARP.

Plus d'informations : <https://manned.org/arp>.

- Affiche la table ARP courante :

```
arp -a
```

- Supprime une entrée spécifique :

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

- Crée une entrée dans la table ARP :

```
arp -s {{adresse}} {{adresse_mac}}
```

# arping

Découvre et sonde des hôtes dans un réseau en utilisant le protocole ARP.

Très utile pour la découverte d'adresse MAC.

Plus d'informations : <https://github.com/ThomasHabets/arping>.

- Ping un hôte par paquets de requête ARP :

```
arping {{ip_hôte}}
```

- Ping un hôte sur une interface spécifique :

```
arping -I {{interface}} {{ip_hôte}}
```

- Ping un hôte et arrête à la première réponse :

```
arping -f {{ip_hôte}}
```

- Ping un hôte un certain nombre de fois :

```
arping -c {{nombre_d_appel}} {{ip_hôte}}
```

- Diffuse les paquets de requête ARP pour mettre à jour les caches ARP voisin :

```
arping -U {{ip_à_diffuser}}
```

- Détecte les adresses IP dupliquées dans le réseau en envoyant les requêtes ARP avec une expiration de 3 secondes :

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

# asar

Un outil d'archivage pour la plateforme Electron.

Plus d'informations : <https://github.com/electron/asar>.

- Archive un fichier ou un dossier :

```
asar pack {{chemin/vers/fichier_ou_dossier}} {{archive.asar}}
```

- Extrais une archive :

```
asar extract {{archive.asar}}
```

- Extrais un fichier spécifique d'une archive :

```
asar extract-file {{archive.asar}} {{fichier}}
```

- Liste les éléments contenus dans une archive :

```
asar list {{archive.asar}}
```



# asciidoctor

Un processeur qui convertit des fichiers AsciiDoc vers un format publiable.

Plus d'informations : <https://docs.asciidoctor.org>.

- Convertis un fichier .adoc vers un fichier HTML (le format de sortie par défaut) :

```
asciidoctor {{chemin/vers/fichier.adoc}}
```

- Convertis un fichier .adoc vers un fichier HTML et lie une feuille de style CSS :

```
asciidoctor -a stylesheet {{chemin/vers/feuille_de_style.css}} {{chemin/vers/fichier.adoc}}
```

- Convertis un fichier .adoc vers un fichier HTML embarqué, en enlevant tout sauf le body :

```
asciidoctor --embedded {{chemin/vers/fichier.adoc}}
```

- Convertis un fichier .adoc vers un PDF en utilisant la librairie asciidoctor-pdf :

```
asciidoctor --backend {{pdf}} --require {{asciidoctor-pdf}} {{chemin/vers/fichier.adoc}}
```

# asciinema

Enregistre et rejoue les sessions de terminal, et également partageable sur [asciinema.org](https://asciinema.org).

Plus d'informations : <https://docs.asciinema.org/manual/cli/usage>.

- Associe l'installation locale de `asciinema` avec un compte `asciinema.org` :

```
asciinema auth
```

- Crée un nouvel enregistrement (une fois fini, l'utilisateur sera questionné pour l'enregistrer localement ou l'envoyer en ligne) :

```
asciinema rec
```

- Crée un nouvel enregistrement et l'enregistre dans un fichier local :

```
asciinema rec {{chemin/vers/enregistrement.cast}}
```

- Rejoue un enregistrement depuis un fichier local :

```
asciinema play {{chemin/vers/enregistrement.cast}}
```

- Rejoue un enregistrement depuis `asciinema.org` :

```
asciinema play https://asciinema.org/a/  
{{id_d_enregistrement}}
```

- Crée un nouvel enregistrement, en limitant le temps d'[i]nactivité au maximum à 2.5 secondes :

```
asciinema rec {{-i|--idle-time-limit}} 2.5
```

- Affiche la sortie complète d'un enregistrement local :

```
asciinema cat {{chemin/vers/enregistrement.cast}}
```

- Envoie un enregistrement local vers `asciinema.org` :

```
asciinema upload {{chemin/vers/enregistrement.cast}}
```

# asdf

Interface en ligne de commande pour gérer les versions de différents paquets.

Plus d'informations : <https://asdf-vm.com>.

- Liste toutes les extensions disponibles :

```
asdf plugin list all
```

- Installe une extension :

```
asdf plugin add {{nom}}
```

- Liste toutes les versions disponible d'un paquet :

```
asdf list all {{nom}}
```

- Installe une version spécifique d'un paquet :

```
asdf install {{nom}} {{version}}
```

- Fixe au global une version d'un paquet :

```
asdf global {{nom}} {{version}}
```

- Fix en local la version d'un paquet :

```
asdf local {{nom}} {{version}}
```

# assimp

Client en ligne de commande pour l'Open Asset Import Library.

Supporte le chargement de plus de 40 formats de fichiers 3D, et exporte vers quelques formats 3D populaire.

Plus d'informations : <https://assimp-docs.readthedocs.io/>.

- Liste tous les formats d'import supportés :

```
assimp listext
```

- Liste tous les formats de sortie supportés :

```
assimp listexport
```

- Convertis un fichier vers un format de sortie supporté, avec les paramètres par défaut :

```
assimp export {{fichier_d_entrée.stl}}  
{{fichier_de_sortie.obj}}
```

- Convertis un fichier avec des paramètres personnalisés (le fichier dox\_cmd.h dans le code source de assimp liste tous les paramètres disponible) :

```
assimp export {{fichier_d_entrée.stl}}  
{{fichier_de_sortie.obj}} {{paramètres}}
```

- Affiche un sommaire du contenu d'un fichier 3D :

```
assimp info {{chemin/vers/fichier}}
```

- Liste toutes les sous-commandes supportées ("mots") :

```
assimp help
```

- Affiche l'aide d'un sous-commande (e.g les paramètres qui lui sont spécifique) :

```
assimp {{sous_commande}} --help
```

# astronomer

Outil qui détecte les étoiles non-légitimes d'un robot sur les projets GitHub.

Plus d'informations : <https://github.com/Ullaakut/astronomer>.

- Analyse un dépôt :

```
astronomer {{tldr-pages/tldr-node-client}}
```

- Analyse le nombre maximum d'étoiles sur un dépôt :

```
astronomer {{tldr-pages/tldr-node-client}} --stars {{50}}
```

- Analyse un dépôt en affichant les rapports comparatifs :

```
astronomer {{tldr-pages/tldr-node-client}} --verbose
```

# astyle

Indente, formate, et embelli du code source pour des langages de programmation comme C, C++, C# et Java.

Pendant l'exécution, une copie du fichier original est créé avec un ".orig" suffixé au nom de fichier original.

Plus d'informations : <https://astyle.sourceforge.net>.

- Applique le style par défaut de 4 espaces pour l'indentation et pas de changement de formatage :

```
astyle {{fichier_source}}
```

- Applique le style Java avec le style attached :

```
astyle --style=java {{chemin/vers/fichier}}
```

- Applique le style allman :

```
astyle --style=allman {{chemin/vers/fichier}}
```

- Applique une indentation personnalisé avec des espaces. Choisi entre 2 et 20 espaces :

```
astyle --indent=spaces={{nombre_d_espaces}} {{chemin/vers/fichier}}
```

- Applique une indentation personnalisé avec des tabulations. Choisi entre 2 et 20 tabulations :

```
astyle --indent=tab={{nombre_de_tabulations}} {{chemin/vers/fichier}}
```

# at

Planifie l'exécution d'une commande une fois à un moment donné.

Le service atd (ou atrun) doit être actif pour l'exécution des commandes planifiées.

Plus d'informations : <https://manned.org/at>.

- Planifie l'exécution de la commande donnée dans l'entrée standard dans 5 minutes (Appuyer sur `Ctrl + D` une fois la commande inscrite) :

```
at now + 5 minutes
```

- Planifie l'exécution d'une commande depuis l'entrée standard (impression echo redirigée dans un tube) aujourd'hui à 10h00 :

```
echo "{{./ma_commande.sh}}" | at 1000
```

- Planifie l'exécution des commandes incluses dans un [f]ichier pour mardi prochain 21h30 :

```
at -f {{chemin/vers/fichier}} 9:30 PM Tue
```

# atom

Un éditeur de texte multiplateforme proposant de nombreuses extensions.

Les extensions sont gérées par **apm**.

Plus d'informations : <https://atom.io/>.

- Ouvrir un fichier ou un dossier :

```
atom {{chemin/vers/fichier_ou_dossier}}
```

- Ouvrir un fichier ou un dossier dans une nouvelle fenêtre :

```
atom -n {{chemin/vers/fichier_ou_dossier}}
```

- Ouvrir un fichier ou un dossier dans une fenêtre existante :

```
atom --add {{chemin/vers/fichier_ou_dossier}}
```

- Ouvrir en mode sans-échec (les extensions ne seront pas chargées) :

```
atom --safe
```

- Empêcher Atom de se lancer en arrière-plan, en le forçant à s'attacher au terminal :

```
atom --foreground
```

- Attendre la fermeture de la fenêtre avant de quitter (utile pour l'éditeur de commits Git) :

```
atom --wait
```



# atoum

Un framework de test unitaire pour PHP simple, moderne et intuitif.

Plus d'informations : <https://atoum.org>.

- Initialise un fichier de configuration :

```
atoum --init
```

- Lance les tests :

```
atoum
```

- Lance les tests avec un fichier de configuration donné :

```
atoum -c {{chemin/vers/fichier}}
```

- Lance un fichier de test spécifique :

```
atoum -f {{chemin/vers/fichier}}
```

- Lance les tests présent dans dossier donné :

```
atoum -d {{chemin/vers/dossier}}
```

- Lance tous les tests sous un certain namespace :

```
atoum -ns {{namespace}}
```

- Lance tous les tests avec un certain tag :

```
atoum -t {{tag}}
```

- Charge un fichier d'amorce avant de lancer les tests :

```
atoum --bootstrap-file {{chemin/vers/fichier}}
```

# atq

Affiche les travaux programmés par la commande **at** ou **batch**.

Plus d'informations : <https://manned.org/atq>.

- Affiche les travaux programmés de l'utilisateur courant :

```
atq
```

- Affiche les travaux de la file nommé 'a' (les files ont des noms avec une seule lettre) :

```
atq -q {{a}}
```

- Affiche les travaux de tous les utilisateurs (lancé en tant que super-utilisateur) :

```
sudo atq
```

# atrm

Supprime les travaux programmés par la commande **at** ou **batch**.

Pour retrouver les numéros des travaux, utilise **atq**.

Plus d'informations : <https://manned.org/atrm>.

- Supprime le travail numéro 10 :

```
atrm {{10}}
```

- Supprime plusieurs travaux, séparés par un espace :

```
atrm {{15}} {{17}} {{22}}
```

# atuin

Enregistre votre historique shell dans une base de donnée.

Peut également synchroniser votre historique chiffré entre plusieurs machines.

Plus d'informations : <https://atuin.sh/docs/commands>.

- Installe atuin dans votre shell :

```
eval "$(atuin init {{bash|zsh|fish}})"
```

- Importe l'historique du shell par défaut :

```
atuin import auto
```

- Recherche dans l'historique shell une commande spécifique :

```
atuin search {{commande}}
```

- Inscrire un compte sur le serveur de synchronisation par défaut :

```
atuin register -u {{nom_d_utilisateur}} -e {{email}} -p  
{{mot_de_passe}}
```

- Connexion au serveur de synchronisation par défaut :

```
atuin login -u {{nom_d_utilisateur}} -p {{mot_de_passe}}
```

- Synchronise l'historique avec le serveur :

```
atuin sync
```

# auditd

Réponds aux requêtes depuis l'outil d'audition et de notifications du kernel.

Il ne doit pas être utilisé manuellement.

Plus d'informations : <https://manned.org/auditd>.

- Démarre le démon :

```
auditd
```

- Démarre le démon en mode débogage :

```
auditd -d
```

- Démarre le démon à la demande depuis launchd :

```
auditd -l
```

# autoconf

Génère des scripts de configuration pour configurer automatiquement les paquets du code source.

Plus d'informations : <https://www.gnu.org/software/autoconf>.

- Génère un script de configuration depuis `configure.ac` (si présent) ou `configure.in` et le sauvegarde dans `configure` :

```
autoconf
```

- Génère un script de configuration depuis un modèle spécifié; et l'affiche la sortie standard :

```
autoconf {{fichier_de_template}}
```

- Génère un script de configuration depuis un modèle spécifié (même si the fichier d'entrée n'a pas changé) et l'écrit dans un fichier :

```
autoconf --force --output {{fichier_de_sortie}}  
{{fichier_de_template}}
```

# autoflake

Un outil pour enlever les imports et les variables inutilisés d'un code Python.

Plus d'informations : <https://github.com/myint/autoflake>.

- Enlève les variables non-utilisées d'un fichier et affiche la différence :

```
autoflake --remove-unused-variables {{chemin/vers/  
fichier.py}}
```

- Enlève les imports non-utilisés de plusieurs fichiers et affiche la différence :

```
autoflake --remove-all-unused-imports {{chemin/vers/  
fichier1.py chemin/vers/fichier2.py ...}}
```

- Enlève les variables non-utilisées d'un fichier, surcharge ce dernier :

```
autoflake --remove-unused-variables --in-place {{chemin/vers/  
fichier.py}}
```

- Enlève les variables non-utilisées de tous les fichiers d'un dossier de manière récursive, en les surchargeant :

```
autoflake --remove-unused-variables --in-place --recursive  
{{chemin/vers/dossier}}
```

# autojump

Accède rapidement aux dossiers que vous visitez le plus.

Les alias comme j or jc sont fournis pour simplifier leurs utilisation.

Plus d'informations : <https://github.com/wting/autojump>.

- Accède à un dossier qui contiens le motif suivant :

```
j {{motif}}
```

- Accède à un sous-dossier (enfant) du repertoire courant qui contiens the motif suivant :

```
jc {{motif}}
```

- Ouvre le dossier qui contiens the motif suivant dans le gestionnaire de fichier du système d'exploitation :

```
jo {{motif}}
```

- Enlève les dossiers qui n'existent plus de la base de données de autojump :

```
j --purge
```

- Affiche les entrées dans la base de données de autojump :

```
j -s
```



# autopep8

Formate du code Python en accord avec le style PEP 8.

Plus d'informations : <https://github.com/hhatto/autopep8>.

- Formate un fichier vers la sortie standard, avec une taille de ligne maximale personnalisée :

```
autopep8 {{chemin/vers/fichier.py}} --max-line-length  
{{longueur}}
```

- Formate un fichier, en affichant les changements :

```
autopep8 --diff {{chemin/vers/fichier}}
```

- Formate un fichier et sauvegarde les changements :

```
autopep8 --in-place {{chemin/vers/fichier.py}}
```

- Formate récursivement les fichiers dans un dossier et sauvegarde les changements :

```
autopep8 --in-place --recursive {{chemin/vers/dossier}}
```

# autossh

Lance, surveille et redémarre les connections SSH.

Reconnecte automatiquement pour garder le tunnel de transfert de port ouvert.  
Accepte toutes les options de SSH.

Plus d'informations : <https://www.harding.motd.ca/autossh>.

- Démarre une session SSH, redémarre quand le port échoue à renvoyer de la data :

```
autossh -M {{port_surveillé}} "{{commande_ssh}}"
```

- Fait suivre un port local vers un port distant, redémarre si nécessaire :

```
autossh -M {{port_surveillé}} -L {{port_local}}:localhost:  
{{port_distant}} {{utilisateur}}@{{hôte}}
```

- Diverge autossh en arrière plan avant de lancer SSH et n'ouvre pas de shell distant :

```
autossh -f -M {{port_surveillé}} -N "{{commande_ssh}}"
```

- Démarre en arrière plan, sans surveillance de port et à la place envoie des paquets SSH "keep-alive" toutes les 10 secondes pour détecter les échecs :

```
autossh -f -M 0 -N -o "ServerAliveInterval 10" -o  
"ServerAliveCountMax 3" "{{commande_ssh}}"
```

- Démarre en arrière plan, sans surveillance de port ni shell distant et s'arrête si le partage de port échoue :

```
autossh -f -M 0 -N -o "ServerAliveInterval 10" -o  
"ServerAliveCountMax 3" -o ExitOnForwardFailure=yes -L  
{{port_local}}:localhost:{{port_distant}} {{utilisateur}}  
@{{hôte}}
```

- Démarre en arrière plan, logue la sortie de débogage d'autossh et la sortie verbeuse de SSH dans des fichiers :

```
AUTOSSH_DEBUG=1 AUTOSSH_LOGFILE={{chemin/vers/  
fichier_logs_autossh.log}} autossh -f -M {{port_surveillé}} -  
v -E {{chemin/vers/fichier_logs_ssh.log}} {{commande_ssh}}
```

# avo

La ligne de commande officielle pour Avo.

Plus d'informations : <https://www.avo.app/docs/implementation/cli>.

- Initialise un espace de travail dans le dossier courant :

```
avo init
```

- Connecte la cli à la plateforme Avo :

```
avo login
```

- Change de branche vers une branche existante Avo :

```
avo checkout {{nom_de_branche}}
```

- Récupère les couvertures analytique pour le chemin courant :

```
avo pull
```

- Affiche le status de l'implémentation Avo :

```
avo status
```

- Résous les conflits Git dans les fichiers Avo :

```
avo conflict
```

- Ouvre l'espace de travail courant Avo dans le navigateur web par défaut :

```
avo edit
```

- Affiche l'aide pour une sous-commande :

```
avo {{sous_commande}} --help
```

# avrdude

Pilotes pour programmer les microcontrôleurs Atmel AVR.

Plus d'informations : <https://www.nongnu.org/avrdude/>.

- Lire le contenu du microcontrôleur AVR :

```
avrdude -p {{appareil_AVR}} -c {{programmeur}} -U flash:r:{{fichier.hex}}:i
```

- Programme le microcontrôleur AVR :

```
avrdude -p {{appareil_AVR}} -c {{programmeur}} -U flash:w:{{fichier.hex}}
```

- Affiche les appareils AVR disponibles :

```
avrdude -p \?
```

- Affiche les programmeurs AVR disponibles :

```
avrdude -c \?
```

# awk

Langage de programmation polyvalent pour travailler sur des fichiers.

Plus d'informations : <https://github.com/onetrueawk/awk>.

- Affiche la cinquième colonne (ou le champ) dans un fichier qui utilise des espaces comme séparateur :

```
awk '{print $5}' {{nom_de_fichier}}
```

- Affiche la deuxième colonne dans des lignes contenant "quelque-chose" dans un fichier qui utilise des espaces comme séparateur :

```
awk '/{{quelque-chose}}/ {print $2}' {{nom_de_fichier}}
```

- Affiche la dernière colonne de chaque ligne d'un fichier en utilisant une virgule (au lieu des espaces) comme séparateur :

```
awk -F ',' '{print $NF}' {{nom_de_fichier}}
```

- Additionne les valeurs de la première colonne des lignes d'un fichier et affiche le total :

```
awk '{s+=$1} END {print s}' {{nom_de_fichier}}
```

- Additionne les valeurs de la première colonne des lignes d'un fichier et affiche ces valeurs puis affiche le total :

```
awk '{s+=$1; print $1} END {print "-----"; print s}' {{nom_de_fichier}}
```

- Affiche une ligne sur trois en partant de la première ligne :

```
awk 'NR%3==1' {{nom_de_fichier}}
```

- Affiche les lignes dont la valeur de la colonne 10 vaut la valeur recherchée :

```
awk '($10 == valeur)'
```

- Affiche les lignes dont la valeur de la colonne 10 est comprise entre un min et un max :

```
awk '($10 >= valeur_min && $10 <= valeur_max)'
```

# aws cur

Crée, requête et supprime des rapports de définition d'utilisation AWS.

Plus d'informations : <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/cur/index.html>.

- Créé un rapport de définition de coût et d'utilisation AWS depuis un fichier JSON :

```
aws cur put-report-definition --report-definition file://  
{{chemin/vers/rapport_de_définition.json}}
```

- Liste les rapports de définition définis pour le compte connecté :

```
aws cur describe-report-definitions
```

- Supprime un rapport de définition d'utilisation :

```
aws cur --region {{region_aws}} delete-report-definition --  
report-name {{rapport}}
```

# aws ec2

CLI pour AWS EC2.

Provisionne, sécurise et des capacités de calcul redimensionnable dans le cloud AWS pour accélérer le développement et le déploiement d'applications.

Plus d'informations : <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/index.html>.

- Affiche les informations sur une instance spécifique :

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

- Affiche les informations de toutes les instances :

```
aws ec2 describe-instances
```

- Affiche les informations sur tous les volumes EC2 :

```
aws ec2 describe-volumes
```

- Supprime un volume EC2 :

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

- Crée une sauvegarde de votre volume EC2 :

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

- Liste toutes les AMIs (Images de Machine Amazon) disponible :

```
aws ec2 describe-images
```

- Affiche la liste de toutes les commandes EC2 disponible :

```
aws ec2 help
```

- Affiche l'aide pour une sous-commande EC2 spécifique :

```
aws ec2 {{sous-commande}} help
```

# aws ecr

Pousse, récupère et gère les images de conteneur.

Plus d'informations : <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ecr/index.html>.

- Connecte Docker avec le registre par défaut (le nom d'utilisateur est AWS) :

```
aws ecr get-login-password --region {{région}} | {{docker login}} --username AWS --password-stdin {{id_de_compte_aws}}.dkr.ecr.{{région}}.amazonaws.com
```

- Crée un dépôt :

```
aws ecr create-repository --repository-name {{dépôt}} --image-scanning-configuration scanOnPush={{true|false}} --region {{région}}
```

- Tag une image locale pour ECR :

```
docker tag {{nom_de_conteneur}}:{{tag}} {{id_de_compte_aws}}.dkr.ecr.{{région}}.amazonaws.com/{{nom_de_conteneur}}:{{tag}}
```

- Pousse une image dans le dépôt :

```
docker push {{id_de_compte_aws}}.dkr.ecr.{{région}}.amazonaws.com/{{nom_de_conteneur}}:{{tag}}
```

- Récupère une image depuis un dépôt :

```
docker pull {{id_de_compte_aws}}.dkr.ecr.{{région}}.amazonaws.com/{{nom_de_conteneur}}:{{tag}}
```

- Supprime une image d'un dépôt :

```
aws ecr batch-delete-image --repository-name {{dépôt}} --image-ids imageTag={{latest}}
```

- Supprime un dépôt :

```
aws ecr delete-repository --repository-name {{dépôt}} --force
```

- Liste les images dans un dépôt :

```
aws ecr list-images --repository-name {{dépôt}}
```



# aws glue

CLI pour AWS Glue.

Définie un endpoint publique pour le service AWS Glue.

Plus d'informations : <https://docs.aws.amazon.com/cli/latest/reference/glue/>.

- Liste les tâches :

```
aws glue list-jobs
```

- Démarre une tâche :

```
aws glue start-job-run --job-name {{nom_de_la_tâche}}
```

- Démarre un flux de travail :

```
aws glue start-workflow-run --name {{nom_flux_de_travail}}
```

- Liste les déclencheurs :

```
aws glue list-triggers
```

- Démarre un déclencheur :

```
aws glue start-trigger --name {{nom_du_déclencheur}}
```

- Créé un endpoint de développement :

```
aws glue create-dev-endpoint --endpoint-name {{nom}} --role-arn {{rôle_arn_utilisé_par_l_endpoint}}
```

# aws-google-auth

Outil en ligne de commande pour obtenir des credentials AWS temporaire (STS) en utilisant Google Apps comme un fournisseur de fédération (SSO).

Plus d'informations : <https://github.com/cevoaustralia/aws-google-auth>.

- Connecte l'utilisateur avec le SSO Google en utilisant les identifiants IDP et SP et donne une durée de vie d'une heure à la connexion :

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

- Connecte l'utilisateur en lui demandant quel rôle utiliser (dans le cas où il y a plusieurs rôles SAML) :

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

- Résous les alias pour des comptes AWS :

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

- Affiche l'aide :

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

# aws help

Affiche l'aide sur la CLI AWS.

Plus d'informations : <https://docs.aws.amazon.com/cli/latest/userguide/cli-usage-help.html>.

- Affiche l'aide :

```
aws help
```

- Liste tous les sujets disponibles :

```
aws help topics
```

- Affiche l'aide à propos d'un sujet particulier :

```
aws help {{nom_du_sujet}}
```

# aws history

Affiche l'historique de l'outil en ligne de commandes aws (l'enregistrement de l'historique de aws cli doit avoir été préalablement activé).

Plus d'informations : <https://docs.aws.amazon.com/cli/latest/reference/history/>.

- Liste l'historique des commandes avec leurs identifiants :

```
aws history list
```

- Affiche les événements liés à une commande spécifiée par son identifiant :

```
aws history show {{identifiant_commande}}
```

# aws iam

CLI pour AWS IAM.

Plus d'informations : <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/index.html>.

- Liste les utilisateurs :

```
aws iam list-users
```

- Liste les politiques :

```
aws iam list-policies
```

- Liste les groupes :

```
aws iam list-groups
```

- Récupère les utilisateurs dans un groupe :

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

- Décrit une politique IAM :

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

- Liste les clés d'accès :

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

- Liste les clés d'accès pour un utilisateur spécifique :

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

- Affiche la page d'aide pour `aws iam` (avec toutes les commandes iam disponibles) :

```
aws iam help
```

# aws kinesis

CLI officiel d'AWS pour les services de streaming d'Amazon Kinesis.

Plus d'informations : <https://docs.aws.amazon.com/cli/latest/reference/kinesis/index.html#cli-aws-kinesis>.

- Affiche tous les streams du compte :

```
aws kinesis list-streams
```

- Écris une entrée dans le stream Kinesis :

```
aws kinesis put-record --stream-name {{nom}} --partition-key  
{{clé}} --data {{message_encodé_en_base64}}
```

- Écris une entrée dans le stream Kinesis avec un encodage base64 inline :

```
aws kinesis put-record --stream-name {{nom}} --partition-key  
{{clé}} --data "$( echo "{{mon message}}" | base64 )"
```

- Liste tous les fragments disponible dans un stream :

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

- Récupère un fragment pour lire depuis le plus vieux message dans la stream de ce dernier :

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

- Lis les entrées d'un fragment en utilisant un itérateur de fragment :

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

# aws lambda

CLI pour AWS lambda.

Plus d'informations : <https://docs.aws.amazon.com/cli/latest/reference/lambda/>.

- Lance une fonction :

```
aws lambda invoke --function-name {{nom}} {{chemin/vers/la/réponse.json}}
```

- Lance une fonction avec pour donnée d'entrée, un document JSON :

```
aws lambda invoke --function-name {{nom}} --payload {{json}}  
{{chemin/vers/la/réponse.json}}
```

- Liste les fonctions :

```
aws lambda list-functions
```

- Affiche la configuration d'une fonction :

```
aws lambda get-function-configuration --function-name {{nom}}
```

- Affiche les alias d'une fonction :

```
aws lambda list-aliases --function-name {{nom}}
```

- Affiche la configuration de concurrence pour une fonction :

```
aws lambda get-function-concurrency --function-name {{nom}}
```

- Affiche quel service AWS peut appeler une fonction :

```
aws lambda get-policy --function-name {{nom}}
```

# aws quicksight

CLI pour AWS QuickSight.

Accès aux entrées QuickSight.

Plus d'informations : <https://docs.aws.amazon.com/cli/latest/reference/quicksight/>.

- Liste les datasets :

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

- Liste les utilisateurs :

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

- Liste les groupes :

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

- Liste les tableaux de bords :

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

- Affiche les informations détaillées sur un dataset :

```
aws quicksight describe-data-set --aws-account-id  
{{id_compte_aws}} --data-set-id {{id_data_set}}
```

- Affiche les personnes qui peuvent accéder au dataset et quelles actions ils peuvent effectuer sur ce dernier :

```
aws quicksight describe-data-set-permissions --aws-account-id  
{{id_compte_aws}} --data-set-id {{id_data_set}}
```



# aws rds

CLI AWS pour Relational Database Service.

Crée et gère des bases de données relationnelles.

Plus d'informations : <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/rds/index.html>.

- Affiche l'aide pour une sous-commande RDS donnée :

```
aws rds {{sous_commande}} help
```

- Stoppe une instance :

```
aws rds stop-db-instance --db-instance-identifier  
{{identifiant_de_l_instance}}
```

- Démarre une nouvelle instance :

```
aws rds start-db-instance --db-instance-identifier  
{{identifiant_de_l_instance}}
```

- Modifie une instance RDS :

```
aws rds modify-db-instance --db-instance-identifier  
{{identifiant_de_l_instance}} {{paramètres}} --apply-  
immediately
```

- Applique les mises à jour à une instance RDS :

```
aws rds apply-pending-maintenance-action --resource-  
identifier {{arn_de_la_base_de_données}} --apply-action  
{{mise_à_jour_du_système}} --opt-in-type {{immediate}}
```

- Change l'identifiant d'une instance :

```
aws rds modify-db-instance --db-instance-identifier  
{{ancien_identifiant_de_l_instance}} --new-db-instance-  
identifiant {{nouvel_identifiant_de_l_instance}}
```

- Redémarre une instance :

```
aws rds reboot-db-instance --db-instance-identifier  
{{identifiant_de_l_instance}}
```

- Supprime une instance :

```
aws rds delete-db-instance --db-instance-identifier  
{{identifiant_de_l_instance}} --final-db-snapshot-identifier  
{{identifiant_de_l_image}} --delete-automated-backups
```

# aws s3

CLI pour AWS S3 - fournis du stockage à travers les services web.

Plus d'informations : <https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/index.html>.

- Affiche les fichiers d'un bucket :

```
aws s3 ls {{nom_du_bucket}}
```

- Synchronise les fichiers et dossiers locaux avec un bucket :

```
aws s3 sync {{chemin/vers/les/fichiers}} s3://  
{{nom_du_bucket}}
```

- Synchronise les fichiers et dossiers d'un bucket avec le ceux en local :

```
aws s3 sync s3://{{nom_du_bucket}} {{chemin/vers/cible}}
```

- Synchronise les fichiers et les dossiers avec des exclusions :

```
aws s3 sync {{chemin/vers/les/fichiers}} s3://  
{{nom_du_bucket}} --exclude {{chemin/vers/le/fichier}} --  
exclude {{chemin/vers/le/dossier}}/*
```

- Supprime un fichier d'un bucket :

```
aws s3 rm s3://{{bucket}}/{{chemin/vers/le/fichier}}
```

- Prévisualise uniquement les changements :

```
aws s3 {{n_importe_quelle_commande}} --dryrun
```

# aws

La CLI officielle pour Amazon Web Services.

Certaines commandes comme **aws s3** ont leur propre documentation.

Plus d'informations : <https://aws.amazon.com/cli>.

- Configure la ligne de commande AWS :

```
aws configure wizard
```

- Configure la ligne de commande AWS en utilisant le SSO :

```
aws configure sso
```

- Récupère l'identité de l'appelant (utilisé pour déboguer les permissions) :

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

- Liste les ressources AWS d'une région et affiche le résultat en YAML :

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

- Utilise l'aide automatique au remplissage d'une commande :

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

- Utilise un guide interactif pour une ressource AWS :

```
aws dynamodb wizard {{nouvelle_table}}
```

- Génère un squelette CLI en JSON (utile pour l'Infrastructure as Code) :

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

- Voir l'aide pour une commande AWS :

```
aws {{commande}} help
```

# axel

Accélérateur de téléchargement.

Supporte HTTP, HTTPS, et FTP.

Plus d'informations : <https://github.com/axel-download-accelerator/axel>.

- Télécharge depuis une URL vers un fichier :

```
axel {{url}}
```

- Télécharge et spécifie le nom de fichier :

```
axel {{url}} -o {{nom_de_fichier}}
```

- Télécharge avec plusieurs connections :

```
axel -n {{nombre_de_connections}} {{url}}
```

- Recherche des miroirs :

```
axel -S {{nombre_de_miroirs}} {{url}}
```

- Limite la vitesse de téléchargement (en octets par secondes) :

```
axel -s {{vitesse}} {{url}}
```

# az

La CLI officielle de Microsoft Azure.

Certaines commandes comme **az login** ont leur propre documentation.

Plus d'informations : <https://learn.microsoft.com/cli/azure>.

- Connexion à Azure :

```
az login
```

- Gère les informations de souscription azure :

```
az account
```

- Affiche tous les disques managés Azure :

```
az disk list
```

- Liste toutes les machines virtuelles Azure :

```
az vm list
```

- Gère Azure Kubernetes Services :

```
az aks
```

- Gère les ressources réseaux Azure :

```
az network
```

# azure-cli

Cette commande est un alias de **az**.

Plus d'informations : <https://learn.microsoft.com/cli/azure>.

- Voir la documentation de la commande originale :

```
tldr az
```

# b2sum

Calcule les sommes de contrôle cryptographiques BLAKE2.

Plus d'informations : <https://www.gnu.org/software/coreutils/b2sum>.

- Calcule la somme de contrôle BLAKE2 d'un fichier :

```
b2sum {{chemin/vers/fichier}}
```

- Calcule les sommes de contrôle BLAKE2 pour plusieurs fichiers :

```
b2sum {{chemin/vers/fichier1 chemin/vers/fichier2 ...}} > {{chemin/vers/fichier.b2}}
```

- Calcule la somme de contrôle BLAKE2 depuis `stdin` :

```
{{commande}} | b2sum
```

- Lis un fichier contenant des sommes de contrôle BLAKE2 et des noms de fichier et vérifie que tous les fichiers ont des sommes de contrôle correspondantes :

```
b2sum --check {{chemin/vers/fichier.b2}}
```

- Affiche un message uniquement pour les fichiers manquants ou lorsque la vérification échoue :

```
b2sum --check --quiet {{chemin/vers/fichier.b2}}
```

- N'affiche un message que pour les fichiers pour lesquels la vérification a échoué, en ignorant les fichiers manquants :

```
b2sum --ignore-missing --check --quiet {{chemin/vers/fichier.b2}}
```



# b3sum

Calcule les sommes de contrôle cryptographiques BLAKE3.

Plus d'informations : <https://github.com/BLAKE3-team/BLAKE3/tree/master/b3sum>.

- Calcule la somme de contrôle BLAKE3 pour un ou plusieurs fichiers :

```
b3sum {{chemin/vers/fichier1 chemin/vers/fichier2 ...}}
```

- Calcule et enregistre la liste des sommes de contrôle BLAKE3 dans un fichier :

```
b3sum {{chemin/vers/fichier1 chemin/vers/fichier2 ...}} >  
{{chemin/vers/fichier.b3}}
```

- Calculer une somme de contrôle BLAKE3 à partir de `stdin` :

```
{{commande}} | b3sum
```

- Lit un fichier de sommes et de noms de fichiers BLAKE3 et vérifie que tous les fichiers ont des sommes de contrôle correspondantes :

```
b3sum --check {{chemin/vers/fichier.b3}}
```

- N'affiche un message que pour les fichiers manquants ou en cas d'échec de la vérification :

```
b3sum --check --quiet {{chemin/vers/fichier.b3}}
```

# babel

Un transpileur qui convertit du code JavaScript avec la syntaxe ES6/ES7 en syntaxe ES5.

Plus d'informations : <https://babeljs.io/>.

- Transpile un fichier spécifié et affiche le résultat dans la sortie standard :

```
babel {{chemin/vers/fichier}}
```

- Transpile un fichier spécifié et enregistre le résultat dans un autre fichier :

```
babel {{chemin/vers/fichier/d_entrée}} --out-file {{chemin/vers/fichier/de/sortie}}
```

- Transpile un fichier d'entrée à chaque fois qu'il est modifié :

```
babel {{chemin/vers/fichier/d_entrée}} --watch
```

- Transpile tout un dossier et ses fichiers :

```
babel {{chemin/vers/dossier}}
```

- Ignore des fichiers (séparés par une virgule) dans un dossier :

```
babel {{chemin/vers/dossier}} --ignore {{fichiers_ignorés}}
```

- Transpile et minifie la sortie :

```
babel {{chemin/vers/fichier/d_entrée}} --minified
```

- Sélectionne un lot de pré-configuration pour le formatage de sortie :

```
babel {{chemin/vers/fichier/d_entrée}} --presets {{pré-configurations}}
```

- Affiche toutes les options disponibles :

```
babel --help
```

# balena

Interagis avec balenaCloud, openBalena et l'API balena depuis la ligne de commande.

Plus d'informations : <https://www.balena.io/docs/reference/cli/>.

- Connexion à balenaCloud :

```
balena login
```

- Crée une application balenaCloud ou openBalena :

```
balena app create {{nom_d_application}}
```

- Affiche toutes les applications balenaCloud ou openBalena du compte :

```
balena apps
```

- Affiche tous les appareils associés au compte balenaCloud ou openBalena :

```
balena devices
```

- Flash une image balenaOS sur l'appareil local :

```
balena local flash {{chemin/vers/balenaos.img}} --drive  
{{localisation_de_l_appareil}}
```

# base32

Encode ou décode un fichier ou l'entrée standard vers ou depuis la base 32, et retourne le résultat à la sortie standard.

Plus d'informations : <https://manned.org/base32>.

- Encode un fichier :

```
base32 {{chemin/vers/fichier}}
```

- Envelopper la sortie codée à une largeur spécifique (0 désactive l'enveloppement) :

```
base32 {{-w|--wrap}} {{0|76|...}} {{chemin/vers/fichier}}
```

- Décode un fichier :

```
base32 {{-d|--decode}} {{chemin/vers/fichier}}
```

- Encode depuis `stdin` :

```
{{commande}} | base32
```

- Décode depuis `stdin` :

```
{{commande}} | base32 {{-d|--decode}}
```

# base64

Encode ou décode un fichier ou l'entrée standard vers ou depuis la base 64, et retourne le résultat à la sortie standard.

Plus d'informations : <https://manned.org/base64>.

- Encode un fichier :

```
base64 {{chemin/vers/fichier}}
```

- Envelopper la sortie codée à une largeur spécifique (0 désactive l'enveloppement) :

```
base64 {{-w|--wrap}} {{0|76|...}} {{chemin/vers/fichier}}
```

- Décode un fichier :

```
base64 {{-d|--decode}} {{chemin/vers/fichier}}
```

- Encode depuis `stdin` :

```
{{commande}} | base64
```

- Décode depuis `stdin` :

```
{{commande}} | base64 {{-d|--decode}}
```

# basename

Retourne la portion ne contenant pas de dossiers d'un chemin complet.

Plus d'informations : <https://www.gnu.org/software/coreutils/basename>.

- N'afficher que le nom du fichier depuis un chemin :

```
basename {{chemin/vers/fichier}}
```

- N'afficher que le nom du dernier répertoire depuis un chemin :

```
basename {{chemin/vers/répertoire/}}
```

- N'afficher que le nom du fichier depuis un chemin, en ôtant un préfixe donné :

```
basename {{chemin/vers/fichier}} {{suffixe}}
```

# bash

Bourne-Again SHell, un interpréteur de ligne de commande compatible avec **sh**.

Voir aussi **histexpand** pour l'expansion de l'historique.

Plus d'informations : <https://www.gnu.org/software/bash/>.

- Démarre une session shell interactive :

```
bash
```

- Exécute une commande, puis termine la session :

```
bash -c "{{commande}}"
```

- Exécute un script :

```
bash {{chemin/vers/le/script.sh}}
```

- Exécute un script en affichant chaque commande avant de l'exécuter :

```
bash -x {{chemin/vers/le/script.sh}}
```

- Exécute un script en s'arrêtant à la première erreur :

```
bash -e {{chemin/vers/le/script.sh}}
```

- Lit et exécute des commandes depuis l'entrée standard **stdin** :

```
bash -s
```

- Affiche la version de Bash (**\$BASH\_VERSION** ne contenant que la version, sans les informations de license):

```
bash --version
```

# bat

Affiche et concatène le contenu d'un ou plusieurs fichiers.

Un clone de **cat** avec mise en valeur de la syntaxe et intégration avec Git.

Plus d'informations : <https://github.com/sharkdp/bat>.

- Affiche le contenu d'un fichier sur la sortie standard :

```
bat {{fichier}}
```

- Concatène le contenu de plusieurs fichiers vers le fichier de destination :

```
bat {{fichier1}} {{fichier2}} > {{fichier_de_destination}}
```

- Ajoute le contenu d'un fichier à la fin du fichier de destination :

```
bat {{fichier1}} {{fichier2}} >> {{fichier_de_destination}}
```

- Numérote toutes les lignes affichées :

```
bat --number {{fichier}}
```

- Affiche le contenu d'un fichier JSON sur la sortie standard avec mise en valeur de la syntaxe :

```
bat --language json {{chemin/vers/fichier.json}}
```

- Affiche tous les langages pris en charge :

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



# bc

Un langage de calcul de précision arbitraire.

Voir aussi : **dc**.

Plus d'informations : <https://manned.org/bc>.

- Démarre une session interactive :

```
bc
```

- Démarre une session interactive avec la bibliothèque mathématique standard activée :

```
bc --mathlib
```

- Calcule une expression :

```
echo '{{5 / 3}}' | bc
```

- Exécute un script :

```
bc {{chemin/vers/le/script.bc}}
```

- Calcule une expression avec l'échelle spécifiée :

```
echo 'scale = {{10}}; {{5 / 3}}' | bc
```

- Calcule une fonction sinus/cosinus/arctangente/logarithme naturel/exponentielle en utilisant `mathlib` :

```
echo '{{s|c|a|l|e}}({{1}})' | bc --mathlib
```

# bg

Reprend l'exécution de tâches qui ont été suspendues (en utilisant **Ctrl + Z** par exemple) en arrière-plan.

Plus d'informations : <https://manned.org/bg>.

- Reprend l'exécution de la dernière tâche suspendue en arrière-plan :

```
bg
```

- Reprend l'exécution d'une tâche précise (utiliser `jobs -l` pour obtenir son ID) en arrière-plan :

```
bg %{{job_id}}
```

# borg

Outil de sauvegarde avec dé-duplication.

Crée des sauvegardes distantes ou locales qui peuvent être montées comme un système de fichiers.

Plus d'informations : <https://borgbackup.readthedocs.io/en/stable/usage/general.html>.

- Initialise un dépôt local :

```
borg init {/chemin/vers/repertoire_du_depot}}
```

- Sauvegarde un répertoire dans le dépôt en créant une archive appelée "Lundi" :

```
borg create --progress {/chemin/vers/repertoire_du_depot}::  
{{Lundi}} {/chemin/vers/repertoire_source}}
```

- Liste toutes les archives d'un dépôt :

```
borg list {/chemin/vers/repertoire_du_depot}}
```

- Extrait un répertoire donné de l'archive nommée "Lundi" à partir d'un dépôt distant tout en excluant tous les fichiers \*.ext :

```
borg extract {{utilisateur}}@{{hote}}:{/chemin/vers/  
repertoire_du_depot}::{{Lundi}} {chemin/vers/  
repertoire_destination}} --exclude '{{*.ext}}'
```

- Nettoie un dépôt en effaçant toutes les archives âgées de plus de 7 jours tout en affichant les changements :

```
borg prune --keep-within {{7d}} --list {/chemin/vers/  
repertoire_du_depot}}
```

- Monte un dépôt comme un système de fichiers FUSE :

```
borg mount {/chemin/vers/repertoire_du_depot}::{{Lundi}}  
{/chemin/vers/point_de_montage}}
```

- Affiche l'aide sur la création d'archives :

```
borg create --help
```

# brew bundle

Gestionnaire de paquets pour Homebrew, Homebrew Cask et le Mac App Store.

Plus d'informations : <https://github.com/Homebrew/homebrew-bundle>.

- Installe tous les paquets listés dans le Brewfile situé dans le dossier courant :

```
brew bundle
```

- Installe tous les paquets listés dans le Brewfile situé au chemin spécifié :

```
brew bundle --file {{chemin/vers/fichier}}
```

- Crée un Brewfile avec tous les paquets installés actuellement :

```
brew bundle dump
```

- Désinstalle tous les paquets non listés dans le Brewfile :

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

- Vérifie si il y a un ou plusieurs paquets à installer ou à mettre à jour depuis le Brewfile :

```
brew bundle check
```

- Affiche la liste de toutes les entrées dans un Brewfile :

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

# brew

Gestionnaire de paquets pour macOS et Linux.

Plus d'informations : <https://docs.brew.sh/Manpage>.

- Installe la dernière version stable d'une formule ou d'un cask (ajouter `--devel` pour une version de développement):

```
brew install {{formule}}
```

- Liste toutes les formules et les casks installés :

```
brew list
```

- Met à jour une formule ou cask déjà installé (si rien n'est précisé, toutes les formules et tous les casks installés seront mis à jour) :

```
brew upgrade {{formule}}
```

- Récupère la dernière version d'Homebrew et toutes les formules et casks depuis le dépôt source d'Homebrew :

```
brew update
```

- Montre les formules et les casks disposants d'une nouvelle version :

```
brew outdated
```

- Recherche une formule (c.a.d. un package) et/ou un cask (c.a.d une application native) :

```
brew search {{texte}}
```

- Affiche les informations d'une formule ou d'un cask (version, chemin d'installation, dépendances, etc.) :

```
brew info {{formule}}
```

- Vérifie que l'installation locale d'Homebrew n'a pas de problème :

```
brew doctor
```

# btop

Un moniteur de ressources qui affiche des informations sur le processeur, la mémoire, les disques, le réseau et les processus.

Une version C++ de **bpytop**.

Plus d'informations : <https://github.com/aristocratos/btop>.

- Démarre btop :

```
btop
```

- Démarre btop avec les réglages spécifiés :

```
btop --preset {{0..9}}
```

# bun

Moteur d'exécution et boîte à outils JavaScript.

Comprend un empaqueteur, un exécuteur de tests et un gestionnaire de paquets.

Plus d'informations : <https://bun.sh>.

- Exécute un fichier JavaScript ou un script référencé dans `package.json` :

```
bun run {{chemin/vers/fichier|nom_script}}
```

- Exécute les tests unitaires :

```
bun test
```

- Télécharge et installe tous les paquets listés comme dépendances dans `package.json` :

```
bun install
```

- Ajoute une dépendance à `package.json` :

```
bun add {{nom_module}}
```

- Supprime une dépendance de `package.json` :

```
bun remove {{nom_module}}
```

- Crée un nouveau projet Bun dans le répertoire actuel :

```
bun init
```

- Démarre un REPL (shell interactif) :

```
bun repl
```

- Met à jour Bun vers la dernière version :

```
bun upgrade
```

# bundle

Gestionnaire de dépendances pour le langage de programmation Ruby.

Plus d'informations : <https://bundler.io/man/bundle.1.html>.

- Installe toutes les gems définies dans le `Gemfile` attendu dans le répertoire de travail :

```
bundle install
```

- Exécute une commande dans le contexte du bundle actuel :

```
bundle exec {{commande}} {{arguments}}
```

- Mets à jour toutes les gems selon les règles définies dans le `Gemfile` et régénère le fichier `Gemfile.lock` :

```
bundle update
```

- Mets à jour une ou plusieurs gem(s) spécifique(s) définie(s) dans le `Gemfile` :

```
bundle update {{nom_de_la_gem}} {{nom_de_la_gem}}
```

- Mets à jour une ou plusieurs gem(s) spécifique(s) définie(s) dans le `Gemfile` mais seulement vers la prochaine version de patch :

```
bundle update --patch {{nom_de_la_gem}} {{nom_de_la_gem}}
```

- Mets à jour toutes les gem(s) du groupe donné dans le `Gemfile` :

```
bundle update --group {{nom_groupe}}
```

- Liste les gem(s) installée(s) dans le `Gemfile` avec les nouvelles versions disponibles :

```
bundle outdated
```

- Crée un nouveau squelette de gem :

```
bundle gem {{nom_de_la_gem}}
```



# bundler

Gestionnaire de dépendances pour le langage de programmation Ruby.

**bundler** est un nom commun pour la commande **bundle**, mais pas une commande elle-même.

Plus d'informations : <https://bundler.io/man/bundle.1.html>.

- Voir la documentation de la commande originale :

```
tldr bundle
```

# cargo add

Ajoute des dépendences au manifeste **Cargo.toml** d'un projet Rust.

Plus d'informations : <https://doc.rust-lang.org/cargo/commands/cargo-add.html>.

- Ajoute la dernière version d'une dépendance au projet courant :

```
cargo add {{dépendance}}
```

- Ajoute une version spécifique d'une dépendance :

```
cargo add {{dépendance}}@{{version}}
```

- Ajoute une dépendance et active une ou plusieurs fonctionnalités :

```
cargo add {{dépendance}} --features {{fonctionnalité_1}},  
{{fonctionnalité_2}}
```

- Ajoute une dépendance optionnelle, qui sera exposée comme une fonctionnalité du crate :

```
cargo add {{dépendance}} --optional
```

- Ajoute un crate local en tant que dépendance :

```
cargo add --path {{chemin/vers/crate}}
```

- Ajoute une dépendance de développement ou de compilation :

```
cargo add {{dépendance}} --{{dev|build}}
```

- Ajoute une dépendance avec toutes les fonctionnalités par défaut désactivées :

```
cargo add {{dépendance}} --no-default-features
```

# cargo bench

Compile et exécute des benchmarks.

Plus d'informations : <https://doc.rust-lang.org/cargo/commands/cargo-bench.html>.

- Exécute tous les benchmarks d'un paquet :

```
cargo bench
```

- Ne pas arrêter quand un benchmark échoue :

```
cargo bench --no-fail-fast
```

- Compile, mais n'exécute pas les benchmarks :

```
cargo bench --no-run
```

- Exécute le benchmark spécifié :

```
cargo bench --bench {{benchmark}}
```

- Exécute les benchmarks avec un profile spécifique (défaut : bench) :

```
cargo bench --profile {{profile}}
```

- Exécute les benchmarks sur tous les exemples cibles :

```
cargo bench --examples
```

- Exécute les benchmarks sur tous les binaires cibles :

```
cargo bench --bins
```

- Exécute les benchmarks sur la bibliothèque du paquet :

```
cargo bench --lib
```

# cargo build

Compile un paquet local et toutes ses dépendances.

Plus d'informations : <https://doc.rust-lang.org/cargo/commands/cargo-build.html>.

- Compile un ou plusieurs paquets définis dans le manifeste `Cargo.toml` dans le dossier local :

```
cargo build
```

- Compile les artefacts avec le mode publication, avec des optimisations :

```
cargo build --release
```

- Le fichier `Cargo.lock` doit être à jour :

```
cargo build --locked
```

- Compile tous les paquets de l'espace de travail :

```
cargo build --workspace
```

- Compile un paquet spécifique :

```
cargo build --package {{paquet}}
```

- Compile uniquement le binaire spécifié :

```
cargo build --bin {{nom_du_binaire}}
```

- Compile uniquement le test cible spécifié :

```
cargo build --test {{nom_du_test}}
```

# cargo init

Crée un nouveau paquet Cargo.

Équivalent de **cargo new**, mais où spécifier un dossier est optionnel.

Plus d'informations : <https://doc.rust-lang.org/cargo/commands/cargo-init.html>.

- Initialise un projet Rust binaire dans le dossier courant :

```
cargo init
```

- Initialise un projet de binaire Rust dans le dossier spécifié :

```
cargo init {{chemin/vers/dossier}}
```

- Initialise un projet de bibliothèque Rust dans le dossier spécifié :

```
cargo init --lib
```

- Initialise un dépôt de système de gestion de version dans le dossier du projet (défaut : git):

```
cargo init --vcs {{git|hg|pijul|fossil|none}}
```

- Définit le nom du paquet (défaut : nom du dossier):

```
cargo init --name {{nom_du_paquet}}
```

# cargo

Gestion d'un projet Rust et ses dependences (crates).

Certaines sous-commandes comme **build** ont leurs propres documentations.

Plus d'informations : <https://doc.rust-lang.org/cargo>.

- Rechercher des crates :

```
cargo search {{recherche}}
```

- Installer un crate :

```
cargo install {{nom_du_crate}}
```

- Lister les crates déjà installés :

```
cargo install --list
```

- Crée un nouveau binaire ou librairie du projet Rust dans les dossiers spécifiés (ou dans le dossier courant par défaut) :

```
cargo init --{{bin|lib}} {{chemin/vers/dossier}}
```

- Compile le projet Rust dans le dossier courant en utilisant le profil release :

```
cargo build --release
```

- Compiler le projet Rust dans le dossier courant en utilisant le compilateur nightly :

```
cargo +nightly build
```

- Compiler en utilisant un nombre spécifique de threads (par défaut on prend le nombre de coeurs du CPU) :

```
cargo build --jobs {{nombre_de_threads}}
```

# cat

Affiche et concatène le contenu d'un ou plusieurs fichiers.

Plus d'informations : <https://manned.org/cat.1posix>.

- Affiche le contenu d'un fichier sur la sortie standard :

```
cat {{chemin/vers/fichier}}
```

- Concatène le contenu de plusieurs fichiers vers le fichier de destination :

```
cat {{chemin/vers/fichier1 chemin/vers/fichier2 ...}} >  
{{chemin/vers/fichier_de_destination}}
```

- Ajoute le contenu d'un fichier à la fin du fichier de destination :

```
cat {{chemin/vers/fichier1 chemin/vers/fichier2 ...}} >>  
{{chemin/vers/fichier_de_destination}}
```

# cd

Modifie le répertoire de travail actuel.

Plus d'informations : <https://manned.org/cd>.

- Se déplace vers le répertoire donné :

```
cd {{chemin/vers/dossier}}
```

- Remonte vers le parent du répertoire actuel :

```
cd ..
```

- Se déplace vers le répertoire personnel de l'utilisateur actuel :

```
cd
```

- Se déplace vers le répertoire personnel de l'utilisateur donné :

```
cd ~{{nom_utilisateur}}
```

- Retourne au répertoire précédent :

```
cd -
```

- Se déplace vers le répertoire racine :

```
cd /
```



# chgrp

Change la propriété de groupe des fichiers et des répertoires.

Plus d'informations : <https://www.gnu.org/software/coreutils/chgrp>.

- Change le groupe propriétaire d'un fichier/répertoire :

```
chgrp {{groupe}} {{chemin/vers/fichier_ou_répertoire}}
```

- Change récursivement le groupe propriétaire d'un répertoire et de son contenu :

```
chgrp -R {{groupe}} {{chemin/vers/répertoire}}
```

- Change le groupe propriétaire d'un lien symbolique :

```
chgrp -h {{groupe}} {{chemin/vers/liens_symbolique}}
```

- Modifie le groupe propriétaire d'un fichier/répertoire pour qu'il corresponde à un fichier de référence :

```
chgrp --reference {{chemin/vers/fichier_référence}} {{chemin/vers/fichier_ou_répertoire}}
```

# chmod

Modifie les droits d'accès d'un fichier ou d'un répertoire.

Plus d'informations : <https://www.gnu.org/software/coreutils/chmod>.

- Donne les droits d'e[x]écution à l'[u]tilisateur auquel le fichier appartient :

```
chmod u+x {{chemin/vers/fichier}}
```

- Donne à l'[u]tilisateur les droits de lecture [r] et d'écriture [w] sur un fichier/répertoire :

```
chmod u+rw {{chemin/vers/fichier_ou_répertoire}}
```

- Enlève les droits d'e[x]écution pour le [g]roupe :

```
chmod g-x {{chemin/vers/fichier}}
```

- Donne à tous [a] les utilisateurs les droits de lecture [r] et d'e[x]écution :

```
chmod a+rx {{chemin/vers/fichier}}
```

- Donne aux autres [o] utilisateurs (qui sont dans un autre groupe) les mêmes droits que ceux du [g]roupe propriétaire :

```
chmod o=g {{chemin/vers/fichier}}
```

- Retire tous les droits aux autres [o] utilisateurs :

```
chmod o= {{chemin/vers/fichier}}
```

- Modifie les permissions récursivement en donnant aux membres du [g]roupe et aux autres [o] utilisateurs le droit d'écriture [w] :

```
chmod -R g+w,o+w {{chemin/vers/répertoire}}
```

- Donne récursivement à tous [a] les utilisateurs les droits de lecture [r] de fichiers et d'e[X]écution de sous-répertoires à l'intérieur d'un répertoire :

```
chmod -R a+rX {{chemin/vers/répertoire}}
```

# chown

Modifie l'utilisateur et le groupe propriétaire des fichiers et dossiers.

Plus d'informations : <https://www.gnu.org/software/coreutils/chown>.

- Modifie le propriétaire d'un fichier/dossier :

```
chown {{utilisateur}} {{chemin/vers/fichier_ou_dossier}}
```

- Modifie l'utilisateur et le groupe propriétaire d'un fichier/dossier :

```
chown {{utilisateur}}:{{groupe}} {{chemin/vers/fichier_ou_dossier}}
```

- Modifie le propriétaire et le groupe pour qu'ils aient tous les deux le nom utilisateur :

```
chown {{utilisateur}}: {{chemin/vers/fichier_ou_dossier}}
```

- Modifie récursivement le propriétaire d'un dossier et de son contenu :

```
chown -R {{utilisateur}} {{chemin/vers/dossier}}
```

- Modifie le propriétaire d'un lien symbolique :

```
chown -h {{utilisateur}} {{chemin/vers/liensymbolique}}
```

- Modifie le propriétaire d'un fichier / dossier pour correspondre à un fichier de référence :

```
chown --reference {{chemin/vers/fichier_de_référence}}  
{{chemin/vers/fichier_ou_dossier}}
```

# chromium

Navigateur Web open source principalement développé et maintenu par Google.

Plus d'informations : <https://www.chromium.org/developers/how-tos/run-chromium-with-flags/>.

- Ouvre une URL ou un fichier spécifique :

```
chromium {{https://exemple.com|chemin/vers/fichier.html}}
```

- Ouvre en mode navigation privée :

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

- Ouvre dans une nouvelle fenêtre :

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

- Ouvre en mode application (sans barres d'outils, barre d'URL, boutons, etc) :

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

- Utilise un serveur proxy :

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

- Ouvre dans un répertoire de profil personnalisé :

```
chromium --user-data-dir={{chemin/vers/répertoire}}
```

- Ouvre sans validation CORS (utile pour tester une API) :

```
chromium --user-data-dir={{chemin/vers/répertoire}} --  
disable-web-security
```

- Ouvre avec une fenêtre outils de développement pour chaque onglet ouvert :

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

# chroot

Exécute une commande ou un shell interactif avec un répertoire racine spécial.

Plus d'informations : <https://www.gnu.org/software/coreutils/chroot>.

- Exécute la commande en tant que nouveau répertoire racine :

```
chroot {{chemin/vers/nouveau/répertoire/racine}} {{command}}
```

- Spécifie l'utilisateur et le groupe (ID ou nom) à utiliser :

```
chroot --userspec={{utilisateur:groupe}}
```

# chsh

Change le shell de connexion de l'utilisateur.

Plus d'informations : <https://manned.org/chsh>.

- Change le shell de connexion de l'utilisateur actuel de manière interactive :

```
chsh
```

- Change le shell de connexion de l'utilisateur actuel :

```
chsh -s {{chemin/vers/shell}}
```

- Change le shell de connexion pour un utilisateur donné :

```
chsh -s {{chemin/vers/shell}} {{nom_utilisateur}}
```

- Liste les shells disponibles :

```
chsh -l
```

# clang-cpp

Cette commande est un alias de **clang++**.

- Voir la documentation de la commande originale :

```
tldr clang++
```

# clear

Efface l'écran du terminal.

Plus d'informations : <https://manned.org/clear>.

- Effacer l'écran (identique à la séquence Contrôle-L sur une interface Bash) :

```
clear
```

- Effacer l'écran mais conserve le tampon de défilement du terminal :

```
clear -x
```

- Indiquer le type de terminal à effacer (utilise par défaut la variable d'environnement `TERM`) :

```
clear -T {{type_de_terminal}}
```

- Afficher la version de `ncurses` utilisée par `clear` :

```
clear -V
```



# clojure

Cette commande est un alias de **clj**.

- Voir la documentation de la commande originale :

```
tldr clj
```

# cmake

Système de construction logicielle multiplateforme, qui permet de générer des recettes de construction pour les systèmes de construction natifs.

Plus d'informations : <https://cmake.org/cmake/help/latest/manual/cmake.1.html>.

- Génère une recette de construction `CMakeLists.txt` depuis le répertoire d'un projet :

```
cmake {{chemin/vers/le/répertoire_du_projet}}
```

- Génère une recette de construction, en définissant le type de construction à `Release` à l'aide d'une variable `CMake` :

```
cmake {{chemin/vers/le/répertoire_du_projet}} -D  
{{CMAKE_BUILD_TYPE=Release}}
```

- Utilise une recette déjà générée dans un répertoire donné pour construire les artefacts :

```
cmake --build {{chemin/vers/le/répertoire_de_construction}}
```

- Installe les artefacts de construction dans `/usr/local/` et retire les symboles de débogage :

```
cmake --install {{chemin/vers/le/répertoire_de_construction}}  
--strip
```

- Installe les artefacts de construction en utilisant un préfixe personnalisé pour les chemins :

```
cmake --install {{chemin/vers/le/répertoire_de_construction}}  
--strip --prefix {{chemin/vers/le/répertoire}}
```

- Lance une cible de construction personnalisée :

```
cmake --build {{chemin/vers/le/répertoire_de_construction}}  
--target {{nom_de_la_cible}}
```

# code

Éditeur de code multiplateforme et extensible.

Plus d'informations : <https://github.com/microsoft/vscode>.

- Démarre Visual Studio Code :

```
code
```

- Ouvre des fichiers/répertoires spécifiques :

```
code {{chemin/vers/fichier_ou_répertoire1 chemin/vers/fichier_ou_répertoire2 ...}}
```

- Compare deux fichiers spécifiques :

```
code --diff {{chemin/vers/fichier1}} {{chemin/vers/fichier2}}
```

- Ouvre des fichiers/répertoires spécifiques dans une nouvelle fenêtre :

```
code --new-window {{chemin/vers/fichier_ou_répertoire1 chemin/vers/fichier_ou_répertoire2 ...}}
```

- Installe/désinstalle une extension spécifique :

```
code --{{install|uninstall}}-extension {{éditeur.extension}}
```

- Affiche les extensions installées :

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

- Affiche les extensions installées avec leurs versions :

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

- Démarre l'éditeur en tant que super utilisateur (root) tout en stockant les données utilisateur dans un répertoire spécifique :

```
sudo code --user-data-dir {{chemin/vers/répertoire}}
```

# cola

Cette commande est un alias de **git-cola**.

- Voir la documentation de la commande originale :

`tldr git-cola`

# colorls

Embellit la sortie de la commande `ls`, avec des icônes coloreés ou provenant de font-awesome. Disponible sous forme de gemme Ruby.

Plus d'informations : <https://github.com/athityakumar/colorls>.

- Liste les fichiers, un par ligne :

```
colorls -l
```

- Liste tous les fichiers, y compris les fichiers cachés :

```
colorls --all
```

- Liste au format long (autorisations, propriété, taille et date de modification) de tous les fichiers :

```
colorls --long --all
```

- Ne liste que les répertoires :

```
colorls --dirs
```

# cp

Copie des fichiers et des répertoires.

Plus d'informations : <https://www.gnu.org/software/coreutils/cp>.

- Copier un fichier vers un autre emplacement :

```
cp {{chemin/vers/fichier_source.ext}} {{chemin/vers/
fichier_cible.ext}}
```

- Copier un fichier vers un autre répertoire en conservant le nom du fichier :

```
cp {{chemin/vers/fichier_source.ext}} {{chemin/vers/
répertoire_parent_cible}}
```

- Copier récursivement le contenu d'un répertoire vers un autre emplacement (si la destination existe, le répertoire est copié à l'intérieur) :

```
cp -R {{chemin/vers/répertoire_source}} {{chemin/vers/
répertoire_cible}}
```

- Copier un répertoire récursivement, en mode verbeux (affiche les fichiers au fur et à mesure de leur copie) :

```
cp -vR {{chemin/vers/répertoire_source}} {{chemin/vers/
répertoire_cible}}
```

- Copier les fichiers texte vers un autre emplacement, en mode interactif (demande confirmation avant d'écraser) :

```
cp -i {{*.txt}} {{chemin/vers/répertoire_cible}}
```

- Déréférencer les liens symboliques avant de copier :

```
cp -L {{link}} {{chemin/vers/répertoire_cible}}
```

# crystal

Outil de gestion du code source de Crystal.

Plus d'informations : [https://crystal-lang.org/reference/using\\_the\\_compiler](https://crystal-lang.org/reference/using_the_compiler).

- Exécute un fichier Crystal :

```
crystal {{chemin/vers/fichier.cr}}
```

- Compile un fichier et toutes ses dépendances en un seul exécutable :

```
crystal build {{chemin/vers/fichier.cr}}
```

- Lit le code source Crystal à partir de la ligne de commande ou de `stdin`, et l'exécute :

```
crystal eval '{{code}}'
```

- Génère la documentation de l'API à partir commentaires dans les fichiers Crystal :

```
crystal docs
```

- Compile et exécute une suite de spécifications Crystal :

```
crystal spec
```

- Démarre un serveur interactif local pour tester le langage :

```
crystal play
```

- Crée un répertoire de projet pour une application Crystal :

```
crystal init app {{nom_application}}
```

- Affiche toutes les options d'aide :

```
crystal help
```

# csch

Le shell (interpréteur de commandes) affiche une syntaxe proche de C.

Voir aussi : **tcsh**.

Plus d'informations : <https://www.mkssoftware.com/docs/man1/csh.1.asp>.

- Démarrer une session interactive :

```
csch
```

- Démarrer une session interactive sans prendre en compte le fichier de configuration au démarrage :

```
csch -f
```

- Exécuter une commande :

```
csch -c "{{echo 'Exécution de la commande echo par csh'}}"
```

- Exécuter un script :

```
csch {{chemin/vers/script.csh}}
```



# curl

Transfère des données depuis ou vers un serveur.

Accepte la plupart des protocoles, notamment HTTP, FTP et POP3.

Plus d'informations : <https://curl.se/docs/manpage.html>.

- Télécharger le contenu d'une URL dans un fichier :

```
curl {{http://exemple.fr}} --output {{nom_fichier}}
```

- Télécharger le contenu d'une URL dans un fichier nommé comme indiqué par l'URL :

```
curl --remote-name {{http://exemple.fr/nom_fichier}}
```

- Télécharger un fichier, en suivant les redirections, et poursuivre (reprendre) automatiquement un transfert de fichier précédent et renvoyer une erreur lors d'erreurs serveurs :

```
curl --fail --remote-name --location --continue-at -  
{{http://exemple.fr/nom_fichier}}
```

- Envoyer des données de formulaire encodées (requête POST de type application/x-www-form-urlencoded). Utiliser --data @file\_name ou --data @'-' pour lire depuis stdin :

```
curl --data {{'nom=bob'}} {{http://exemple.fr/formulaire}}
```

- Envoyer une requête avec un en-tête supplémentaire, en spécifiant la méthode HTTP :

```
curl --header {{'X-Mon-En-Tete: 123'}} --request {{PUT}}  
{{http://exemple.fr}}
```

- Envoyez des données au format JSON, en spécifiant l'en-tête content-type adéquate :

```
curl --data {{'{"nom":"bob"}'}} --header {{'Content-Type:  
application/json'}} {{http://exemple.fr/utilisateurs/1234}}
```

- Fournir un nom d'utilisateur et demander pour un mot de passe pour une authentification auprès du serveur :

```
curl --user {{identifiant}} {{http://exemple.fr}}
```

- Fournir le certificat et la clé du client pour une ressource, en évitant la validation du certificat :

```
curl --cert {{client.pem}} --key {{cle.pem}} --insecure  
{{https://exemple.fr}}
```

# dart

Ligne de commande pour gérer un projet Dart.

Plus d'informations : <https://dart.dev/tools/dart-tool>.

- Initialise un nouveau projet Dart dans un dossier du même nom :

```
dart create {{nom_du_projet}}
```

- Exécuter un fichier Dart :

```
dart run {{chemin/vers/fichier.dart}}
```

- Télécharger les dépendances pour le projet courant :

```
dart pub get
```

- Exécuter les tests unitaire pour le projet courant :

```
dart test
```

- Mettre à jour les dépendances d'un projet pour supporter null-safety :

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

- Compiler un fichier Dart vers un binaire natif :

```
dart compile exe {{chemin/vers/fichier.dart}}
```

# dash

Debian Almquist SHell, une implémentation de **sh** moderne, conforme à POSIX (non compatible avec Bash).

Plus d'informations : <https://manned.org/dash>.

- Démarre une session shell interactive :

```
dash
```

- Exécute une commande, puis termine la session :

```
dash -c "{{commande}}"
```

- Exécute un script :

```
dash {{chemin/vers/le/script.sh}}
```

- Exécute un script en affichant chaque commande avant de l'exécuter :

```
dash -x {{chemin/vers/le/script.sh}}
```

- Exécute un script en s'arrêtant à la première erreur :

```
dash -e {{chemin/vers/le/script.sh}}
```

- Lit et exécute des commandes depuis l'entrée standard `stdin` :

```
dash -s
```

# deluge

Client BitTorrent à base de ligne de commande.

Plus d'informations : <https://deluge-torrent.org>.

- Télécharge un torrent :

```
deluge {{url|magnet|chemin/vers/fichier}}
```

- Télécharge un torrent à l'aide d'un fichier de configuration particulier :

```
deluge -c {{chemin/vers/fichier_configuration}} {{url|magnet|chemin/vers/fichier}}
```

- Télécharge un torrent et lance un interface usager particulier :

```
deluge -u {{gtk|web|console}} {{url|magnet|chemin/vers/fichier}}
```

- Télécharge un torrent et enregistre les journaux dans un fichier :

```
deluge -l {{chemin/vers/fichier_journalisation}} {{url|magnet|chemin/vers/fichier}}
```

# deluged

Un processus démon pour le client BitTorrent Deluge.

Plus d'informations : <https://deluge-torrent.org>.

- Lance le démon Deluge :

```
deluged
```

- Lance le démon Deluge sur un port spécifique :

```
deluged -p {{port}}
```

- Lance le démon Deluge à l'aide d'un fichier de configuration spécifique :

```
deluged -c {{chemin/vers/fichier_configuration}}
```

- Lance le démon Deluge et enregistre les journaux dans un fichier :

```
deluged -l {{chemin/vers/fichier_journalisation}}
```

# deno

Un environnement d'exécution sécurisé pour JavaScript et TypeScript.

Plus d'informations : <https://deno.land>.

- Exécute un fichier JavaScript ou TypeScript :

```
deno run {{chemin/du/fichier.ts}}
```

- Démarre un REPL (shell interactif) :

```
deno
```

- Exécute un fichier avec l'accès au réseau activé :

```
deno run --allow-net {{chemin/du/fichier.ts}}
```

- Exécute un fichier à partir d'une URL :

```
deno run {{https://deno.land/std/examples/welcome.ts}}
```

- Installe un script exécutable à partir d'une URL :

```
deno install {{https://deno.land/std/examples/colors.ts}}
```

# df

Montre un aperçu de l'utilisation de l'espace disque.

Plus d'informations : <https://manned.org/df.1posix>.

- Afficher tous les systèmes de fichiers et leur utilisation d'espace disque :

```
df
```

- Afficher tous les systèmes de fichiers et leur utilisation d'espace disque dans un format plus facilement :

```
df -h
```

- Afficher le système de fichiers et son utilisation d'espace disque rattaché au chemin donné :

```
df {{chemin/vers/fichier_ou_dossier}}
```

- Afficher des statistiques sur le nombre d'inodes disponibles :

```
df -i
```

- Afficher les systèmes de fichiers sauf ceux de types spécifiques :

```
df -x {{squashfs}} -x {{tmpfs}}
```



# diff

Compare deux fichiers ou répertoires.

Plus d'informations : <https://manned.org/diff>.

- Compare deux fichiers (liste les changements pour transformer ancien\_fichier en nouveau\_fichier):

```
diff {{ancien_fichier}} {{nouveau_fichier}}
```

- Compare deux fichiers, en ignorant l'espacement :

```
diff --ignore-all-space {{ancien_fichier}}  
{{nouveau_fichier}}
```

- Compare deux fichiers, en affichant différences côte à côte :

```
diff --side-by-side {{ancien_fichier}} {{nouveau_fichier}}
```

- Compare deux fichiers, en affichant les différences dans un format unifié (comme git diff):

```
diff --unified {{ancien_fichier}} {{nouveau_fichier}}
```

- Compare récursivement deux répertoires directories (affiche les noms de fichiers et de répertoires divergents ainsi que les modifications de fichiers) :

```
diff --recursive {{ancien_répertoire}} {{nouveau_répertoire}}
```

- Compare deux répertoires, en affichant uniquement les fichiers qui diffèrent :

```
diff --recursive --brief {{ancien_répertoire}}  
{{nouveau_répertoire}}
```

- Crée un fichier patch des différences entre deux fichiers texte pour Git, en traitant les fichiers inexistants comme fichiers vides :

```
diff --text --unified --new-file {{ancien_fichier}}  
{{nouveau_fichier}} > {{diff.patch}}
```

# docker build

Construit une image à partir d'un Dockerfile.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/buildx/build/>.

- Construire une image Docker en utilisant le Dockerfile du répertoire courant :

```
docker build .
```

- Construire une image Docker à partir d'un Dockerfile situé à une URL précisée :

```
docker build {{github.com/creack/docker-firefox}}
```

- Construire une image Docker et l'étiquette :

```
docker build --tag {{nom:etiquette}} .
```

- Construit une image Docker sans contexte de construction :

```
docker build --tag {{nom:etiquette}} - < {{Dockerfile}}
```

- Ne pas utiliser le cache lors de la construction de l'image :

```
docker build --no-cache --tag {{nom:etiquette}} .
```

- Construire une image Docker utilisant un Dockerfile spécifique :

```
docker build --file {{Dockerfile}} .
```

- Construire avec des variables personnalisées définies à la volée :

```
docker build --build-arg {{HTTP_PROXY=http://  
10.20.30.2:1234}} --build-arg {{FTP_PROXY=http://  
40.50.60.5:4567}} .
```

# docker commit

Créer une nouvelle image depuis les changements d'un conteneur.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/container/commit/>.

- Créer une image à partir d'un conteneur spécifique :

```
docker commit {{conteneur}} {{image}}:{{etiquette}}
```

- Appliquer une instruction CMD du Dockerfile à l'image créée :

```
docker commit --change "CMD {{commande}}" {{conteneur}}  
{{image}}:{{etiquette}}
```

- Appliquer une instruction ENV du Dockerfile à l'image créée :

```
docker commit --change "ENV {{name}}={{value}}" {{conteneur}}  
{{image}}:{{etiquette}}
```

- Créer une image avec un auteur spécifique dans les métadonnées :

```
docker commit --author "{{auteur}}" {{conteneur}} {{image}}:  
{{etiquette}}
```

- Créer une image avec un commentaire spécifique dans les métadonnées :

```
docker commit --message "{{commentaire}}" {{conteneur}}  
{{image}}:{{etiquette}}
```

- Créer une image sans mettre en pause le conteneur pendant la création :

```
docker commit --pause {{false}} {{conteneur}} {{image}}:  
{{etiquette}}
```

- Afficher l'aide :

```
docker commit --help
```

# docker compose

Exécute et gère des applications au travers de plusieurs conteneurs Docker.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/compose/>.

- Liste tous les conteneurs en cours d'exécution :

```
docker compose ps
```

- Crée et démarre en arrière-plan tous les conteneurs décrits dans le fichier `docker-compose.yml` du répertoire courant :

```
docker compose up --detach
```

- Démarre tous les conteneurs après les avoir recréés si nécessaire :

```
docker compose up --build
```

- Démarre tous les conteneurs en spécifiant un nom de projet et un fichier compose alternatif :

```
docker compose -p {{nom_de_projet}} --file {{chemin/vers/fichier}} up
```

- Arrête tous les conteneurs en cours d'exécution :

```
docker compose stop
```

- Arrête et supprime tous les conteneurs, réseaux, images et volumes :

```
docker compose down --rmi all --volumes
```

- Affiche et suit la journalisation de tous les conteneurs :

```
docker compose logs --follow
```

- Affiche et suit la journalisation pour un conteneurs spécifique :

```
docker compose logs --follow {{nom_container}}
```

# docker container

Gère les conteneurs Docker.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/container/>.

- Liste les conteneurs Dockers en cours d'exécution :

```
docker container ls
```

- Démarre un ou plusieurs conteneur arrêtés :

```
docker container start {{nom_conteneur_1}}  
{{nom_conteneur_2}}
```

- Tue un ou plusieurs conteneurs en cours d'exécution :

```
docker container kill {{nom_conteneur}}
```

- Arrête un ou plusieurs conteneurs en cours d'exécution :

```
docker container stop {{nom_conteneur}}
```

- Mets en pause tous les processus d'un ou plusieurs conteneurs :

```
docker container pause {{nom_conteneur}}
```

- Affiche des informations détaillées sur un ou plusieurs conteneurs :

```
docker container inspect {{nom_conteneur}}
```

- Exporte le système de fichiers d'un conteneur sous forme d'archive Tar :

```
docker container export {{nom_conteneur}}
```

- Crée une nouvelle image à partir des changements d'un conteneur :

```
docker container commit {{nom_conteneur}}
```

# docker cp

Copier des fichiers ou des répertoires entre les systèmes de fichiers hôte et conteneur.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/container/cp/>.

- Copier un fichier ou un répertoire de l'hôte vers un conteneur :

```
docker cp {{chemin/vers/le_fichier_ou_le_dossier_de_l_hote}}  
{{nom_du_conteneur}}:{{chemin/vers/  
le_fichier_ou_le_dossier_de_conteneur}}
```

- Copier un fichier ou un répertoire d'un conteneur vers l'hôte :

```
docker cp {{nom_du_conteneur}}:{{chemin/vers/  
le_fichier_ou_le_dossier_de_conteneur}} {{chemin/vers/  
le_fichier_ou_le_dossier_de_l_hote}}
```

- Copier un fichier ou un répertoire de l'hôte vers un conteneur, en suivant les liens symboliques (copie les fichiers liés directement, pas les liens symboliques eux-mêmes) :

```
docker cp --follow-link {{chemin/vers/  
le_lien_symbolique_de_l_hote}} {{nom_du_conteneur}}:{{chemin/  
vers/le_fichier_ou_le_dossier_de_conteneur}}
```

# docker diff

Inspecte les changements apportés aux fichiers ou dossiers sur le système de fichiers d'un conteneur.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/container/diff/>.

- Inspecte les changements apportés à un conteneur depuis sa création :

```
docker diff {{conteneur}}
```

- Affiche l'aide :

```
docker diff --help
```

# docker exec

Exécute une commande dans un conteneur déjà en cours d'exécution.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/container/exec/>.

- Entrer dans un shell interactif dans un conteneur en cours d'exécution :

```
docker exec --interactive --tty {{nom_du_conteneur}} {{/bin/bash}}
```

- Exécuter une commande en arrière-plan (détachée) dans un conteneur en cours d'exécution :

```
docker exec --detach {{nom_du_conteneur}} {{commande}}
```

- Sélectionner le répertoire de travail pour une commande donnée à exécuter :

```
docker exec --interactive --tty --workdir {{chemin/vers/le/répertoire}} {{nom_du_conteneur}} {{commande}}
```

- Exécuter une commande en arrière-plan sur un conteneur existant mais garder stdin ouvert :

```
docker exec --interactive --detach {{nom_du_conteneur}} {{commande}}
```

- Définir une variable d'environnement dans une session Bash en cours d'exécution :

```
docker exec --interactive --tty --env {{variable_d_environnement}}={{valeur}} {{nom_du_conteneur}} {{/bin/bash}}
```

- Exécuter une commande en tant qu'utilisateur spécifique :

```
docker exec --user {{utilisateur}} {{nom_du_conteneur}} {{commande}}
```



# docker image

Gérer les images Docker.

Voir aussi **docker build**, **docker import**, and **docker pull**.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/image/>.

- Lister les images Docker locales :

```
docker image ls
```

- Supprimer les images Docker locales inutilisées :

```
docker image prune
```

- Supprimer toutes les images inutilisées (pas seulement celles sans étiquette) :

```
docker image prune --all
```

- Afficher l'historique d'une image Docker locale :

```
docker image history {{image}}
```

# docker images

Gérer les images Docker.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/image/ls/>.

- Lister toutes les images Docker :

```
docker images
```

- Lister toutes les images Docker, y compris les intermédiaires :

```
docker images --all
```

- Lister les images Docker en mode silencieux (seulement les IDs numériques) :

```
docker images --quiet
```

- Lister toutes les images Docker non utilisées par un conteneur :

```
docker images --filter dangling=true
```

- Lister les images Docker qui contiennent une sous-chaîne dans leur nom :

```
docker images "{{*nom*}}"
```

# docker inspect

Retour d'informations de bas niveau sur les objets Docker.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/inspect/>.

- Montrer l'aide :

```
docker inspect
```

- Afficher les informations de configuration d'un conteneur, image ou volume en utilisant un nom ou un ID :

```
docker inspect {{conteneur|image|ID}}
```

- Afficher l'adresse IP d'un conteneur :

```
docker inspect --format '{{range .NetworkSettings.Networks}}\n{{.IPAddress}}\n{{end}}' {{conteneur}}
```

- Afficher le chemin du fichier journal d'un conteneur :

```
docker inspect --format='{{.LogPath}}' {{conteneur}}
```

- Afficher le nom de l'image d'un conteneur :

```
docker inspect --format='{{.Config.Image}}' {{conteneur}}
```

- Afficher les informations de configuration en JSON :

```
docker inspect --format='{{json .Config}}' {{conteneur}}
```

- Afficher toutes les liaisons de port :

```
docker inspect --format='{{range $p,\n$conf := .NetworkSettings.Ports}} {{.}} -> {{(index\n$conf 0).HostPort}} {{end}}' {{conteneur}}
```

# docker load

Charge des images Docker depuis des fichiers ou **stdin**.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/image/load/>.

- Charge une image Docker depuis stdin :

```
docker load < {{chemin/vers/fichier_image.tar}}
```

- Charge une image Docker depuis un fichier spécifique :

```
docker load --input {{chemin/vers/fichier_image.tar}}
```

- Charge une image Docker depuis un fichier spécifique en mode silencieux :

```
docker load --quiet --input {{chemin/vers/fichier_image.tar}}
```

# docker login

Se connecter à un registre Docker.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/login/>.

- Se connecter de manière interactive à un registre :

```
docker login
```

- Se connecter à un registre avec un nom d'utilisateur spécifique (l'utilisateur sera invité à saisir un mot de passe) :

```
docker login --username {{nom_d_utilisateur}}
```

- Se connecter à un registre avec un nom d'utilisateur et un mot de passe spécifiques :

```
docker login --username {{nom_d_utilisateur}} --password  
{{mot_de_passe}} {{serveur}}
```

- Se connecter à un registre avec un mot de passe depuis l'entrée standard :

```
echo "{{mot_de_passe}}" | docker login --username  
{{nom_d_utilisateur}} --password-stdin
```

# docker logs

Affiche les journaux d'un conteneur.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/container/logs/>.

- Afficher les journaux d'un conteneur :

```
docker logs {{nom_du_conteneur}}
```

- Afficher les journaux d'un conteneur en les suivants :

```
docker logs -f {{nom_du_conteneur}}
```

- Afficher les 5 dernière lignes des journaux d'un conteneur :

```
docker logs {{nom_du_conteneur}} --tail {{5}}
```

- Afficher les journaux d'un conteneur avec l'horodatage :

```
docker logs -t {{nom_du_conteneur}}
```

- Afficher les journaux d'un conteneur depuis un certain temps (i.e. 23m, 10s, 2013-01-02T13:23:37) :

```
docker logs {{nom_du_conteneur}} --until {{temps}}
```

# docker-machine

Créer et gérer des machines qui exécutent Docker.

Plus d'informations : <https://github.com/docker/machine>.

- Lister les machines Docker actuellement en cours d'exécution :

```
docker-machine ls
```

- Créer une nouvelle machine Docker avec un nom spécifique :

```
docker-machine create {{nom}}
```

- Récupérer les informations d'une machine Docker :

```
docker-machine status {{nom}}
```

- Démarrer une machine Docker :

```
docker-machine start {{nom}}
```

- Arrêter une machine Docker :

```
docker-machine stop {{nom}}
```

- Inspecter les informations d'une machine Docker :

```
docker-machine inspect {{nom}}
```

# docker network

Créer et gérer des réseaux Docker.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/network/>.

- Lister tous les réseaux disponible et configuré du service Docker :

```
docker network ls
```

- Créer un réseau défini par l'utilisateur :

```
docker network create --driver {{nom_du_driver}}  
{{nom_du_reseau}}
```

- Afficher les informations détaillées des réseaux séparés par des espaces :

```
docker network inspect {{nom_du_reseau}}
```

- Connecter un conteneur à un réseau en utilisant un nom ou un ID :

```
docker network connect {{nom_du_reseau}} {{nom_du_conteneur|  
ID}}
```

- Déconnecter un conteneur d'un réseau en utilisant un nom ou un ID :

```
docker network disconnect {{nom_du_reseau}}  
{{nom_du_conteneur|ID}}
```

- Supprimer tous les réseaux non utilisés (non reliés à un conteneur) :

```
docker network prune
```

- Supprimer les réseaux séparés par des espaces :

```
docker network rm {{nom_du_reseau}}
```



# docker ps

Lister les conteneurs Docker.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/container/ls/>.

- Lister les conteneurs Docker en cours d'exécution :

```
docker ps
```

- Lister tous les conteneurs Docker (en cours d'exécution et arrêtés) :

```
docker ps --all
```

- Afficher le dernier conteneur Docker créé (avec tous les états) :

```
docker ps --latest
```

- Afficher les conteneurs avec une chaîne de caractère dans leur nom :

```
docker ps --filter "name={{name}}"
```

- Afficher les conteneurs avec une même image comme parent :

```
docker ps --filter "ancestor={{image}}:{{etiquette}}"
```

- Afficher les conteneurs avec un code de sorti spécifique :

```
docker ps --all --filter="exited={{code}}"
```

- Afficher les conteneurs avec un statut spécifique (créé, en cours d'exécution, en cours de suppression, en pause, arrêté, mort) :

```
docker ps --filter "status={{status}}"
```

- Afficher les conteneurs avec un point de montage spécifique :

```
docker ps --filter "volume={{chemin/vers/le/dossier}}" --format "table {{.ID}}\t{{.Image}}\t{{.Names}}\t{{.Mounts}}"
```

# docker pull

Télécharge une image Docker depuis le registre.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/image/pull/>.

- Télécharge une image Docker spécifique :

```
docker pull {{image}}:{{étiquette}}
```

- Télécharge une image Docker spécifique en mode silencieux :

```
docker pull --quiet {{image}}:{{étiquette}}
```

- Télécharge toutes les étiquettes d'une image Docker spécifique :

```
docker pull --all-tags {{image}}
```

- Télécharge un image Docker pour une plateforme spécifique, ex : linux/amd64 :

```
docker pull --platform {{linux/amd64}} {{image}}:{{étiquette}}
```

- Affiche l'aide :

```
docker pull --help
```

# docker rename

Renomme un conteneur.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/container/rename/>.

- Renomme un conteneur :

```
docker rename {{conteneur}} {{nouveau_nom}}
```

- Affiche l'aide :

```
docker rename --help
```

# docker rm

Supprime un ou plusieurs conteneurs.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/container/rm/>.

- Supprimer des conteneurs :

```
docker rm {{conteneur1 conteneur2 ...}}
```

- Supprimer des conteneurs par la force :

```
docker rm --force {{conteneur1 conteneur2 ...}}
```

- Supprimer un conteneur et ses volumes :

```
docker rm --volumes {{conteneur}}
```

- Affiche l'aide :

```
docker rm --help
```

# docker rmi

Supprimer une ou plusieurs images Docker.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/image/rm/>.

- Afficher l'aide :

```
docker rmi
```

- Supprimer une ou plusieurs images en fonction de leurs noms :

```
docker rmi {{image1 image2 ...}}
```

- Supprimer une image en forçant la suppression :

```
docker rmi --force {{image}}
```

- Supprimer une image sans supprimer les parents non étiquetés :

```
docker rmi --no-prune {{image}}
```

# docker run

Exécuter une commande dans un nouveau conteneur Docker.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/container/run/>.

- Exécuter une commande dans un nouveau conteneur Docker avec une image étiquetée :

```
docker run {{image:etiquette}} {{commande}}
```

- Exécuter une commande dans un nouveau conteneur Docker en mode détaché (en arrière-plan) et afficher l'ID du conteneur :

```
docker run --detach {{image}} {{commande}}
```

- Exécuter une commande dans un conteneur effémera avec une mode interactif et un terminal pseudo-TTY :

```
docker run --rm --interactive --tty {{image}} {{commande}}
```

- Exécuter une commande dans un nouveau conteneur avec des variables d'environnement :

```
docker run --env '{{variable}}={{valeur}}' --env {{variable}}  
{{image}} {{commande}}
```

- Exécuter une commande dans un nouveau conteneur avec des volumes montés :

```
docker run --volume {{/chemin/vers/l_hote}}:{{chemin/vers/le/  
conteneur}} {{image}} {{commande}}
```

- Exécuter une commande dans un nouveau conteneur avec des ports publiés :

```
docker run --publish {{port_de_l_hote}}:{{port_du_conteneur}}  
{{image}} {{commande}}
```

- Exécuter une commande dans un nouveau conteneur en écrasant l'entrée du point d'entrée de l'image :

```
docker run --entrypoint {{commande}} {{image}}
```

- Exécuter une commande dans un nouveau conteneur en le connectant à un réseau :

```
docker run --network {{reseau}} {{image}}
```

# docker save

Exporter une ou plusieurs images Docker dans une archive.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/image/save/>.

- Sauvegarder une image en redirigeant la sortie standard vers une archive tar :

```
docker save {{image}}:{{etiquette}} > {{chemin/vers/fichier.tar}}
```

- Sauvegarder une image dans une archive tar :

```
docker save --output {{chemin/vers/fichier.tar}} {{image}}:{{etiquette}}
```

- Sauvegarder toutes les étiquettes de l'image :

```
docker save --output {{chemin/vers/fichier.tar}} {{nom_de_l_image}}
```

- Sélectionner des étiquettes particulières d'une image à sauvegarder :

```
docker save --output {{chemin/vers/fichier.tar}} {{nom_de_l_image:etiquette1 nom_de_l_image:etiquette2 ...}}
```

# docker secret

Gérer les secrets de Docker swarm.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/secret/>.

- Créer un nouveau secret depuis l'entrée standard :

```
{{commande}} | docker secret create {{nom_du_secret}} -
```

- Créer un nouveau secret depuis un fichier :

```
docker secret create {{nom_du_secret}} {{chemin/vers/
fichier}}
```

- Lister tous les secrets :

```
docker secret ls
```

- Afficher des informations détaillées sur un ou plusieurs secrets dans un format humain :

```
docker secret inspect --pretty {{nom_du_secret1
nom_du_secret2 ...}}
```

- Supprimer un ou plusieurs secrets :

```
docker secret rm {{nom_du_secret1 nom_du_secret2 ...}}
```



# docker service

Gérer les services sur un démon Docker.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/service/>.

- Lister les services sur un démon Docker :

```
docker service ls
```

- Créer un nouveau service :

```
docker service create --name {{nom_du_service}} {{image}}:{{etiquette}}
```

- Afficher des informations détaillées d'une liste de services séparée par des espaces :

```
docker service inspect {{nom_du_service|ID}}
```

- Lister les tâches d'une liste de services séparée par des espaces :

```
docker service ps {{nom_du_service|ID}}
```

- Redimensionner à un nombre spécifique de répliques pour une liste de services séparée par des espaces :

```
docker service scale {{nom_du_service}}={{count_of_replicas}}
```

- Supprimer une liste de services séparée par des espaces :

```
docker service rm {{nom_du_service|ID}}
```

# docker-slim

Analyser et optimiser les images Docker.

Plus d'informations : <https://github.com/slimtoolkit/slim>.

- Démarrer DockerSlim en mode interactif :

```
docker-slim
```

- Analyser les couches Docker à partir d'une image spécifique :

```
docker-slim xray --target {{image:etiquette}}
```

- Linter un Dockerfile :

```
docker-slim lint --target {{chemin/vers/Dockerfile}}
```

- Analyser et générer une image Docker optimisée :

```
docker-slim build {{image:etiquette}}
```

- Afficher l'aide pour une sous-commande :

```
docker-slim {{subcommande}} --help
```

# docker start

Lancer un ou plusieurs conteneurs arrêtés.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/container/start/>.

- Afficher l'aide :

```
docker start
```

- Lancer un conteneur Docker :

```
docker start {{conteneur}}
```

- Lancer un conteneur, en attachant `stdout` et `stderr` et en transférant les signaux :

```
docker start --attach {{conteneur}}
```

- Lancer un ou plusieurs conteneurs séparés par des espaces :

```
docker start {{conteneur1 conteneur2 ...}}
```

# docker stats

Affiché un flux en direct des statistiques d'utilisation des ressources pour les conteneurs.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/container/stats/>.

- Afficher un flux en direct des statistiques d'utilisation des ressources pour tous les conteneurs :

```
docker stats
```

- Afficher un flux en direct des statistiques d'utilisation des ressources pour un ou plusieurs conteneurs séparés par des espaces :

```
docker stats {{nom_du_conteneur}}
```

- Change le format de sortie pour afficher l'utilisation CPU du conteneur en pourcentage :

```
docker stats --format "{{.Name}}:\t{{.CPUPerc}}"
```

- Afficher les statistiques d'utilisation des ressources pour tous les conteneurs (y compris ceux qui ne sont pas en cours d'exécution) :

```
docker stats --all
```

- Désactiver le flux en direct des statistiques d'utilisation des ressources et afficher les statistiques une seule fois :

```
docker stats --no-stream
```

# docker swarm

Outil d'orchestration de conteneurs.

Plus d'informations : <https://docs.docker.com/engine/swarm/>.

- Initialiser un cluster swarm :

```
docker swarm init
```

- Afficher le jeton pour rejoindre un cluster swarm en tant que nœud manager ou worker :

```
docker swarm join-token {{worker|manager}}
```

- Rejoindre un nouveau nœud au cluster :

```
docker swarm join --token {{jeton}} {{url_du_manager:2377}}
```

- Supprimer un worker du cluster (à exécuter dans le nœud worker) :

```
docker swarm leave
```

- Afficher le certificat CA actuel au format PEM :

```
docker swarm ca
```

- Changer la certificat CA actuel et afficher le nouveau certificat :

```
docker swarm ca --rotate
```

- Changer la période de validité des certificats des nœuds :

```
docker swarm update --cert-expiry {{heures}}h{{minutes}}m{{secondes}}s
```

# docker system

Gérer les données Docker et afficher des informations sur l'ensemble du système.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/system/>.

- Afficher l'aide :

```
docker system
```

- Afficher l'utilisation du disque par Docker :

```
docker system df
```

- Afficher des informations détaillées sur l'utilisation du disque par Docker :

```
docker system df --verbose
```

- Supprimer les données non utilisées :

```
docker system prune
```

- Supprimer les données non utilisées de plus d'un temps donné dans le passé :

```
docker system prune --filter "until={{heures}}h{{minutes}}m"
```

- Afficher les événements du démon Docker en temps réel :

```
docker system events
```

- Afficher les événements du démon Docker en temps réel avec un format JSON :

```
docker system events --filter 'type=container' --format '{{json .}}'
```

- Afficher les informations sur le système Docker :

```
docker system info
```

# docker volume

Gérer les volumes de Docker.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/volume/>.

- Créer un volume :

```
docker volume create {{nom_du_volume}}
```

- Créer un volume avec une étiquette spécifique :

```
docker volume create --label {{étiquette}} {{nom_du_volume}}
```

- Créer un volume tmpfs avec une taille de 100 Mo et un uid de 1000 :

```
docker volume create --opt {{type}}={{tmpfs}} --opt  
{{device}}={{tmpfs}} --opt {{o}}={{size=100m,uid=1000}}  
{{nom_du_volume}}
```

- Lister tous les volumes :

```
docker volume ls
```

- Supprimer un volume :

```
docker volume rm {{nom_du_volume}}
```

- Afficher des informations sur un volume :

```
docker volume inspect {{nom_du_volume}}
```

- Supprimer tous les volumes locaux non utilisés :

```
docker volume prune
```

- Afficher l'aide pour une sous-commande :

```
docker volume {{sous_commande}} --help
```

# docker

Gestion des conteneurs et des images Docker.

Certaines commandes comme **docker run** ont leur propre documentation.

Plus d'informations : <https://docs.docker.com/reference/cli/docker/>.

- Liste tous les conteneurs Docker (en cours d'exécution ou arrêtés) :

```
docker ps --all
```

- Démarre un conteneur à partir d'une image, avec un nom personnalisé :

```
docker run --name {{nom_conteneur}} {{image}}
```

- Démarre ou arrête un conteneur existant :

```
docker {{start|stop}} {{nom_conteneur}}
```

- Télécharge une image depuis un registre Docker :

```
docker pull {{image}}
```

- Affiche les images déjà téléchargées :

```
docker images
```

- Ouvre un shell dans un conteneur déjà en cours d'exécution :

```
docker exec -it {{nom_conteneur}} {{sh}}
```

- Supprime un conteneur arrêté :

```
docker rm {{nom_conteneur}}
```

- Récupère et suit les journaux de message d'un conteneur :

```
docker logs -f {{nom_conteneur}}
```



# drill

Effectue diverses requêtes DNS.

Plus d'informations : <https://manned.org/drill>.

- Recherche des adresses IP(s) associées à un nom d'hôte (enregistrements A) :

```
drill {{example.com}}
```

- Recherche le(s) serveur(s) de messagerie associé(s) à un nom de domaine donné (enregistrement MX) :

```
drill mx {{example.com}}
```

- Obtient tous les types d'enregistrements pour un nom de domaine donné :

```
drill any {{example.com}}
```

- Spécifie un autre serveur DNS à interroger :

```
drill {{example.com}} @{{8.8.8.8}}
```

- Effectue une recherche DNS inversée sur une adresse IP (enregistrement PTR) :

```
drill -x {{8.8.8.8}}
```

- Suit la traçabilité DNSSEC des serveurs racines jusqu'à un nom de domaine :

```
drill -TD {{example.com}}
```

- Affiche le(s) enregistrement(s) DNSKEY d'un nom de domaine :

```
drill -s dnskey {{example.com}}
```

# echo

Affiche les paramètres donnés dans la console.

Plus d'informations : <https://www.gnu.org/software/coreutils/echo>.

- Affiche un message (les guillemets sont facultatifs) :

```
echo "{{Hello World}}"
```

- Affiche un message avec des variables d'environnement :

```
echo "{{Ma variable PATH est $PATH}}"
```

- Affiche un message sans retour à la ligne :

```
echo -n "{{Hello World}}"
```

- Ajoute un message à un fichier :

```
echo "{{Hello World}}" >> {{fichier.txt}}
```

- Active l'interprétation des spécificateurs d'échappement :

```
echo -e "{{Colonne 1\tColonne 2}}"
```

# egrep

Recherche de motifs dans un texte. Supporte la version étendue des expressions régulières (? , + , {}, ( ), et |).

Plus d'informations : <https://manned.org/egrep>.

- Recherche une chaîne de caractères précise :

```
egrep "{{chaîne_recherchée}}" {{chemin/vers/fichier}}
```

- Recherche une chaîne de caractères dans plusieurs fichiers :

```
egrep "{{chaîne_recherchée}}" {{chemin/vers/fichier1 chemin/vers/fichier2 ...}}
```

- Utilise l'entrée standard au lieu d'un fichier :

```
cat {{chemin/vers/fichier}} | egrep {{chaîne_recherchée}}
```

- Affiche le nom du fichier avec la ligne correspondante pour chaque concordance :

```
egrep --with-filename --line-number  
"{{chaîne_recherchée}}" {{chemin/vers/fichier}}
```

- Recherche récursivement dans le dossier courant, en ignorant les fichiers binaires, une chaîne de caractères précise :

```
egrep --recursive --binary-files={{without-match}}  
"{{chaîne_recherchée}}" {{chemin/vers/fichier}}
```

- Inverse le résultat pour exclure des chaînes de caractères spécifiques :

```
egrep --invert-match "{{chaîne_recherchée}}" {{chemin/vers/fichier}}
```

# env

Affiche l'environnement ou exécute un programme dans un environnement modifié.

Plus d'informations : <https://www.gnu.org/software/coreutils/env>.

- Affiche l'environnement :

```
env
```

- Exécute le programme donné. Souvent utilisé dans les scripts après le shebang (#!) pour consulter le chemin vers le programme :

```
env {{programme}}
```

- Exécute le programme donné, avec un environnement vide :

```
env -i {{programme}}
```

- Supprime une variable d'environnement et exécute le programme donné :

```
env -u {{variable}} {{programme}}
```

- Définit ou modifie une variable d'environnement et exécute le programme donné :

```
env {{variable}}={{valeur}} {{programme}}
```

- Définit ou modifie plusieurs variables d'environnement et exécute le programme donné :

```
env {{variable1}}={{valeur}} {{variable2}}={{valeur}}  
{{variable3}}={{valeur}} {{programme}}
```

# errno

Recherche les noms et descriptions des codes errno.

Plus d'informations : <https://joeyh.name/code/moreutils/>.

- Recherche la description d'une erreur par nom ou par code :

```
errno {{nom|code}}
```

- Liste tous les noms, codes et descriptions d'errno :

```
errno --list
```

- Cherche les codes dont la description contient le texte entier indiqué :

```
errno --search {{texte}}
```

- Cherche les codes dont la description contient le texte entier indiqué (dans toutes les langues) :

```
errno --search-all-locales {{texte}}
```

# exa

Une alternative moderne à `ls` (pour lister le contenu de répertoires).

Plus d'informations : <https://the.exa.website>.

- Liste les fichiers, un par ligne :

```
exa --oneline
```

- Liste tous les fichiers, y compris les fichiers cachés :

```
exa --all
```

- Liste au format long (autorisations, propriété, taille et date de modification) de tous les fichiers :

```
exa --long --all
```

- Liste les fichiers avec le plus volumineux en haut :

```
exa --reverse --sort={{taille}}
```

- Affiche une arborescence de fichiers, sur trois niveaux de profondeur :

```
exa --long --tree --level={{3}}
```

- Liste des fichiers triés par date de modification (le plus ancien en premier) :

```
exa --long --sort={{modifié}}
```

- Liste les fichiers avec leur en-tête, leur icône et leur statut Git :

```
exa --long --header --icons --git
```

- Liste les fichiers sauf ceux mentionnés dans `.gitignore` :

```
exa --git-ignore
```

# fdupes

Trouve les fichiers dupliqués dans les dossiers donnés.

Plus d'informations : <https://github.com/adrianlopezroche/fdupes>.

- Chercher dans un dossier :

```
fdupes {{dossier}}
```

- Chercher dans plusieurs dossiers :

```
fdupes {{dossier1}} {{dossier2}}
```

- Chercher dans un dossier récursivement :

```
fdupes -r {{dossier}}
```

- Chercher dans plusieurs dossiers dont un récursivement :

```
fdupes {{dossier1}} -R {{dossier2}}
```

- Chercher récursivement les dupliqués et demander les fichiers à conserver, supprimant les autres :

```
fdupes -rd {{dossier}}
```

- Chercher récursivement et supprimer les dupliqués automatiquement :

```
fdupes -rdN {{dossier}}
```

# ffuf

Un outil d'énumération web rapide écrit en Go.

Le mot-clé **FUZZ** est utilisé comme substitut. **ffuf** essaiera d'atteindre l'URL en remplaçant le mot **FUZZ** par tous les mots de la liste donnée.

Plus d'informations : <https://github.com/ffuf/ffuf#usage>.

- Énumère les répertoires en utilisant une sortie [c]olorée et une liste ([w]ordlist) spécifiant une [u]RL cible :

```
ffuf -c -w {{chemin/vers/liste.txt}} -u {{http://cible/FUZZ}}
```

- Énumère les sous-domaines en changeant la position du mot-clé :

```
ffuf -w {{chemin/vers/sous-domaines.txt}} -u {{http://FUZZ.cible.com}}
```

- Énumère avec le nombre de fils d'exécution ([t]hreads) spécifié (par défaut : 40) et passe le trafic par un serveur mandataire (pro[x]y) et sauvegarde la sortie ([o]utput) dans un fichier :

```
ffuf -o -w {{chemin/vers/liste.txt}} -u {{http://cible/FUZZ}}  
-t {{500}} -x {{http://127.0.0.1:8080}}
```

- Énumère dans un en-tête ([H]eader) spécifique ("Nom : Valeur") et n'affiche que les [c]odes d'état HTTP correspondant ([m]atch) :

```
ffuf -w {{chemin/vers/liste.txt}} -u {{http://cible.com}} -H  
"{{Host: FUZZ}}" -mc {{200}}
```

- Énumère en spécifiant une méthode HTTP spécifiée et des [d]onnées, tout en [f]iltrant les [c]odes d'état séparés par des virgules :

```
ffuf -w {{chemin/vers/postdata.txt}} -X {{POST}} -d  
"{{username=admin&password=FUZZ}}" -u {{http://cible/  
login.php}} -fc {{401,403}}
```



# find

Trouve récursivement des fichiers ou des dossiers dans l'arborescence spécifiée.

Plus d'informations : <https://manned.org/find>.

- Trouve des fichiers par extension :

```
find {{racine}} -name '{{*.ext}}'
```

- Trouve des fichiers correspondant à plusieurs chemins ou motifs :

```
find {{racine}} -path '{{**/chemin/**/*.*}}' -or -name  
'{{*motif*}}'
```

- Trouve des dossiers correspondant à un nom donné sans vérifier la casse :

```
find {{racine}} -type d -iname '{{*lib*}}'
```

- Trouve des fichiers correspondant à un motif donné en excluant certains chemins de la recherche :

```
find {{racine}} -name '{{*.py}}' -not -path '{{*/site-  
packages/*}}'
```

- Trouve des fichiers dans une fourchette de tailles et limite la profondeur récursive à "1" :

```
find {{racine}} -maxdepth 1 -size {{+500k}} -size {{-10M}}
```

- Exécute une commande pour chaque fichier (utiliser {} dans la commande pour utiliser le nom des fichiers) :

```
find {{racine}} -name '{{*.ext}}' -exec {{wc -l {} }}\;
```

- Trouve les fichiers modifiés dans les 7 derniers jours :

```
find {{racine}} -daystart -mtime -{{7}}
```

- Trouve les fichiers vides (de taille nulle) et les supprimer :

```
find {{racine}} -type {{f}} -empty -delete
```

# fish

Friendly Interactive SHell, un interpréteur de ligne de commande, conçu pour être facile à utiliser.

Plus d'informations : <https://fishshell.com>.

- Démarre une session shell interactive :

```
fish
```

- Exécute une commande, puis termine la session :

```
fish -c "{{commande}}"
```

- Exécute un script :

```
fish {{chemin/vers/le/script.fish}}
```

- Vérifie les erreurs de syntaxe dans un script :

```
fish --no-execute {{chemin/vers/le/script.fish}}
```

- Démarre une session shell interactive en mode privé, dans laquelle le shell n'a pas accès à l'historique et n'y écrit rien :

```
fish --private
```

- Affiche les informations de version :

```
fish --version
```

- Ajoute et exporte une variable d'environnement, qui persiste entre les redémarrages du shell (à exécuter depuis le shell uniquement) :

```
set -Ux {{nom_de_la_variable}} {{valeur_de_la_variable}}
```

# fossil ci

Cette commande est un alias de **fossil commit**.

Plus d'informations : <https://fossil-scm.org/home/help/commit>.

- Voir la documentation de la commande originale :

```
tldr fossil-commit
```

# fossil delete

Cette commande est un alias de **fossil rm**.

Plus d'informations : <https://fossil-scm.org/home/help/delete>.

- Voir la documentation de la commande originale :

```
tldr fossil rm
```

# fossil forget

Cette commande est un alias de **fossil rm**.

Plus d'informations : <https://fossil-scm.org/home/help/forget>.

- Voir la documentation de la commande originale :

```
tldr fossil rm
```

# fossil new

Cette commande est un alias de **fossil init**.

Plus d'informations : <https://fossil-scm.org/home/help/new>.

- Voir la documentation de la commande originale :

```
tldr fossil-init
```

# ftp

Outils permettant d'interagir avec un serveur avec le protocole FTP.

Plus d'informations : <https://manned.org/ftp>.

- Connexion à un serveur FTP :

```
ftp {{ftp.exemple.com}}
```

- Connexion à un serveur FTP en spécifiant son adresse IP et son port :

```
ftp {{adresse_IP}} {{port}}
```

- Passe en mode de transfert binaire (médias, fichiers compressés, etc) :

```
binary
```

- Transfère plusieurs fichiers sans demander de confirmation pour chaque :

```
prompt off
```

- Télécharge plusieurs fichiers :

```
mget {{*.png}}
```

- Téléverse plusieurs fichiers :

```
mput {{*.zip}}
```

- Supprime plusieurs fichiers sur le serveur distant :

```
mdelete {{*.txt}}
```

- Renomme un fichier sur le serveur distant :

```
rename {{ancien_fichier}} {{nouveau_fichier}}
```

# fzf

Recherche approximative en ligne de commande.

Similaire à **sk**.

Plus d'informations : <https://github.com/junegunn/fzf>.

- Lance `fzf` sur tous les fichiers du répertoire spécifié :

```
find {{chemin/vers/répertoire}} -type f | fzf
```

- Démarre `fzf` pour les processus en cours :

```
ps aux | fzf
```

- Sélectionne plusieurs fichiers avec `Shift + Tab` et écrit dans un fichier :

```
find {{chemin/vers/répertoire}} -type f | fzf --multi > {{chemin/vers/fichier}}
```

- Lance `fzf` avec une requête donnée :

```
fzf --query "{{query}}"
```

- Lance `fzf` sur les entrées qui commencent par `core` et se terminent par `go`, `rb`, ou `py` :

```
fzf --query "^core go$ | rb$ | py$"
```

- Lance `fzf` sur les entrées qui ne correspondent pas à `pyc` et qui correspondent exactement à `travis` :

```
fzf --query "!pyc 'travis'"
```



# gcc

Préprocesse et compile des fichiers sources C et C++, pour ensuite les assembler et les lier entre eux.

Plus d'informations : <https://gcc.gnu.org>.

- Compile plusieurs fichiers source en un exécutable :

```
gcc {{chemin/vers/le/fichier1.c chemin/vers/le/
fichier2.c ...}} -o {{chemin/vers/l'exécutable/à/produire}}
```

- Active les avertissements et les symboles de débogage dans la sortie de commande :

```
gcc {{chemin/vers/le/fichiersource.c}} -Wall -Og -o {{chemin/
vers/l'exécutable/à/produire}}
```

- Inclut les bibliothèques depuis un chemin différent de celui par défaut :

```
gcc {{chemin/vers/le/fichiersource.c}} -o {{chemin/vers/
l'exécutable/à/produire}} -I{{chemin/vers/les/fichiers/d'en-
tête}} -L{{chemin/vers/la/bibliothèque}} -l{{bibliothèque}}
```

- Compile le code source en des instructions d'assemblage :

```
gcc -S {{chemin/vers/le/fichiersource.c}}
```

- Compile le code source en un fichier objet sans le lier :

```
gcc -c {{chemin/vers/le/fichiersource.c}}
```

# `gdb`

Le débogueur GNU.

Plus d'informations : <https://www.gnu.org/software/gdb>.

- Débogue un exécutable :

```
gdb {{exécutable}}
```

- Attache un processus à gdb :

```
gdb -p {{identifiant_processus}}
```

- Débogue avec un fichier comme image mémoire :

```
gdb -c {{fichier}} {{exécutable}}
```

- Exécute les commandes gdb données au démarrage :

```
gdb -ex "{{commandes}}" {{exécutable}}
```

- Démarre gdb en passant des arguments à l'exécutable :

```
gdb --args {{exécutable}} {{argument1}} {{argument2}}
```

# gem

Un gestionnaire de paquets pour le langage de programmation Ruby.

Plus d'informations : <https://guides.rubygems.org>.

- Recherche des gems distantes et affiche toutes les versions disponibles :

```
gem search {{expression_régulière}} --all
```

- Installe la dernière version d'une gem :

```
gem install {{nom_gem}}
```

- Installe une version spécifique d'une gem :

```
gem install {{nom_gem}} --version {{1.0.0}}
```

- Installe la dernière version correspondante (SemVer) d'une gem :

```
gem install {{nom_gem}} --version '~> {{1.0}}'
```

- Mise à jour d'une gem :

```
gem update {{nom_gem}}
```

- Liste toutes les gems locales :

```
gem list
```

- Désinstalle une gem :

```
gem uninstall {{nom_gem}}
```

- Désinstalle une version spécifique d'une gem :

```
gem uninstall {{nom_gem}} --version {{1.0.0}}
```

# gh alias

Gérer les alias de commandes GitHub CLI depuis la ligne de commande.

Plus d'informations : [https://cli.github.com/manual/gh\\_alias](https://cli.github.com/manual/gh_alias).

- Liste tous les alias pour lesquels gh est configuré :

```
gh alias list
```

- Crée un alias de sous-commande pour gh :

```
gh alias set {{alias}} '{{commande_gh}}'
```

- Définit une commande shell comme sous-commande de gh :

```
gh alias set --shell {{nom_de_l'alias}} {{commande}}
```

- Supprime un alias :

```
gh alias delete {{nom_de_l'alias}}
```

- Affiche l'aide pour la sous-commande alias :

```
gh alias
```

# gh cs

Cette commande est un alias de **gh codespace**.

Plus d'informations : [https://cli.github.com/manual/gh\\_codespace](https://cli.github.com/manual/gh_codespace).

- Voir la documentation de la commande originale :

```
tldr gh-codespace
```

# gh

Travailler harmonieusement avec GitHub depuis la ligne de commande.

Certaines commandes comme **gh config** ont leur propre documentation.

Plus d'informations : <https://cli.github.com/>.

- Clone un dépôt GitHub localement :

```
gh repo clone {{utilisateur}}/{{dépôt}}
```

- Crée une nouvelle issue :

```
gh issue create
```

- Affiche et filtre les issues ouvertes du dépôt courant :

```
gh issue list
```

- Affiche une issue dans le navigateur Web par défaut :

```
gh issue view --web {{numéro_de_l'issue}}
```

- Crée une pull request :

```
gh pr create
```

- Affiche une pull request dans le navigateur Web par défaut :

```
gh pr view --web {{numéro_de_la_PR}}
```

- Observe une pull request spécifique localement :

```
gh pr checkout {{numéro_de_la_PR}}
```

- Affiche le statut des pull requests du dépôt courant :

```
gh pr status
```

# ghdl

Simulateur à source ouvert pour le langage VHDL.

Plus d'informations : <https://ghdl.github.io/ghdl/>.

- Analyse un fichier de source VHDL et génère un fichier objet :

```
ghdl -a {{fichier.vhdl}}
```

- Élabore un design (où `design` est le nom d'une unité de configuration, d'entité, ou d'architecture) :

```
ghdl -e {{design}}
```

- Exécute un design élaboré :

```
ghdl -r {{design}}
```

- Exécute un design élaboré et sauvegarde la sortie à un fichier de forme d'onde :

```
ghdl -r {{design}} --wave={{sortie.ghw}}
```

- Vérifie le syntaxe d'un fichier de source VHDL :

```
ghdl -s {{fichier.vhdl}}
```

- Affiche l'aide générale :

```
ghdl --help
```

# gimp

Outil d'édition et de retouche d'image, libre et multiplateforme.

Voir aussi : **krita**.

Plus d'informations : <https://docs.gimp.org/en/gimp-fire-up.html#gimp-concepts-running-command-line>.

- Démarre GIMP :

```
gimp
```

- Ouvre les fichiers spécifiés :

```
gimp {{chemin/vers/image1 chemin/vers/image2 ...}}
```

- Ouvre les fichiers spécifiés avec une nouvelle Démarre une nouvelle cadre :

```
gimp --new-instance {{chemin/vers/image1 chemin/vers/  
image2 ...}}
```

- Démarre sans l'écran de démarrage :

```
gimp --no-splash
```

- Affiche les erreurs et les avertissements sur la console au lieu de les afficher dans une boîte de dialogue :

```
gimp --console-messages
```

- Active les routines de débogage des signaux non fatals :

```
gimp --debug-handlers
```



# git add

Ajoute les fichiers modifiés à l'index.

Plus d'informations : <https://git-scm.com/docs/git-add>.

- Ajouter un fichier à l'index :

```
git add {{chemin/vers/fichier}}
```

- Ajouter tous les fichiers (suivis et non-suivis) :

```
git add -A
```

- Ajoute les modifications des fichiers déjà suivis :

```
git add -u
```

- Ajoute aussi les fichiers ignorés :

```
git add -f
```

- Ajoute des parties de fichiers interactivement :

```
git add -p
```

- Ajoute interactivement les parties d'un fichier spécifié :

```
git add -p {{chemin/vers/fichier}}
```

- Ajouter un fichier interactivement :

```
git add -i
```

# git am

Appliquer des fichiers de "patch" Git. Utile lorsque l'on reçoit des commits par email.

Voir aussi **git format-patch**, pour générer des fichiers de patch.

Plus d'informations : <https://git-scm.com/docs/git-am>.

- Appliquer un fichier de patch :

```
git am {{chemin/vers/fichier.patch}}
```

- Annuler l'application d'un fichier de patch :

```
git am --abort
```

- Appliquer autant de fichiers de correctif que possible, en enregistrant les morceaux échoués pour rejeter le fichier :

```
git am --reject {{chemin/vers/fichier.patch}}
```

# git annex

Gérez les fichiers avec Git, sans archiver leur contenu.

Lorsqu'un fichier est annexé, son contenu est déplacé dans un stockage clé-valeur et un lien symbolique est créé qui pointe vers le contenu.

Plus d'informations : <https://git-annex.branchable.com>.

- Initialise le dépôt :

```
git annex init
```

- Ajoute un fichier :

```
git annex add {{chemin/vers/fichier_ou_repertoire}}
```

- Affiche le statut courant d'un fichier ou répertoire :

```
git annex status {{chemin/vers/fichier_ou_repertoire}}
```

- Synchronise un dépôt local avec un distant :

```
git annex {{distant}}
```

- Récupère un fichier ou un répertoire :

```
git annex get {{chemin/vers/fichier_ou_repertoire}}
```

- Affiche l'aide :

```
git annex help
```

# git apply

Applique un correctif à un fichier et/ou à l'index.

Plus d'informations : <https://git-scm.com/docs/git-apply>.

- Afficher les messages à propos des fichiers corrigés :

```
git apply --verbose {{chemin/vers/fichier}}
```

- Applique le correctif et ajoute les fichiers à l'index :

```
git apply --index {{chemin/vers/fichier}}
```

- Applique un correctif depuis une source distante :

```
curl -L {{https://example.com/file.patch}} | git apply
```

- Affiche les différences résultantes et applique le correctif :

```
git apply --stat --apply {{chemin/vers/fichier}}
```

- Applique le correctif en ordre inverse :

```
git apply --reverse {{chemin/vers/fichier}}
```

- Stocke le résultat du correctif dans l'index sans modifier la branche courante :

```
git apply --cache {{chemin/vers/fichier}}
```

# git archive

Crée une archive de fichiers depuis un branche donnée.

Plus d'informations : <https://git-scm.com/docs/git-archive>.

- Crée une archive `.tar` avec le contenu de la HEAD et l'affiche sur la sortie standard :

```
git archive {{-v|--verbose}} HEAD
```

- Crée une archive Zip avec le contenu de la HEAD et l'affiche sur la sortie standard :

```
git archive {{-v|--verbose}} --format zip HEAD
```

- Pareil que ci-dessus mais écrit dans l'archive spécifiée :

```
git archive {{-v|--verbose}} {{-o|--output}} {{chemin/vers/fichier.zip}} HEAD
```

- Crée une archive depuis le dernier commit de la branche spécifiée :

```
git archive {{-o|--output}} {{chemin/vers/fichier.tar}}  
{{nom_de_branche}}
```

- Crée une archive avec le contenu d'un répertoire donné :

```
git archive {{-o|--output}} {{chemin/vers/fichier.tar}} HEAD:  
{{chemin/vers/repertoire}}
```

- Ajoutez un chemin d'accès à chaque fichier pour l'archiver dans un répertoire spécifique :

```
git archive {{-o|--output}} {{chemin/vers/fichier.tar}} --  
prefix {{chemin/vers/cible}}/ HEAD
```

# git bisect

Utiliser une recherche binaire pour trouver le commit qui a introduit un bug.

Git saute automatiquement d'avant en arrière dans le graphe de commit pour isoler le commit défectueux.

Plus d'informations : <https://git-scm.com/docs/git-bisect>.

- Démarrez une dissection sur une plage de commit délimitée par un bug connu et un commit propre connu (généralement plus ancien) :

```
git bisect start {{bad_commit}} {{good_commit}}
```

- Pour chaque `git bisect` sélectionné, le marquer comme "mauvais" (`bad`) ou "bon" (`good`) après l'avoir testé pour le problème :

```
git bisect {{good|bad}}
```

- Après que `git bisect` pointe vers le mauvais commit, terminer la dissection et retourner à la branche précédente :

```
git bisect reset
```

- Sauter un commit lors de la dissection (e.g. celui qui échoue les tests pour une autre raison) :

```
git bisect skip
```

# git blame

Affiche le hash de commit et le dernier auteur de chaque ligne d'un fichier.

Plus d'informations : <https://git-scm.com/docs/git-blame>.

- Affiche le hash de commit et le nom de l'auteur en face de chaque ligne :

```
git blame {{file}}
```

- Affiche le hash de commit le nom et l'email de l'auteur en face de chaque ligne :

```
git blame {{-e|--show-email}} {{file}}
```

# git branch

Commande Git principale pour travailler avec des branches.

Plus d'informations : <https://git-scm.com/docs/git-branch>.

- Liste toutes les branches (locale et distantes) :

```
git branch --all
```

- Affiche le nom de la branche courante :

```
git branch --show-current
```

- Crée une nouvelle branche depuis le commit courant :

```
git branch {{nom_de_branche}}
```

- Crée une nouvelle branche depuis un commit en particulier :

```
git branch {{nom_de_branche}} {{commit_hash}}
```

- Renommer une branche (ne pas se trouver sur la branche pour le faire) :

```
git branch {{-m|--move}} {{ancien_nom_de_branche}}  
{{nouveau_nom_de_branche}}
```

- Supprimer une branche locale (ne pas se trouver sur la branche pour le faire) :

```
git branch {{-d|--delete}} {{nom_de_branche}}
```

- Supprimer une branche distante :

```
git push {{nom_distant}} --delete {{nom_de_branche_distante}}
```



# git bundle

Empaquetez des objets et des références dans une archive.

Plus d'informations : <https://git-scm.com/docs/git-bundle>.

- Empaquetez tout les objets et les références d'une branche spécifiée :

```
git bundle create {{chemin/vers/fichier.bundle}}  
{{nom_de_branche}}
```

- Crée un empaquetage de tout les fichiers de toutes les branches :

```
git bundle create {{chemin/vers/fichier.bundle}} --all
```

- Crée un empaquetage des 5 derniers commits de la branche courante :

```
git bundle create {{chemin/vers/fichier.bundle}} -5 {{HEAD}}
```

- Crée un empaquetage des 7 derniers jours :

```
git bundle create {{chemin/vers/fichier.bundle}} --since  
7.days {{HEAD}}
```

- Vérifie qu'un empaquetage est valide et peut être appliqué à la branche courante :

```
git bundle verify {{chemin/vers/fichier.bundle}}
```

- Affiche sur la sortie standard la liste des références contenues dans un empaquetage :

```
git bundle unbundle {{chemin/vers/fichier.bundle}}
```

- Extraire une branche spécifique d'un fichier de bundle dans le référentiel actuel :

```
git pull {{chemin/vers/fichier.bundle}} {{nom_de_branche}}
```

- Créer un nouveau dépôt depuis un empaquetage :

```
git clone {{chemin/vers/fichier.bundle}}
```

# git cat-file

Fournir des informations sur le contenu ou le type et la taille des objets du dépôt Git.

Plus d'informations : <https://git-scm.com/docs/git-cat-file>.

- Obtenir la taille [s] du commit HEAD en octets :

```
git cat-file -s HEAD
```

- Obtenir le type [t] (blob, tree, commit, tag) d'un objet Git spécifié :

```
git cat-file -t {{8c442dc3}}
```

- Affiche le contenu [p] d'un objet Git basé sur son type :

```
git cat-file -p {{HEAD~2}}
```

# git check-ignore

Analyser et déboguer les fichiers ignorés / exclus (".gitignore") de Git.

Plus d'informations : <https://git-scm.com/docs/git-check-ignore>.

- Vérifie qu'un fichier ou répertoire est ignoré :

```
git check-ignore {{chemin/vers/fichier_ou_répertoire}}
```

- Vérifie que plusieurs fichiers ou répertoires sont ignorés :

```
git check-ignore {{chemin/vers/fichier_ou_répertoire1 chemin/vers/fichier_ou_répertoire2 ...}}
```

- Utilisez des chemins d'accès, un par ligne, de stdin :

```
git check-ignore --stdin < {{chemin/vers/fichier_annexe}}
```

- Ne pas vérifier l'index (utilisé pour déboguer pourquoi les chemins ont été suivis et non ignorés) :

```
git check-ignore --no-index {{chemin/vers/fichier_ou_répertoire1 chemin/vers/fichier_ou_répertoire2 ...}}
```

- Inclure les détails pour chaque occurrence dans le chemin :

```
git check-ignore --verbose {{chemin/vers/fichier_ou_répertoire1 chemin/vers/fichier_ou_répertoire2 ...}}
```

# git checkout

Extraire une branche ou des chemins vers l'arborescence de travail.

Plus d'informations : <https://git-scm.com/docs/git-checkout>.

- Créer une branche et basculer dessus :

```
git checkout -b {{nom_de_branche}}
```

- Créer une branche depuis une référence spécifique et basculer dessus (par exemple, branche locale/distante, tag, commit) :

```
git checkout -b {{nom_de_branche}} {{référence}}
```

- Basculer sur une branche locale existante :

```
git checkout {{nom_de_branche}}
```

- Basculer sur la branche précédente :

```
git checkout -
```

- Basculer sur une branche distante existante :

```
git checkout --track {{nom_distant}}/{{nom_de_branche}}
```

- Annule tout les changements dans le répertoire courant (voir `git reset` pour plus de commandes d'annulation) :

```
git checkout .
```

- Annule tout les changements dans le fichier spécifié :

```
git checkout {{filename}}
```

- Remplace un fichier par sa version d'une autre branche :

```
git checkout {{nom_de_branche}} -- {{filename}}
```

# git cherry-pick

Appliquer les modifications introduites par les commits existants à la branche actuelle.

Pour appliquer les changements a une autre branche, utiliser d'abord **git checkout** pour basculer sur la branche désirée.

Plus d'informations : <https://git-scm.com/docs/git-cherry-pick>.

- Applique un commit à la branche courante :

```
git cherry-pick {{commit}}
```

- Appliquer une plage de commits à la branche courante (voir aussi **git rebase --onto**):

```
git cherry-pick {{start_commit}}~..{{end_commit}}
```

- Appliquer plusieurs commits non séquentiels à la branche courante :

```
git cherry-pick {{commit_1 commit_2 ...}}
```

- Appliquer les changements d'un commit à la branche courante sans créer de commit :

```
git cherry-pick --no-commit {{commit}}
```

# git cherry

Rechercher des commits qui n'ont pas encore été appliqués en amont.

Plus d'informations : <https://git-scm.com/docs/git-cherry>.

- Afficher les commits (et leurs messages) avec des commits équivalents en amont :

```
git cherry {{-v|--verbose}}
```

- Spécifiez une branche amont et une branche de rubrique différentes :

```
git cherry {{origin}} {{topic}}
```

- Limiter les commits à ceux dans la limite donnée :

```
git cherry {{origin}} {{topic}} {{base}}
```

# git clean

Supprimer les fichiers non-suivis d'un dépôt Git.

Plus d'informations : <https://git-scm.com/docs/git-clean>.

- Supprimer les fichiers non-suivis :

```
git clean
```

- Supprimer les fichiers non-suivis de manière interactive :

```
git clean {{-i|--interactive}}
```

- Affiche les fichiers non-suivis qui peuvent être supprimés :

```
git clean --dry-run
```

- Nettoyage forcé des fichiers non-suivis :

```
git clean {{-f|--force}}
```

- Nettoyage forcé des répertoires non-suivis :

```
git clean {{-f|--force}} -d
```

- Supprime tous les fichiers suivis, incluant ceux répertoriés par `.gitignore` et `.git/info/exclude` :

```
git clean -x
```

# git clone

Clone un dépôt existant.

Plus d'informations : <https://git-scm.com/docs/git-clone>.

- Clone un dépôt existant dans un répertoire spécifique :

```
git clone {{emplacement_du_depot_distant}} {{chemin/vers/repertoire}}
```

- Clone un dépôt existant et ses sous-modules :

```
git clone --recursive {{emplacement_du_depot_distant}}
```

- Clone un dépôt local :

```
git clone --local {{chemin/vers/depot/local}}
```

- Clone silencieusement :

```
git clone --quiet {{emplacement_du_depot_distant}}
```

- Clone un dépôt existant en ne récupérant que les 10 commits les plus récents sur la branche par défaut (plus rapide) :

```
git clone --depth {{10}} {{emplacement_du_depot_distant}}
```

- Clone un dépôt existant en ne récupérant qu'une branche spécifique :

```
git clone --branch {{nom}} --single-branch  
{{emplacement_du_depot_distant}}
```

- Clone un dépôt existant en utilisant une commande SSH spécifique :

```
git clone --config core.sshCommand="{{ssh -i chemin/vers/  
clef_ssh_privee}}" {{emplacement_du_depot_distant}}
```



# git commit

Enregistrer (**commit**) les fichiers dans le dépôt.

Plus d'informations : <https://git-scm.com/docs/git-commit>.

- Commit les fichiers en stage dans le dépôt avec un message :

```
git commit -m "{{message}}"
```

- Commit tous les fichiers modifiés avec un message :

```
git commit -am "{{message}}"
```

- Met à jour le dernier commit avec les modifications en stage :

```
git commit --amend
```

- Commit seulement les fichiers spécifiés (qui sont déjà en stage) :

```
git commit {{chemin/vers/mon/fichier1 chemin/vers/mon/
fichier2 ...}}
```

# git config

Gérer les options de configuration personnalisées pour les référentiels Git.

Ces configurations peuvent être locales (pour le référentiel courant) ou globales (pour l'utilisateur).

Plus d'informations : <https://git-scm.com/docs/git-config>.

- Liste les entrées de configurations locales (stockées dans `.git/config` du répertoire courant) :

```
git config --list --local
```

- Liste les entrées de configuration globales (stockées dans `~/.gitconfig`) :

```
git config --list --global
```

- Récupère la valeur d'une entrée de configuration :

```
git config alias.unstage
```

- Attribue la valeur d'une entrée de configuration :

```
git config --global alias.unstage "reset HEAD --"
```

- Restaure la valeur d'une entrée de configuration globale à sa valeur par défaut :

```
git config --global --unset alias.unstage
```

- Édite le fichier de configuration du référentiel courant dans l'éditeur par défaut :

```
git config --edit
```

- Édite le fichier de configuration globale dans l'éditeur par défaut :

```
git config --global --edit
```

# git describe

Créer un nom unique et lisible pour un objet à partir d'une référence disponible.

Plus d'informations : <https://git-scm.com/docs/git-describe>.

- Créer un nom unique pour le commit courant (le nom contient le tag le plus récent, le nombre de commits additionnels, et l'empreinte abrégée du commit) :

```
git describe
```

- Créer un nom avec une empreinte de commit de 4 caractères :

```
git describe --abbrev={{4}}
```

- Générer un nom avec le chemin complet du tag :

```
git describe --all
```

- Décrire un tag Git :

```
git describe {{v1.0.0}}
```

- Créer un nom pour le dernier commit d'une branche donnée :

```
git describe {{nom_de_branche}}
```

# git diff

Afficher les changements sur les fichiers suivis.

Plus d'informations : <https://git-scm.com/docs/git-diff>.

- Afficher les changements sur les fichiers suivis :

```
git diff
```

- Afficher tous les changements sur les fichiers par rapport à la tête de branche :

```
git diff HEAD
```

- Afficher tous les changements sur les fichiers ajoutés mais pas encore commités :

```
git diff --staged
```

- Afficher les changements de tous les commits à partir d'une date / heure donnée (expression de dates, ex : "1 week 2 days" pour 1 semaine et 2 jours ou une date ISO) :

```
git diff 'HEAD@{3 months|weeks|days|hours|seconds ago}'
```

- Afficher seulement les noms des fichiers modifiés depuis un commit donné :

```
git diff --name-only {{commit}}
```

- Afficher un résumé des créations de fichiers, renommages ou changements de droits depuis un commit :

```
git diff --summary {{commit}}
```

- Comparer un fichier entre deux branches ou commits :

```
git diff {{branche_1}}..{{branche_2}} [--] {{chemin/vers/fichier}}
```

- Comparer plusieurs fichiers de la branche courante avec une autre branche :

```
git diff {{branche}}:{{chemin/vers/fichier2}} {{chemin/vers/fichier}}
```

# git difftool

Afficher les modifications apportées aux fichiers à l'aide d'outils de comparaison externes. Accepte les mêmes options et arguments que Git diff.

Plus d'informations : <https://git-scm.com/docs/git-difftool>.

- Lister les outils de comparaison disponibles :

```
git difftool --tool-help
```

- Configurer Meld comme outil de comparaison par défaut :

```
git config --global diff.tool "{{meld}}"
```

- Utiliser l'outil de comparaison par défaut pour afficher les fichiers modifiés :

```
git difftool --staged
```

- Utiliser un outil de comparaison spécifique (opendiff) pour afficher les changements depuis un commit :

```
git difftool --tool={{opendiff}} {{commit}}
```

# git fetch

Cherche les objets et références depuis un dépôt distant.

Plus d'informations : <https://git-scm.com/docs/git-fetch>.

- Cherche les dernières modifications du référentiel amont distant par défaut (si configuré) :

```
git fetch
```

- Cherche les nouvelles branches depuis un registre distant :

```
git fetch {{nom_distant}}
```

- Cherche les nouvelles branches depuis tous les registres distants :

```
git fetch --all
```

- Recherche également les tags depuis le registre courant :

```
git fetch --tags
```

- Supprime les références locales de branches ayant été supprimés du registre distant :

```
git fetch --prune
```

# git flow

Une collection d'extensions Git pour procurer des opérations supplémentaires sur les dépôts.

Plus d'informations : <https://github.com/nvie/gitflow>.

- Initialiser dans un registre Git existant :

```
git flow init
```

- Commencer le travail sur une fonctionnalité basé sur la branche `develop` :

```
git flow feature start {{feature}}
```

- Terminer le travail sur une branche de fonctionnalité, la fusionner dans la branche `develop` puis la supprimer :

```
git flow feature finish {{feature}}
```

- Publier une fonctionnalité sur le serveur distant :

```
git flow feature publish {{feature}}
```

- Récupérer une fonctionnalité publiée par un autre utilisateur :

```
git flow feature pull origin {{feature}}
```

# git format-patch

Préparer des fichiers de correctifs, utiles pour les envoyer par courriel.

Voir également **git am**, qui peut appliquer des fichiers de correctifs générés.

Plus d'informations : <https://git-scm.com/docs/git-format-patch>.

- Créer un fichier de correctif `.patch` nommé automatiquement pour tout les commits non poussés :

```
git format-patch {{origin}}
```

- Créer un fichier correctif `.patch` pour les changements entre 2 révisions :

```
git format-patch {{revision_1}}..{{revision_2}}
```

- Créer un fichier correctif `.patch` pour les 3 derniers commits :

```
git format-patch -{{3}}
```



# git fsck

Vérifier la validité et la connectivité des nœuds dans un dépôt Git.

N'applique aucune modification. Voir **git gc** pour nettoyer.

Plus d'informations : <https://git-scm.com/docs/git-fsck>.

- Vérifier le registre courant :

```
git fsck
```

- Lister tous les tags trouvés :

```
git fsck --tags
```

- Lister tout les nœuds racine trouvés :

```
git fsck --root
```

# git gc

Optimise le dépôt local git en nettoyant les fichiers inutiles.

Plus d'informations : <https://git-scm.com/docs/git-gc>.

- Optimise le dépôt :

```
git gc
```

- Optimise le dépôt plus agressivement, plus long :

```
git gc --aggressive
```

- Afficher les objets à supprimer :

```
git gc --no-prune
```

- Supprime tout les objets trouvés sans les afficher sur la sortie standard :

```
git gc --quiet
```

- Afficher le manuel :

```
git gc --help
```

# git-grep

Rechercher une occurrence de texte n'importe où dans l'historique d'un dépôt git.

Comprend la plupart des arguments du **grep** classique.

Plus d'informations : <https://git-scm.com/docs/git-grep>.

- Rechercher une occurrence dans les fichiers suivis :

```
git grep {{chaîne_recherché}}
```

- Rechercher une occurrence dans les fichiers suivis d'après un pattern de fichiers :

```
git grep {{chaîne_recherché}} -- {{file_glob_pattern}}
```

- Rechercher une occurrence dans les fichiers suivis et les sous-modules :

```
git grep --recurse-submodules {{chaîne_recherché}}
```

- Rechercher une occurrence à partir d'un point spécifique dans l'historique :

```
git grep {{chaîne_recherché}} {{HEAD~2}}
```

- Rechercher une occurrence dans toutes les branches :

```
git grep {{chaîne_recherché}} $(git rev-list --all)
```

# git help

Afficher le manuel de Git.

Plus d'informations : <https://git-scm.com/docs/git-help>.

- Afficher le manuel d'une sous-commande :

```
git help {{subcommand}}
```

- Même chose dans un navigateur :

```
git help --web {{subcommand}}
```

- Afficher la liste des sous-commandes disponibles :

```
git help --all
```

- Lister les manuels disponibles :

```
git help --guide
```

- Lister toutes les variables de configuration disponibles :

```
git help --config
```

# git ignore-io

Générer le fichier .gitignore depuis des templates prédéfinis.

Plus d'informations : <https://github.com/tj/git-extras/blob/master/Commands.md#git-ignore-io>.

- Lister les templates disponibles :

```
git ignore-io list
```

- Générer un template .gitignore :

```
git ignore-io {{item_a,item_b,item_n}}
```

# git-imerge

Générer un **git merge** ou un **git rebase** entre deux branches de manière incrémentale.

Les conflits entre les branches sont suivis en paires de commits individuels, pour simplifier la résolution des conflits.

Plus d'informations : <https://github.com/mhagger/git-imerge>.

- Démarrer un imerge rebase (se placer dans la branche à rebaser d'abord) :

```
git imerge rebase {{branche_sur_laquelle_rebaser}}
```

- Démarrer imerge merge (se placer dans la branche depuis laquelle merger d'abord) :

```
git imerge merge {{branche_a_merger}}
```

- Afficher le diagramme ASCII du merge ou rebase en cours :

```
git imerge diagram
```

- Continuer l'opération après une résolution de conflit (d'abord **git add** les fichiers en conflits) :

```
git imerge continue --no-edit
```

- Terminer l'opération i-merge après la résolution de tous les conflits :

```
git imerge finish
```

- Annuler l'opération et retourner à la branche précédente :

```
git-imerge remove && git checkout {{previous_branch}}
```

# git init

Initialise un nouveau registre Git.

Plus d'informations : <https://git-scm.com/docs/git-init>.

- Initialise un nouveau registre Git local :

```
git init
```

- Initialiser un référentiel barebones, adapté à une utilisation distante via SSH :

```
git init --bare
```

# git instaweb

Outil pour le lancement d'un serveur gitweb.

Plus d'informations : <https://git-scm.com/docs/git-instaweb>.

- Démarre un serveur gitweb depuis le dépôt (repository) courant :

```
git instaweb --start
```

- Écoute uniquement sur le port localhost :

```
git instaweb --start --local
```

- Écoute sur un port spécifique :

```
git instaweb --start --port {{1234}}
```

- Utiliser un daemon HTTP spécifique :

```
git instaweb --start --httpd {{lighttpd|apache2|mongoose|  
plackup|webrick}}
```

- Lancer en même temps qu'un navigateur web :

```
git instaweb --start --browser
```

- Stoppe le serveur :

```
git instaweb --stop
```

- Redémarre le serveur :

```
git instaweb --restart
```



# git lfs

Travailler dans un registre Git avec des fichiers volumineux.

Plus d'informations : <https://git-lfs.com>.

- Initialise le Git LFS :

```
git lfs install
```

- Suivre des fichiers correspondant à un pattern :

```
git lfs track '{{*.bin}}'
```

- Changer l'URL du point de terminaison Git LFS (utile si le serveur LFS est séparé du serveur Git) :

```
git config {{-f|--file}} .lfsconfig lfs.url  
{{lfs_endpoint_url}}
```

- Lister les pattern de fichiers suivis :

```
git lfs track
```

- Lister les fichiers suivis ayant été commité :

```
git lfs ls-files
```

- Pousser tout les objets LFS vers le serveur distant :

```
git lfs push --all {{nom_distant}} {{nom_de_branche}}
```

- Chercher tout les objets LFS :

```
git lfs fetch
```

- Charger tout les objets LFS :

```
git lfs checkout
```

# git log

Afficher un historique de commits.

Plus d'informations : <https://git-scm.com/docs/git-log>.

- Afficher la séquence de commits à partir de l'actuel, dans l'ordre chronologique inverse du dépôt Git dans le répertoire de travail actuel :

```
git log
```

- Afficher l'historique de fichiers ou répertoires en particulier :

```
git log {{-p|-u|--patch}} {{chemin/vers/  
fichier_ou_repertoire}}
```

- Afficher la liste des fichiers modifiés pour chaque commit :

```
git log --stat
```

- Afficher un graphique des commits dans la branche actuelle en utilisant uniquement la première ligne de chaque message de commit :

```
git log --oneline --graph
```

- Afficher un graphique de tout les commits, tags et branches dans le dépôt entier :

```
git log --oneline --decorate --all --graph
```

- Afficher uniquement les commits dont le message contient la chaîne (non sensible à la casse) :

```
git log {{-i|--regexp-ignore-case}} --grep  
{{chaîne_recherché}}
```

- Afficher les N derniers commits d'un utilisateur :

```
git log {{-n|--max-count}} {{number}} --author "{{author}}"
```

- Afficher les commits entre deux dates (yyyy-mm-dd):

```
git log --before "{{2017-01-29}}" --after "{{2017-01-17}}"
```

# git ls-remote

Commande Git pour répertorier les références dans un dépôt distant en fonction du nom ou de l'URL.

Si aucun nom ou URL n'est donné, alors la branche amont configurée sera utilisée, ou l'origine distante si la première n'est pas configurée.

Plus d'informations : <https://git-scm.com/docs/git-ls-remote>.

- Afficher les références du dépôt configuré par défaut :

```
git ls-remote
```

- Afficher uniquement les références HEAD du dépôt configuré par défaut :

```
git ls-remote --heads
```

- Afficher uniquement les tags du dépôt configuré par défaut :

```
git ls-remote --tags
```

- Afficher les références du dépôt précisé :

```
git ls-remote {{url-du-dépôt}}
```

- Afficher les références du dépôt précisé filtrés par un motif :

```
git ls-remote {{nom-du-dépôt}} "{{motif}}"
```

# git ls-tree

Lister le contenu d'un arbre.

Plus d'informations : <https://git-scm.com/docs/git-ls-tree>.

- Lister le contenu de l'arbre dans la branche :

```
git ls-tree {{nom_de_branche}}
```

- Lister le contenu de l'arbre dans le commit, récursif avec les sous-arbres :

```
git ls-tree -r {{commit_hash}}
```

- Lister uniquement les noms de fichiers de l'arbre dans un commit :

```
git ls-tree --name-only {{commit_hash}}
```

# git merge

Pour fusionner des branches **git**.

Plus d'informations : <https://git-scm.com/docs/git-merge>.

- Fusionne une branche dans votre branche courante :

```
git merge {{nom_de_branche}}
```

- Editer le message de fusion (merge commit) :

```
git merge --edit {{nom_de_branche}}
```

- Fusionner une branche et créer un commit de fusion (merge commit) :

```
git merge --no-ff {{nom_de_branche}}
```

- Annuler une fusion en cas de conflit :

```
git merge --abort
```

# git mergetool

Executer un utilitaire de différences pour résoudre les conflits de merge.

Plus d'informations : <https://git-scm.com/docs/git-mergetool>.

- Démarrer l'outil de différences par défaut :

```
git mergetool
```

- Lister les outils de différences valides :

```
git mergetool --tool-help
```

- Démarrer l'outil de différences en précisant son nom :

```
git mergetool --tool {{tool_name}}
```

- Démarrer l'outil de différences sans dialogues :

```
git mergetool --no-prompt
```

- Utiliser explicitement l'outil de différences graphique (voir la variable de config `merge.guitool`):

```
git mergetool --gui
```

- Utiliser explicitement l'outil de différences classique (voir la variable de config `merge.tool`):

```
git mergetool --no-gui
```

# git mv

Déplacer ou renommer des fichiers inscrits dans l'index.

Plus d'informations : <https://git-scm.com/docs/git-mv>.

- Déplace les fichiers dans l'index Git, valide au prochain commit :

```
git mv {{chemin/vers/fichier}} {{new/path/to/file}}
```

- Renome un fichier et met a jour l'index, valide au prochain commit :

```
git mv {{filename}} {{new_filename}}
```

- Force l'écrasement d'un fichier :

```
git mv --force {{file}} {{cible}}
```

# git notes

Ajoute ou inspecte des notes d'objets.

Plus d'informations : <https://git-scm.com/docs/git-notes>.

- Lister toutes les notes et leurs objets rattachés :

```
git notes list
```

- Lister toutes les notes attachées à un objet donné :

```
git notes list [{{objet}}]
```

- Afficher les notes attachées à un objet donné :

```
git notes show [{{objet}}]
```

- Ajoute une note à un objet donné :

```
git notes append {{objet}}
```

- Ajoute une note à un objet donné, en spécifiant le message :

```
git notes append --message="{{texte_du_message}}"
```

- Édite une note existante :

```
git notes edit [{{objet}}]
```

- Copie la note d'un objet vers un autre :

```
git notes copy {{objet_source}} {{objet_cible}}
```

- Supprime toutes les notes d'un objet donné :

```
git notes remove {{objet}}
```



# git pr

Récupère les pull-requests GitHub localement.

Plus d'informations : <https://github.com/tj/git-extras/blob/master/Commands.md#git-pr>.

- Récupère une pull-request spécifique :

```
git pr {{pr_number}}
```

- Récupère une pull-request d'un dépôt spécifique :

```
git pr {{pr_number}} {{distant}}
```

- Récupère une pull-request depuis son URL :

```
git pr {{url}}
```

- Nettoie les branches de pull-requests terminées :

```
git pr clean
```

# git pull

Récupère une branche depuis le serveur distant et la fusionne dans la branche local.

Plus d'informations : <https://git-scm.com/docs/git-pull>.

- Télécharge les changements depuis le serveur distant par défaut et fusionne les :

```
git pull
```

- Télécharge les changements depuis le serveur distant par défaut et applique les changements locaux par dessus :

```
git pull --rebase
```

- Télécharge les changements depuis un serveur et une branche distante, puis fusionne les dans HEAD :

```
git pull {{nom_distant}} {{branche}}
```

# git push

Pousse les commits vers un dépôt distant.

Plus d'informations : <https://git-scm.com/docs/git-push>.

- Envoie les changements locaux dans la branche courante vers sa contrepartie distante :

```
git push
```

- Envoie les changements locaux d'une branche spécifique vers sa contrepartie distante :

```
git push {{nom_distant}} {{local_branch}}
```

- Publie la branche courante vers un dépôt distant, crée le nom de la branche distante :

```
git push {{nom_distant}} -u {{branche_distante}}
```

- Envoi les changements locaux sur toutes les branches locales vers leur contrepartie sur le dépôt distant :

```
git push --all {{nom_distant}}
```

- Supprime une branche dans un dépôt distant :

```
git push {{nom_distant}} --delete {{branche_distante}}
```

- Supprime les branches distantes qui n'ont pas de contrepartie locale :

```
git push --prune {{nom_distant}}
```

- Publie les tags qui ne sont pas sur le dépôt distant :

```
git push --tags
```

# git rebase

Rejoue les commits d'une branche par dessus une autre.

Communément utilisé pour dupliquer les commits d'une branche dans une autre, en créant de nouveaux commits dans la branche de destination.

Plus d'informations : <https://git-scm.com/docs/git-rebase>.

- Rejouer les commits de la branche courante sur la branche master :

```
git rebase {{master}}
```

- Rejouer les comits interactivement, ce qui permet aux commits d'être re-arrangés, exclus, combinés ou modifiés :

```
git rebase {{-i|--interactive}} {{branche_de_base_ou_commit}}
```

- Continuer le re-jeu des commits après la résolution d'un conflit :

```
git rebase --continue
```

- Continuer le re-jeu des commits en sautant la résolution d'un conflit :

```
git rebase --skip
```

- Annule l'opération (ex : en cas de conflit) :

```
git rebase --abort
```

- Déplacez une partie de la branche actuelle sur une nouvelle base, fournissant l'ancienne base à partir de laquelle commencer :

```
git rebase --onto {{new_base}} {{old_base}}
```

- Rejoue les 5 derniers commits, ce qui permet aux commits d'être re-arrangés, exclus, combinés ou modifiés :

```
git rebase {{-i|--interactive}} {{HEAD~5}}
```

- Résoudre automatiquement les conflits en précisant la version à conserver (theirs signifie la version des fichiers à privilégier) :

```
git rebase {{-X|--strategy-option}} theirs {{master}}
```

# git reflog

Affiche un log des changements locaux comme HEAD, tags et branches.

Plus d'informations : <https://git-scm.com/docs/git-reflog>.

- Afficher le reflog de HEAD :

```
git reflog
```

- Affiche le reflog d'une branche spécifique :

```
git reflog {{nom_de_branche}}
```

- Affiche les 5 dernières entrées dans le reflog :

```
git reflog {{-n|--max-count}} 5
```

# git remote

Organisation des dépôts suivis ("remotes").

Plus d'informations : <https://git-scm.com/docs/git-remote>.

- Affiche les dépôts existants, leur nom et URL :

```
git remote {{-v|--verbose}}
```

- Affiche les informations à propos d'un dépôt :

```
git remote show {{nom_distant}}
```

- Ajoute un dépôt :

```
git remote add {{nom_distant}} {{url_distant}}
```

- Change l'URL d'un dépôt (ajouter --add pour conserver l'URL existante) :

```
git remote set-url {{nom_distant}} {{new_url}}
```

- Supprime un dépôt :

```
git remote remove {{nom_distant}}
```

- Renomme un dépôt :

```
git remote rename {{old_name}} {{new_name}}
```

# git repack

Empaqueter les objets décompressés dans un dépôt Git.

Plus d'informations : <https://git-scm.com/docs/git-repack>.

- Empaqueter les objets décompressés dans le dépôt courant :

```
git repack
```

- Également supprimer les objets redondants après empaquetage :

```
git repack -d
```

# git request-pull

Générer une requête demandant au projet en amont d'inclure les modifications dans son arborescence.

Plus d'informations : <https://git-scm.com/docs/git-request-pull>.

- Produire une requête résumant les changements entre la version v1.1 et le master :

```
git request-pull {{v1.1}} {{https://example.com/project}}  
{{master}}
```

- Produire une requête résumant les changements entre la version v1.1 sur la branche master et la branche locale foo :

```
git request-pull {{v0.1}} {{https://example.com/project}}  
{{master:foo}}
```



# git reset

Enlève des commits ou des changements en réinitialisant la tête Git à l'état spécifié.

Si un chemin est passé en paramètre, Git reset fonctionne comme «unstage».

Si un hash de commit est passé en paramètre, Git reset annule les commits jusqu'à ce dernier.

Plus d'informations : <https://git-scm.com/docs/git-reset>.

- Tout enlever de la zone de stage :

```
git reset
```

- Enlever des fichiers spécifiques de la zone de stage :

```
git reset {{chemin/vers/fichier(s)}}
```

- Enlever, en mode interactif, des fichiers spécifiques de l'index :

```
git reset --patch {{chemin/vers/fichier}}
```

- Annuler le dernier commit, mais garder les changements effectués dans votre système de fichiers :

```
git reset HEAD~
```

- Défaire les deux derniers commits, et ajouter leurs changements à l'index (dans la zone de stage) :

```
git reset --soft HEAD~2
```

- Enlever tout les changements qui n'ont pas été commit, qu'ils soient dans la zone de stage ou non (pour enlever seulement les changements de la zone de stage, utiliser `git checkout`) :

```
git reset --hard
```

- Réinitialiser le dépôt à un commit spécifique en retirant tout les changements (ceci inclus les changements dans des commits entre la tête et le commit spécifié !) :

```
git reset --hard {{commit}}
```

# git restore

Restaurez les fichiers de l'arborescence de travail. Nécessite la version 2.23+ de Git.

Voir aussi **git checkout**.

Plus d'informations : <https://git-scm.com/docs/git-restore>.

- Restaurer un fichier supprimé à partir du contenu du commit actuel (HEAD) :

```
git restore {{chemin/vers/fichier}}
```

- Restaurer un fichier a la version d'un commit spécifié :

```
git restore --source {{commit}} {{chemin/vers/fichier}}
```

- Annulez toutes les modifications non validées des fichiers suivis, en revenant au HEAD :

```
git restore .
```

# git rev-list

Liste les révisions (commits) dans l'ordre chronologique inverse.

Plus d'informations : <https://git-scm.com/docs/git-rev-list>.

- Lister tout les commits dans la branche courante :

```
git rev-list {{HEAD}}
```

- Lister tout les commits plus récents qu'une date spécifique, sur une branche spécifique :

```
git rev-list --since={{'2019-12-01 00:00:00'}} {{master}}
```

- Lister tout les commits de merge depuis un commit spécifique :

```
git rev-list --merges {{commit}}
```

# git rev-parse

Afficher les métadonnées liées à des révisions spécifiques.

Plus d'informations : <https://git-scm.com/docs/git-rev-parse>.

- Afficher l'empreinte du commit de la branche courante :

```
git rev-parse {{nom_de_branche}}
```

- Affiche le nom de la branche courante :

```
git rev-parse --abbrev-ref {{HEAD}}
```

- Obtenir le chemin absolu du répertoire racine :

```
git rev-parse --show-toplevel
```

# git revert

Créer un nouveau commit qui efface les changements du précédent.

Plus d'informations : <https://git-scm.com/docs/git-revert>.

- Crée un commit qui annule les changements du dernier commit :

```
git revert {{HEAD}}
```

- Crée un commit qui annule les changements des 5 dernier commit :

```
git revert HEAD~{{4}}
```

- Crée un commit qui annule les changements de plusieurs commit :

```
git revert {{master~5..master~2}}
```

- Ne pas créer de nouveau commit, remplacer uniquement dans l'arbre courant :

```
git revert -n {{0c01a9..9a1743}}
```

# git rm

Supprimer des fichiers de l'index, du dépôt et du système de fichiers.

Plus d'informations : <https://git-scm.com/docs/git-rm>.

- Supprimer un fichiers de l'index, du dépôt et du système de fichiers :

```
git rm {{file}}
```

- Supprimer un répertoire de l'index, du dépôt et du système de fichiers :

```
git rm -r {{directory}}
```

- Supprimer un fichiers de l'index, du dépôt mais le conserve sur le système de fichiers :

```
git rm --cached {{file}}
```

# git send-email

Envoyer une collection de correctifs par email.

Les correctifs peuvent être spécifiés sous forme de fichiers, de directions ou de liste de révisions.

Plus d'informations : <https://git-scm.com/docs/git-send-email>.

- Envoyer le dernier commit de la branche courante :

```
git send-email -1
```

- Envoyer un commit spécifique :

```
git send-email -1 {{commit}}
```

- Envoyer de multiples commits de la branche courante (ici : 10) :

```
git send-email {{-10}}
```

- Envoyez un e-mail de présentation de la série de correctifs :

```
git send-email -{{number of commits}} --compose
```

- Consultez et modifiez l'e-mail de chaque patch que vous êtes sur le point d'envoyer :

```
git send-email -{{number of commits}} --annotate
```

# git shortlog

Récapitule la sortie de **git log**.

Plus d'informations : <https://git-scm.com/docs/git-shortlog>.

- Afficher un résumé de tous les commits effectués, regroupés par ordre alphabétique par nom d'auteur :

```
git shortlog
```

- Afficher un résumé de tous les commits effectués, regroupés par le nombre de commits effectués :

```
git shortlog {{-n|--numbered}}
```

- Afficher un résumé de tous les commits effectués, regroupés par le nom et l'email de l'utilisateur :

```
git shortlog {{-c|--committer}}
```

- Afficher un résumé des 5 derniers commits effectués :

```
git shortlog HEAD~5..HEAD
```

- Afficher tout les utilisateurs, emails et le nombre de commits dans la branche :

```
git shortlog {{-s|--summary}} {{-n|--numbered}} {{-e|--email}}
```

- Afficher tout les utilisateurs, emails et le nombre de commits dans toutes les branches :

```
git shortlog {{-s|--summary}} {{-n|--numbered}} {{-e|--email}} --all
```



# git show-branch

Affiche les branches et leurs commits.

Plus d'informations : <https://git-scm.com/docs/git-show-branch>.

- Affiche un résumé du dernier commit dans la branche :

```
git show-branch {{nom_de_branche|ref|commit}}
```

- Comparer des commits avec plusieurs commits ou branches :

```
git show-branch {{nom_de_branche|ref|commit}}
```

- Comparer toutes les branches distantes :

```
git show-branch --remotes
```

- Comparer la branche locale avec la branche distante :

```
git show-branch --all
```

- Lister les derniers commits sur toutes les branches :

```
git show-branch --all --list
```

- Comparer une branche spécifique à la branche courante :

```
git show-branch --current {{commit|nom_de_branche|ref}}
```

- Afficher le nom du commit au lieu du nom relatif :

```
git show-branch --sha1-name --current {{current|  
nom_de_branche|ref}}
```

- Continuez l'affichage d'un certain nombre de commits au-delà de l'ancêtre commun :

```
git show-branch --more {{5}} {{commit|nom_de_branche|ref}}  
{{commit|nom_de_branche|ref}} {{...}}
```

# git show-ref

Commande Git pour lister les références.

Plus d'informations : <https://git-scm.com/docs/git-show-ref>.

- Affiche toutes les références dans le dépôt :

```
git show-ref
```

- Affiche seulement les références des têtes de branches :

```
git show-ref --heads
```

- Affiche seulement les références de tags :

```
git show-ref --tags
```

- Vérifier l'existence d'une référence :

```
git show-ref --verify {{chemin/vers/reference}}
```

# git show

Affiche différents types d'objets Git (commits, tags, etc.).

Plus d'informations : <https://git-scm.com/docs/git-show>.

- Afficher des informations sur le dernier commit (hachage, message, modifications et autres métadonnées) :

```
git show
```

- Affiche les informations du dernier commit :

```
git show {{commit}}
```

- Affiche les informations associés au tag spécifié :

```
git show {{etiquette}}
```

- Affiche les informations à propos du 3ème commit en partant du sommet de la branche :

```
git show {{branche}}~{{3}}
```

- Afficher le message d'un commit sur une seule ligne, en supprimant la sortie diff :

```
git show --oneline -s {{commit}}
```

- Affiche uniquement la liste des fichiers changés dans un commit :

```
git show --stat {{commit}}
```

- Afficher le contenu d'un fichier tel qu'il était à une révision donnée (par exemple, branche, tag ou commit) :

```
git show {{revision}}:{{chemin/vers/fichier}}
```

# git sizer

Calcule diverses métriques de taille de dépôt Git et vous alerte de tout ce qui pourrait causer des problèmes ou des inconvénients.

Plus d'informations : <https://github.com/github/git-sizer>.

- Signaler uniquement les statistiques dont le niveau de préoccupation est supérieur à 0 :

```
git sizer
```

- Signaler toutes les statistiques :

```
git sizer -v
```

- Afficher les options additionnelles :

```
git sizer -h
```

# git stage

Cette commande est un alias de **git add**.

Plus d'informations : <https://git-scm.com/docs/git-stage>.

- Voir la documentation de la commande originale :

```
tldr git add
```

# git stash

Stocker les modifications Git locales dans une zone temporaire.

Plus d'informations : <https://git-scm.com/docs/git-stash>.

- Stocker les changements courants, sauf les fichiers non-suivis :

```
git stash push -m {{nom_de_stash_optionel}}
```

- Stocker les changements courants, incluant les fichiers non-suivis :

```
git stash -u
```

- Stocker les parties d'un fichier interactivement :

```
git stash -p
```

- Lister tous les stashes (affiche leurs noms, les branches relatives et messages) :

```
git stash list
```

- Applique un stash (par défaut, le dernier, nommé stash@{0}) :

```
git stash apply {{nom_de_stash_ou_de_commit_optionel}}
```

- Applique un stash (par défaut le dernier, stash@{0}), et le supprimer de la liste des stashes s'il n'y a pas de conflit :

```
git stash pop {{nom_de_stash_optionel}}
```

- Supprime tous les stashes :

```
git stash clear
```

# git status

Affiche les changements sur les fichiers dans la branche courante.

Affiche les fichiers édités, déplacés, renommés, ajoutés, supprimés par rapport à la référence de la branche courante.

Plus d'informations : <https://git-scm.com/docs/git-status>.

- Affiche les fichiers modifiés qui n'ont pas encore été ajoutés pour le commit :

```
git status
```

- Affiche les fichiers modifiés (version courte) :

```
git status -s
```

- Affiche les fichiers modifiés, sans tenir des comptes des fichiers non-suivis :

```
git status --untracked-files=no
```

- Affiche les fichiers modifiés (version courte) avec les informations de branche :

```
git status -sb
```

# git submodule

Inspecter, mettre à jour et manager des sous-modules.

Plus d'informations : <https://git-scm.com/docs/git-submodule>.

- Installer un sous-module depuis le dépôt courant :

```
git submodule update --init --recursive
```

- Ajout d'un dépôt Git en tant que sous-module :

```
git submodule add {{repository_url}}
```

- Ajout d'un dépôt Git en tant que sous-module à un répertoire donné :

```
git submodule add {{repository_url}} {{chemin/vers/repertoire}}
```

- Mettre à jour tout les sous-modules à leur dernier commit :

```
git submodule foreach git pull
```



# git subtree

Outil pour gérer les dépendances de projet en tant que sous-projets.

Plus d'informations : <https://manpages.debian.org/latest/git-man/git-subtree.1.html>.

- Ajout d'un dépôt Git en tant que sous-arbre :

```
git subtree add --prefix={{chemin/vers/repertoire/}} --squash  
{{repository_url}} {{master}}
```

- Mettre à jour le sous-arbre avec son dernier commit :

```
git subtree pull --prefix={{chemin/vers/repertoire/}}  
{{repository_url}} {{master}}
```

- Merge le dépôt d'un sous arbre dans la branche master :

```
git subtree merge --prefix={{chemin/vers/repertoire/}} --  
squash {{repository_url}} {{master}}
```

- Pousser les commits vers le dépôt d'un sous-arbre :

```
git subtree push --prefix={{chemin/vers/repertoire/}}  
{{repository_url}} {{master}}
```

- Extraire un nouvel historique de projet de l'historique d'un sous-arbre :

```
git subtree split --prefix={{chemin/vers/repertoire/}}  
{{repository_url}} -b {{nom_de_branche}}
```

# git svn

Opération bidirectionnelle entre un référentiel Subversion et Git.

Plus d'informations : <https://git-scm.com/docs/git-svn>.

- Cloner un dépôt SVN :

```
git svn clone {{https://example.com/subversion_repo}}  
{{local_dir}}
```

- Cloner un dépôt SVN à partir d'une révision donnée :

```
git svn clone {{-r|--revision}} {{1234}}:HEAD {{https://  
svn.example.net/subversion/repo}} {{local_dir}}
```

- Mettre à jour le clone local à partir du dépôt SVN distant :

```
git svn rebase
```

- Chercher les changements distants dans le dépôt SVN dans les appliquer sur le HEAD :

```
git svn fetch
```

- Commiter sur le SVN :

```
git svn commit
```

# git switch

Basculez entre les branches Git. Nécessite la version 2.23+ de Git.

Voir également **git checkout**.

Plus d'informations : <https://git-scm.com/docs/git-switch>.

- Basculer sur une branche existante :

```
git switch {{nom_de_branche}}
```

- Créer une nouvelle branche et basculer dessus :

```
git switch --create {{nom_de_branche}}
```

- Créer une nouvelle branche en partant d'un commit donné et basculer dessus :

```
git switch --create {{nom_de_branche}} {{commit}}
```

- Basculer sur la branche précédente :

```
git switch -
```

- Basculer vers une branche et mettre à jour tous les sous-modules pour qu'ils correspondent :

```
git switch --recurse-submodules {{nom_de_branche}}
```

- Basculer vers une branche et fusionner automatiquement la branche actuelle et toutes les modifications non validées dedans :

```
git switch --merge {{nom_de_branche}}
```

# git tag

Créer, lister, vérifier et supprimer des tags.

Un tag est une référence statique vers un commit.

Plus d'informations : <https://git-scm.com/docs/git-tag>.

- Lister tout les tags :

```
git tag
```

- Créer un tag avec le nom donné pointant vers le commit actuel :

```
git tag {{nom_d_etiquette}}
```

- Créer un tag avec le nom donné pointant vers un commit spécifié :

```
git tag {{nom_d_etiquette}} {{commit}}
```

- Créer un tag annoté avec le message spécifié :

```
git tag {{nom_d_etiquette}} -m {{message_d_etiquette}}
```

- Supprimer le tag avec le nom spécifié :

```
git tag -d {{nom_d_etiquette}}
```

- Mettre à jour les tags depuis l'origine :

```
git fetch --tags
```

- Liste toutes les tags dont les ancêtres incluent un commit donné :

```
git tag --contains {{commit}}
```

# git update-index

Commande Git pour manipuler l'index.

Plus d'informations : <https://git-scm.com/docs/git-update-index>.

- Prétendre qu'un fichier modifié est inchangé (`git status` ne l'affichera pas comme modifié) :

```
git update-index --skip-worktree {{chemin/vers/  
fichier_modifié}}
```

# git update-ref

Commande Git pour créer, mettre à jour et supprimer des références Git.

Plus d'informations : <https://git-scm.com/docs/git-update-ref>.

- Supprimer une référence, utile pour la réinitialisation du premier commit :

```
git update-ref -d {{HEAD}}
```

- Mettre à jour une référence avec un message :

```
git update-ref -m {{message}} {{HEAD}} {{4e95e05}}
```

# git worktree

Gérer plusieurs arbres de travail attachés au même dépôt.

Plus d'informations : <https://git-scm.com/docs/git-worktree>.

- Créer un nouvel arbre de travail avec une branche spécifiée :

```
git worktree add {{chemin/vers/répertoire}} {{branche}}
```

- Créer un nouvel arbre de travail avec une nouvelle branche :

```
git worktree add {{chemin/vers/répertoire}} -b  
{{nouvelle_branche}}
```

- Répertorier tous les arbres de travail attachés à ce dépôt :

```
git worktree list
```

- Supprimer les arbres de travail (après avoir supprimé les répertoires de travail) :

```
git worktree prune
```

# git

Système de gestion de versions décentralisé.

Certaines commandes comme **git commit** ont leur propre documentation.

Plus d'informations : <https://git-scm.com/>.

- Exécuter une sous-commande Git :

```
git {{sous_commande}}
```

- Exécuter une sous-commande Git sur un répertoire personnalisé :

```
git -C {{chemin/vers/repertoire}} {{sous_commande}}
```

- Exécuter une sous-commande Git avec un paramètre de configuration spécifique :

```
git -c '{{cle_param_config}}={{valeur}}' {{sous_commande}}
```

- Afficher l'aide générale :

```
git --help
```

- Afficher l'aide sur une sous-commande Git :

```
git help {{sous_commande}}
```

- Obtenir la version de Git :

```
git --version
```



# gnmic sub

Cette commande est un alias de **gnmic subscribe**.

Plus d'informations : <https://gnmic.kmrd.dev/cmd/subscribe>.

- Voir la documentation de la commande originale :

```
tldr gnmic subscribe
```

# go

Outil de gestion du code source Go.

Certaines sous-commandes telles que **build** ont leur propre documentation d'utilisation.

Plus d'informations : <https://golang.org>.

- Télécharger et installer un paquet, spécifié par son chemin d'importation :

```
go get {{chemin_du_paquet}}
```

- Compiler et exécuter un fichier source (il doit contenir un paquet `main`) :

```
go run {{fichier}}.go
```

- Compiler un fichier source dans un exécutable nommé :

```
go build -o {{executable}} {{fichier}}.go
```

- Compile le paquet présent dans le répertoire courant :

```
go build
```

- Exécuter tous les cas de test du paquet courant (les fichiers doivent se terminer par `_test.go`) :

```
go test
```

- Compiler et installer le paquet actuel :

```
go install
```

- Initialiser un nouveau module dans le répertoire courant :

```
go mod init {{nom_du_module}}
```

# google-chrome

Cette commande est un alias de **chromium**.

Plus d'informations : <https://chrome.google.com>.

- Voir la documentation de la commande originale :

```
tldr chromium
```

# grep

Recherche des motifs dans un texte.

Supporte des motifs simples et des expressions régulières.

Plus d'informations : <https://www.gnu.org/software/grep/manual/grep.html>.

- Recherche une chaîne de caractères précise :

```
grep "{{chaîne_recherchée}}" {{chemin/vers/fichier}}
```

- Recherche en ignorant la casse :

```
grep {{-F|--fixed-strings}} "{{chaîne_recherchée}}" {{chemin/vers/fichier}}
```

- Recherche récursivement (en ignorant les fichiers non-texte) dans le dossier courant une chaîne de caractères précise :

```
grep {{-r|--recursive}} {{-n|--line-number}}  
"{{chaîne_recherchée}}" .
```

- Utilise des expressions régulières étendues (supporte ?, +, {}, () et |):

```
grep {{-E|--extended-regexp}} {{-i|--ignore-case}}  
{{expression_régulière}} {{chemin/vers/fichier}}
```

- Affiche 3 lignes de [C]ontexte, avant ([B]efore), ou [A]près chaque concordance :

```
grep --{{context|before-context|after-context}} 3  
"{{chaîne_recherchée}}" {{chemin/vers/fichier}}
```

- Affiche le nom du fichier avec la ligne correspondante pour chaque concordance :

```
grep {{-H|--with-filename}} {{-n|--line-number}} --  
color=always "{{chaîne_recherchée}}" {{chemin/vers/fichier}}
```

- Utilise l'entrée standard au lieu d'un fichier :

```
cat {{chemin/vers/fichier}} | grep "{{chaîne_recherchée}}"
```

# gulp

Système d'exécution de tâches et de construction en continu pour JavaScript.

Les tâches sont définies dans le fichier **gulpfile.js** à la racine du projet.

Plus d'informations : <https://github.com/gulpjs/gulp-cli>.

- Exécute la tâche par défaut :

```
gulp
```

- Exécute des tâches individuelles :

```
gulp {{tâche}} {{autre_tâche}}
```

- Affiche l'arbre de dépendance des tâches pour le gulpfile chargé :

```
gulp --tasks
```

# hashcat

Outil de récupération de mot de passe rapide et avancé.

Plus d'informations : <https://hashcat.net/wiki/doku.php?id=hashcat>.

- Effectue une attaque par force brute (mode 3) avec le masque hashcat par défaut :

```
hashcat --hash-type {{id_type_hash}} --attack-mode {{3}}  
{{valeur_hash}}
```

- Effectue une attaque par force brute (mode 3) avec un motif connu de 4 chiffres :

```
hashcat --hash-type {{id_type_hash}} --attack-mode {{3}}  
{{valeur_hash}} "{{{d?d?d?d}}"
```

- Effectue une attaque par force brute (mode 3) en utilisant au plus 8 caractères parmi tous les caractères ASCII imprimables :

```
hashcat --hash-type {{id_type_hash}} --attack-mode {{3}} --  
increment {{valeur_hash}} "{{{?a?a?a?a?a?a?a?a}}"
```

- Effectue une attaque par dictionnaire (mode 0) en utilisant la liste de mots RockYou d'une machine Kali Linux :

```
hashcat --hash-type {{id_type_hash}} --attack-mode {{0}}  
{{valeur_hash}} {{/usr/share/wordlists/rockyou.txt}}
```

- Effectue une attaque par dictionnaire basée sur des règles (mode 0) en utilisant la liste de mots RockYou modifiée avec des variations de mots de passe courants :

```
hashcat --hash-type {{id_type_hash}} --attack-mode {{0}} --  
rules-file {{/usr/share/hashcat/rules/best64.rule}}  
{{valeur_hash}} {{/usr/share/wordlists/rockyou.txt}}
```

- Effectue une attaque combinée (mode 1) en utilisant la concaténation de mots provenant de deux dictionnaires personnalisés différents :

```
hashcat --hash-type {{id_type_hash}} --attack-mode {{1}}  
{{valeur_hash}} {{/chemin/vers/dictionnaire1.txt}} {{/chemin/  
vers/dictionnaire2.txt}}
```

- Affiche le résultat d'un hash déjà craqué :

```
hashcat --show {{valeur_hash}}
```

# hexyl

Un simple visualiseur hexadécimal pour le terminal. Il utilise des couleurs pour distinguer les différentes catégories d'octets.

Plus d'informations : <https://github.com/sharkdp/hexyl>.

- Affiche la représentation hexadécimale d'un fichier :

```
hexyl {{chemin/vers/fichier}}
```

- Affiche la représentation hexadécimale des n premiers octets d'un fichier :

```
hexyl -n {{n}} {{chemin/vers/fichier}}
```

- Affiche les octets 512 à 1024 d'un fichier :

```
hexyl -r {{512}}:{{1024}} {{chemin/vers/fichier}}
```

- Affiche 512 octets en commençant par le 1024e octet :

```
hexyl -r {{1024}}:+{{512}} {{chemin/vers/fichier}}
```

# history expansion

Réutiliser et développer l'historique des commandes shell dans **sh**, **bash**, **zsh**, **rbash** et **ksh**.

Plus d'informations : [https://www.gnu.org/software/bash/manual/html\\_node/History-Interaction](https://www.gnu.org/software/bash/manual/html_node/History-Interaction).

- Exécute de nouveau la commande précédente en tant que root (! ! est remplacé par la commande précédente) :

```
sudo !!
```

- Exécute une commande avec le dernier argument de la commande précédente :

```
{{commande}} !$
```

- Exécute une commande avec le premier argument de la commande précédente :

```
{{command}} !^
```

- Exécute la n-ème commande de l'historique, en partant de la plus ancienne :

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

- Exécute la n-ème commande de l'historique, en partant de la plus récente :

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

- Exécute la commande contenant **string** la plus récente :

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

- Exécute la commande précédente, en remplaçant **string1** par **string2** :

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

- Effectue une expansion de l'historique, mais affiche la commande qui aurait du être exécutée au lieu de l'exécuter :

```
{{!-n}}:p
```



# hping

Cette commande est un alias de **hping3**.

Plus d'informations : <https://github.com/antirez/hping>.

- Voir la documentation de la commande originale :

```
tldr hping3
```

# hx

Cette commande est un alias de **helix**.

- Voir la documentation de la commande originale :

`tldr helix`

# ifconfig

Configurateur des interfaces réseau.

Plus d'informations : <https://net-tools.sourceforge.io/man/ifconfig.8.html>.

- Affiche les paramètres de réseau d'un adaptateur ethernet :

```
ifconfig eth0
```

- Affiche les détails de toutes les interfaces, y compris les interfaces désactivées :

```
ifconfig -a
```

- Désactive l'interface eth0 :

```
ifconfig eth0 down
```

- Active l'interface eth0 :

```
ifconfig eth0 up
```

- Assigne une adresse IP à l'interface eth0 :

```
ifconfig eth0 {{adresse_ip}}
```

# install

Copie des fichiers et met à jour leurs attributs.

Copie des fichiers (souvent des exécutables) dans un répertoire système comme **/usr/local/bin** et leur donne les permissions et propriétaires appropriés.

Plus d'informations : <https://www.gnu.org/software/coreutils/install>.

- Copie des fichiers vers une destination :

```
install {{chemin/fichier/source}} {{chemin/repertoire/destination}}
```

- Copie des fichiers vers une destination en mettant à jour leur propriétaire :

```
install --owner {{utilisateur}} {{chemin/fichier/source}} {{chemin/repertoire/destination}}
```

- Copie des fichiers vers une destination en mettant à jour leur groupe d'appartenance :

```
install --group {{utilisateur}} {{chemin/fichier/source}} {{chemin/repertoire/destination}}
```

- Copie des fichiers vers une destination en mettant à jour leur mode :

```
install --mode {{+x}} {{chemin/fichier/source}} {{chemin/repertoire/destination}}
```

- Copie des fichiers en mettant à jour leur dates d'accès et de modification à partir de leurs sources respectives :

```
install --preserve-timestamps {{chemin/fichier/source}} {{chemin/repertoire/destination}}
```

# jeekyll

Générateur de site statique simple, adapté aux blogs.

Plus d'informations : <https://jeekyllrb.com/docs/usage/>.

- Génère un serveur de développement qui tourne en `http://localhost:4000/` :

```
jeekyll serve
```

- Active la régénération incrémentale :

```
jeekyll serve --incremental
```

- Active la sortie verbeuse :

```
jeekyll serve --verbose
```

- Génère le répertoire actuel dans `./_site` :

```
jeekyll build
```

- Nettoie le site (c.-à.-d. supprime la sortie du site et le répertoire `cache`) sans compiler :

```
jeekyll clean
```

# jest

Une plateforme de test JavaScript sans configuration.

Plus d'informations : <https://jestjs.io>.

- Exécuter tous les tests disponibles :

```
jest
```

- Exécuter les suites de test de fichiers donnés :

```
jest {{chemin/vers/fichier1 chemin/vers/fichier2 ...}}
```

- Exécuter les suites de test pour des fichiers, dans le répertoire courant et ses sous-répertoires, dont le chemin correspond à l'expression régulière indiquée :

```
jest {{expression_régulière}} {{expression_régulière}}
```

- Exécuter les tests dont les noms correspondent aux expressions régulières indiquées :

```
jest --testNamePattern {{nom_test}}
```

- Exécuter les suites de test associées à un fichier source donné :

```
jest --findRelatedTests {{chemin/vers/fichier_source.js}}
```

- Exécuter les suites de test associées à tous les fichiers non commités :

```
jest --onlyChanged
```

- Surveiller les changements sur les fichiers et ré-exécuter les tests associés :

```
jest --watch
```

- Montrer l'aide :

```
jest --help
```

# john

Outil de cassage de mot de passe.

Plus d'informations : <https://www.openwall.com/john/>.

- Craque les hashes de mots de passe :

```
john {{chemin/vers/hashs.txt}}
```

- Affiche les mots de passe cassés :

```
john --show {{chemin/vers/hashs.txt}}
```

- Affiche les mots de passe cassés des utilisateurs par identifiant d'utilisateur à partir de plusieurs fichiers :

```
john --show --users={{ids_utilisateurs}} {{chemin/vers/hashs1.txt chemin/vers/autres/hashs2.txt ...}}
```

- Craque des hashes de mots de passe, à l'aide d'une liste de mots personnalisée :

```
john --wordlist={{chemin/vers/liste_de_mots.txt}} {{chemin/vers/hashs.txt}}
```

- Liste des formats de hachage disponibles :

```
john --list=formats
```

- Craque les hashes de mots de passe, en utilisant un format de hash spécifique :

```
john --format={{md5crypt}} {{chemin/vers/hashs.txt}}
```

- Craque les hashes de mots de passe, en activant les règles d'altération de mots :

```
john --rules {{chemin/vers/hashs.txt}}
```

- Restaure une session de craquage interrompue à partir d'un fichier d'état, par exemple `mon_cassage.rec` :

```
john --restore={{chemin/vers/mon_cassage.rec}}
```

# jq

Un processeur JSON en ligne de commande qui utilise un langage dédié (DSL).

Plus d'informations : <https://jqlang.github.io/jq/manual/>.

- Exécute une expression spécifique (affiche une sortie JSON coloré et formaté) :

```
{{cat chemin/vers/fichier.json}} | jq '.'
```

- Exécute un script spécifique :

```
{{cat chemin/vers/fichier.json}} | jq --from-file {{chemin/vers/script.jq}}
```

- Transmet des arguments spécifiques :

```
{{cat chemin/vers/fichier.json}} | jq {{--arg "nom1" "valeur1" --arg "nom2" "valeur2" ...}} '{{. + $ARGS.named}}'
```

- Imprime des clés spécifiques :

```
{{cat chemin/vers/fichier.json}} | jq '{{.clé1, .clé2, ...}}'
```

- Imprime des éléments spécifiques du tableau :

```
{{cat chemin/vers/fichier.json}} | jq '{{.[index1], .[index2], ...}}'
```

- Imprime tous les éléments du tableau/les clés de l'objet :

```
{{cat chemin/vers/fichier.json}} | jq '.[[]]'
```

- Ajoute/supprime des clés spécifiques :

```
{{cat chemin/vers/fichier.json}} | jq '. {{+|-}} {{{"clé1": "valeur1", "clé2": "valeur2", ...}}}'
```



# kafkacat

Cette commande est un alias de **kcat**.

- Voir la documentation de la commande originale :

`tldr kcat`

# kill

Envoie un signal à un processus, généralement pour l'interrompre.

Tous les signaux sauf SIGKILL et SIGSTOP peuvent être interceptés par le processus pour pouvoir se terminer proprement.

Plus d'informations : <https://manned.org/kill.1posix>.

- Termine un processus avec le signal SIGTERM (terminaison) par défaut :

```
kill {{identifiant_processus}}
```

- Liste les noms des signaux disponibles (à utiliser sans leurs préfixes SIG) :

```
kill -l
```

- Termine une tâche de fond :

```
kill %{{identifiant_tâche}}
```

- Termine un processus avec le signal SIGHUP ("raccrocher"). Beaucoup de daemons se rafraîchissent au lieu de terminer :

```
kill -{{1|HUP}} {{identifiant_processus}}
```

- Termine un processus avec le signal SIGINT ("interruption"). Généralement initié par l'utilisateur appuyant sur Ctrl + C :

```
kill -{{2|INT}} {{identifiant_processus}}
```

- Demande au système d'exploitation de mettre fin à un processus immédiatement (sans possibilité d'intercepter le signal) :

```
kill -{{9|KILL}} {{identifiant_processus}}
```

- Demande au système d'exploitation de suspendre un processus jusqu'à recevoir le signal SIGCONT ("continue") :

```
kill -{{17|STOP}} {{identifiant_processus}}
```

- Envoie le signal SIGUSR1 à tous les processus dans le groupe avec l'identifiant indiqué :

```
kill -{{SIGUSR1}} -{{identifiant_groupe}}
```

# kosmorro

Calcule les éphémérides et les événements pour une date donnée, à un emplacement donné sur Terre.

Plus d'informations : <https://kosmorro.space>.

- Obtenir les éphémérides pour Paris (France) :

```
kosmorro --latitude={{48.7996}} --longitude={{2.3511}}
```

- Obtenir les éphémérides pour Paris (France), sur le fuseau horaire UTC+2 :

```
kosmorro --latitude={{48.7996}} --longitude={{2.3511}} --  
timezone={{2}}
```

- Obtenir les éphémérides du 9 juin 2020 pour Paris (France) :

```
kosmorro --latitude={{48.7996}} --longitude={{2.3511}} --  
date={{2020-06-09}}
```

- Générer un fichier PDF (TeXLive doit être installé) :

```
kosmorro --format={{pdf}} --output={{chemin/vers/le/  
fichier.pdf}}
```

# ksh

Korn SHell, un interpréteur de ligne de commande compatible avec Bash.

Voir aussi **histexpand** pour l'expansion de l'historique.

Plus d'informations : <http://kornshell.com>.

- Démarre une session shell interactive :

```
ksh
```

- Exécute une commande, puis termine la session :

```
ksh -c "{{commande}}"
```

- Exécute un script :

```
ksh {{chemin/vers/le/script.ksh}}
```

- Exécute un script en affichant chaque commande avant de l'exécuter :

```
ksh -x {{chemin/vers/le/script.ksh}}
```

# llvm-ar

Cette commande est un alias de **ar**.

- Voir la documentation de la commande originale :

```
tldr ar
```

# llvm-g++

Cette commande est un alias de **clang++**.

- Voir la documentation de la commande originale :

```
tldr clang++
```

# llvm-gcc

Cette commande est un alias de **clang**.

- Voir la documentation de la commande originale :

`tldr clang`

# llvm-nm

Cette commande est un alias de **nm**.

- Voir la documentation de la commande originale :

`tldr nm`



# llvm-objdump

Cette commande est un alias de **objdump**.

- Voir la documentation de la commande originale :

`tldr objdump`

# llvm-strings

Cette commande est un alias de **strings**.

- Voir la documentation de la commande originale :

```
tldr strings
```

# ln

Crée des liens vers des fichiers et répertoires.

Plus d'informations : <https://www.gnu.org/software/coreutils/ln>.

- Crée un lien symbolique vers un fichier ou un répertoire :

```
ln -s {{chemin/vers/fichier_ou_repertoire}} {{chemin/vers/lien_symbolique}}
```

- Modifie la cible d'un lien symbolique existant :

```
ln -sf {{chemin/vers/nouveau_fichier}} {{chemin/vers/lien_symbolique}}
```

- Crée un lien dur vers un fichier :

```
ln {{chemin/vers/fichier}} {{chemin/vers/lien_dur}}
```

# ls

Liste le contenu d'un répertoire.

Plus d'informations : <https://www.gnu.org/software/coreutils/ls>.

- Liste les fichiers, un par ligne :

```
ls -l
```

- Liste tous les fichiers, ainsi que les fichiers cachés :

```
ls -a
```

- Liste tous les fichiers, avec les noms de répertoires suivis d'un / :

```
ls -F
```

- Liste tous les fichiers avec un format détaillé (permissions, propriétaire, taille et date de modification) :

```
ls -la
```

- Liste les fichiers avec un format détaillé en utilisant des préfixes d'unités (KiB, MiB, GiB) :

```
ls -lh
```

- Liste les fichiers avec un format détaillé en triant par taille décroissante :

```
ls -lsR
```

- Liste avec un format détaillé tous les fichiers en triant par date de modification (les plus anciennes en premier) :

```
ls -ltr
```

- Liste uniquement les répertoires :

```
ls -d */
```

# lsd

Liste le contenu d'un répertoire.

Commande **ls** de nouvelle génération, écrit en Rust.

Plus d'informations : <https://github.com/Peltoche/lsd>.

- Liste les fichiers et les répertoires, un par ligne :

```
lsd -l
```

- Liste tous les fichiers et les répertoires, ainsi que ceux cachés, du répertoire courant :

```
lsd -a
```

- Liste les fichiers et les répertoires en ajoutant / après le nom des répertoires :

```
lsd -F
```

- Liste tous les fichiers et les répertoires dans le format long (permissions, propriétaire, taille dans un format lisible, et date de modification) :

```
lsd -lha
```

- Liste les fichiers et les répertoires dans le format long, triés par taille (descendant) :

```
lsd -lS
```

- Liste les fichiers et les répertoires dans le format long, triés par date de modification (plus vieux en premier) :

```
lsd -ltr
```

- Liste seulement les répertoires :

```
lsd -d {{*/}}
```

- Liste récursivement tous les répertoires dans un format arborescent :

```
lsd --tree -d
```

# lua

Un langage de programmation puissant, léger, et convenable aux systèmes embarqués.

Plus d'informations : <https://www.lua.org>.

- Démarre une session de commandes interactive Lua :

```
lua
```

- Exécute un script Lua :

```
lua {{nom_du_script.lua}} {{--arguments-facultatifs}}
```

- Exécute une expression Lua :

```
lua -e '{{print( "Hello World" )}}'
```

# lzcat

Cette commande est un alias de **xz**.

Plus d'informations : <https://manned.org/lzcat>.

- Voir la documentation de la commande originale :

```
tldr xz
```

# lzma

Cette commande est un alias de **xz**.

Plus d'informations : <https://manned.org/lzma>.

- Voir la documentation de la commande originale :

```
tldr xz
```



# magick convert

Outil de conversion d'image d'ImageMagick.

Plus d'informations : <https://imagemagick.org/script/convert.php>.

- Convertir une image JPEG en PNG :

```
magick convert {{image.jpg}} {{image.png}}
```

- Redimensionner une image à 50% de ses dimensions d'origine :

```
magick convert {{image.png}} -resize 50% {{image2.png}}
```

- Redimensionner une image en conservant son ratio hauteur/largeur initial pour une taille maximum de 640x480 :

```
magick convert {{image.png}} -resize 640x480 {{image2.png}}
```

- Coller plusieurs images horizontalement :

```
magick convert {{image1.png}} {{image2.png}} {{image3.png}}  
+append {{image123.png}}
```

- Coller plusieurs images verticalement :

```
magick convert {{image1.png}} {{image2.png}} {{image3.png}} -  
append {{image123.png}}
```

- Créer un gif à partir d'une série d'images avec un délai de 100ms entre chaque :

```
magick convert {{image1.png}} {{image2.png}} {{image3.png}} -  
delay {{100}} {{animation.gif}}
```

- Créer une image avec un simple arrière-plan uni :

```
magick convert -size {{800x600}} "xc:  
{{#ff0000}}" {{image.png}}
```

# matlab

Environnement de calcul numérique créé par MathWorks.

Plus d'informations : <https://uk.mathworks.com/help/matlab/>.

- Lance MATLAB sans afficher l'écran de démarrage :

```
matlab -nosplash
```

- Exécute une instruction MATLAB :

```
matlab -r "{{instruction_matlab}}"
```

- Exécute un script MATLAB :

```
matlab -r "run({{chemin/vers/script.m}})"
```

# mc

Midnight Commander, gestionnaire de fichiers à base de console.

La navigation dans les répertoires se fait à l'aide des touches directionnelles ou la souris, ou bien en tapant des commandes dans la console.

Plus d'informations : <https://midnight-commander.org>.

- Démarre Midnight Commander :

```
mc
```

- Démarre Midnight Commander en mode noir et blanc :

```
mc -b
```

# mkdir

Crée un répertoire.

Plus d'informations : <https://www.gnu.org/software/coreutils/mkdir>.

- Crée plusieurs répertoires dans le répertoire actuel :

```
mkdir {{répertoire_1 répertoire_2 ...}}
```

- Crée des répertoires récursivement (utile pour créer des répertoires imbriqués) :

```
mkdir {{-p|--parents}} {{chemin/vers/répertoire}}
```

# mktemp

Crée un fichier ou un répertoire temporaire.

Plus d'informations : <https://man.openbsd.org/mktemp.1>.

- Crée un fichier temporaire vide et affiche son chemin d'accès absolu :

```
mktemp
```

- Utilise un répertoire personnalisé si `$TMPDIR` n'est pas défini (la valeur par défaut dépend de la plateforme, mais est habituellement `/tmp`) :

```
mktemp -p {{/chemin/vers/reptemp}}
```

- Utilise un modèle de chemin personnalisé (les `X` sont remplacés par des caractères alphanumériques aléatoires) :

```
mktemp {{/tmp/exemple.XXXXXXXXXX}}
```

- Utilise un modèle de nom de fichier personnalisé :

```
mktemp -t {{exemple.XXXXXXXXXX}}
```

- Crée un répertoire temporaire vide et affiche son chemin d'accès absolu :

```
mktemp -d
```

# mongo

Client shell pour MongoDB.

Plus d'informations : <https://docs.mongodb.com/manual/reference/program/mongo>.

- Connecte à une base de données (database) sur un hôte (host) distant et un port donné :

```
mongo --host {{hôte}} --port {{port}}  
{{nom_de_la_base_de_données}}
```

- Évalue une expression JavaScript sur une base de données (database) :

```
mongo --eval  
'{{JSON.stringify(db.foo.findOne())}}' {{nom_de_la_base_de_données}}
```

# mscore

Cette commande est un alias de **musescore**.

Plus d'informations : <https://musescore.org/handbook/command-line-options>.

- Voir la documentation de la commande originale :

```
tldr musescore
```

# musl-gcc

Un adaptateur de **gcc** qui définit automatiquement les options pour l'édition de liens avec musl libc.

Toutes les options spécifiées sont passées directement à **gcc**.

Plus d'informations : <https://manned.org/musl-gcc>.

- Voir la documentation de gcc :

`tldr gcc`



# nano

Éditeur de texte simple et convivial. C'est un clone libre et amélioré de Pico.

Plus d'informations : <https://nano-editor.org>.

- Ouvre un fichier :

```
nano {{chemin/vers/fichier}}
```

- Ouvre un fichier en positionnant le curseur à une rangée et colonne donnée :

```
nano +{{ligne}},{{colonne}} {{chemin/vers/fichier}}
```

- Active le défilement fluide :

```
nano -S {{fichier}}
```

- Indente les nouvelles lignes à la même indentation que les lignes précédentes :

```
nano -i {{fichier}}
```

- Avant la modification, sauvegarde le fichier actuel sous le format `nom_du_fichier_actuel~`:

```
nano -B {{fichier}}
```

# netcat

Cette commande est un alias de **nc**.

Plus d'informations : <https://manned.org/nc>.

- Voir la documentation de la commande originale :

```
tldr nc
```

# nm-classic

Cette commande est un alias de **nm**.

- Voir la documentation de la commande originale :

`tldr nm`

# node

Plateforme JavaScript côté serveur.

Plus d'informations : <https://nodejs.org>.

- Exécute un fichier JavaScript :

```
node {{chemin/vers/fichier}}
```

- Démarre un REPL (shell interactif) :

```
node
```

- Évalue du code JavaScript en le passant en argument :

```
node -e "{{code}}"
```

- Évalue et affiche le résultat, très utile pour voir les versions de dépendances node :

```
node -p "{{process.versions}}"
```

- Active l'inspecteur, mets en pause l'exécution jusqu'à ce qu'un debugger soit connecté une fois que le code source est totalement interprété :

```
node --no-lazy --inspect-brk {{chemin/vers/fichier}}
```

# ntl

Cette commande est un alias de **netlify**.

Plus d'informations : <https://cli.netlify.com>.

- Voir la documentation de la commande originale :

```
tldr netlify
```

# passwd

Passwd est un outil de changement de mot de passe utilisateur.

Plus d'informations : <https://manned.org/passwd>.

- Change le mot de passe de l'utilisateur actuel :

```
passwd
```

- Change le mot de passe d'un utilisateur particulier :

```
passwd {{utilisateur}}
```

- Affiche l'état actuel du compte utilisateur :

```
passwd {{-S|--status}}
```

- Supprime le mot de passe de l'utilisateur (supprime l'authentification par mot de passe pour l'utilisateur indiqué) :

```
passwd {{-d|--delete}}
```

# perl

Interpréteur du langage Perl (version 5).

Plus d'informations : <https://www.perl.org>.

- Exécuter le code contenu dans un fichier :

```
perl {{fichier.pl}}
```

- Vérifier la syntaxe sans exécuter le programme :

```
perl -c {{fichier.pl}}
```

- Exécuter une expression Perl :

```
perl -e {{expression}}
```

- Lancer le programme avec le debugger Perl :

```
perl -d {{fichier.pl}}
```

- Itérer sur toutes les lignes d'un fichier, en les modifiant sur place en utilisant une expression de recherche et de remplacement :

```
perl -p -i -e 's/{{recherche}}/{{remplacement}}' {{fichier}}
```

- Lancer une expression de recherche et remplacement sur un fichier, en sauvegardant le fichier original avec une autre extension :

```
perl -p -i'.ancien' -e 's/{{recherche}}/{{remplacement}}/g' {{fichier}}
```

- Lancer une expression de recherche et remplacement sur un fichier, en sauvegardant le résultat dans un autre fichier :

```
perl -p0e 's/{{recherche}}/{{remplacement}}/g' {{fichier_entrée}} > {{fichier_sortie}}
```

- Lancer une expression régulière (RegEx) sur `stdin`, en affichant le premier groupe capturé pour chaque ligne :

```
cat {{fichier_entrée}} | perl -nle 'if (/{{regex}}/) { print "$1"; last;}'
```

# pio init

Cette commande est un alias de **pio project**.

- Voir la documentation de la commande originale :

```
tldr pio project
```



# piodebuggdb

Cette commande est un alias de **pio debug**.

- Voir la documentation de la commande originale :

```
tldr pio debug
```

# pip

Gestionnaire des paquets pour Python.

Certaines commandes comme **pip install** ont leur propre documentation.

Plus d'informations : <https://pip.pypa.io>.

- Installe un paquet :

```
pip install {{paquet}}
```

- Installe une version particulière d'un paquet :

```
pip install {{paquet}}=={{version}}
```

- Installe un paquet dans le répertoire utilisateur au lieu de l'emplacement par défaut système :

```
pip install --user {{paquet}}
```

- Met à jour un paquet :

```
pip install --upgrade {{paquet}}
```

- Désinstalle un paquet :

```
pip uninstall {{paquet}}
```

- Sauvegarde une liste des paquets installés :

```
pip freeze > {{requirements.txt}}
```

- Installe des paquets à partir d'un fichier :

```
pip install --requirement {{requirements.txt}}
```

- Affiche les informations d'un paquet installé :

```
pip show {{paquet}}
```

# pip3

Gestionnaire des paquets pour Python.

Plus d'informations : <https://pip.pypa.io>.

- Recherche un paquet :

```
pip3 search {{paquet}}
```

- Installe un paquet :

```
pip3 install {{paquet}}
```

- Installe une version particulière d'un paquet :

```
pip3 install {{paquet}}=={{version}}
```

- Met à jour un paquet :

```
pip3 install --upgrade {{paquet}}
```

- Désinstalle un paquet :

```
pip3 uninstall {{paquet}}
```

- Sauvegarde une liste des paquets installés :

```
pip3 freeze > {{requirements.txt}}
```

- Installe des paquets à partir d'un fichier :

```
pip3 install --requirement {{requirements.txt}}
```

- Affiche les informations d'un paquet installé :

```
pip3 show {{paquet}}
```

# pipenv

Workflow de développement simple et unifié pour Python.

Gère les paquets et l'environnement virtuel d'un projet.

Plus d'informations : <https://pypi.org/project/pipenv>.

- Crée un nouveau projet :

```
pipenv
```

- Crée un nouveau projet à l'aide de Python 3 :

```
pipenv --three
```

- Installe un paquet :

```
pipenv install {{paquet}}
```

- Installe toutes les dépendances d'un projet :

```
pipenv install
```

- Installe toutes les dépendances d'un projet (y compris les paquets de développement) :

```
pipenv install --dev
```

- Désinstalle un paquet :

```
pipenv uninstall {{paquet}}
```

- Démarre une session de commandes dans l'environnement virtuel :

```
pipenv shell
```

- Génère un `requirements.txt` (une liste de dépendances) pour un projet :

```
pipenv lock --requirements
```

# platformio

Cette commande est un alias de **pio**.

Plus d'informations : <https://docs.platformio.org/en/latest/core/userguide/>.

- Voir la documentation de la commande originale :

```
tldr pio
```

# powershell

Interface en ligne de commande et langage de script spécialement conçu pour l'administration système.

Plus d'informations : [https://learn.microsoft.com/powershell/module/microsoft.powershell.core/about/about\\_pwsh](https://learn.microsoft.com/powershell/module/microsoft.powershell.core/about/about_pwsh).

- Démarre une session Windows PowerShell dans une fenêtre d'invite de commande :

```
powershell
```

- Charge un fichier de console PowerShell spécifique :

```
powershell -PSConsoleFile {{chemin/vers/fichier}}
```

- Démarre une session avec une version spécifiée de PowerShell :

```
powershell -Version {{version}}
```

- Empêche l'interface système de se fermer après avoir exécuté les commandes de démarrage :

```
powershell -NoExit
```

- Décrit le format des données envoyées à PowerShell :

```
powershell -InputFormat {{Texte|XML}}
```

- Détermine comment la sortie de PowerShell est formatée :

```
powershell -OutputFormat {{Texte|XML}}
```

- Affiche l'aide :

```
powershell -Help
```

# ptpython3

Cette commande est un alias de **ptpython**.

- Voir la documentation de la commande originale :

`tldr ptpython`

# pwd

Affiche le nom du répertoire actuel.

Plus d'informations : <https://www.gnu.org/software/coreutils/pwd>.

- Affiche le répertoire actuel :

```
pwd
```

- Affiche le répertoire actuel tout en traduisant les liens symboliques (c.-à-d. afficher le répertoire « physique ») :

```
pwd -P
```



# python3

Cette commande est un alias de **python**.

- Voir la documentation de la commande originale :

`tldr python`

# R

Interpréteur pour le langage R.

Plus d'informations : <https://www.r-project.org>.

- Démarre une session de commande R (REPL) :

```
R
```

- Vérifie la version de R :

```
R --version
```

- Exécute un fichier :

```
R -f {{fichier.R}}
```

# r2

Cette commande est un alias de **radare2**.

- Voir la documentation de la commande originale :

`tldr radare2`

# rcat

Cette commande est un alias de **rc**.

- Voir la documentation de la commande originale :

`tldr rc`

# rsync

Transférer des fichiers vers ou depuis un hôte distant (pas entre deux hôtes distants).

Peut transférer un ou plusieurs fichiers correspondant à un motif.

Plus d'informations : <https://download.samba.org/pub/rsync/rsync.1>.

- Transfère un fichier :

```
rsync {{chemin/vers/origine}} {{chemin/vers/destination}}
```

- Utilise le mode archive (copier récursivement les répertoires, copier les liens symboliques sans résolution et conserver les autorisations, la propriété et les délais de modification) :

```
rsync {{-a|--archive}} {{chemin/vers/origine}} {{chemin/vers/destination}}
```

- Transférer le contenu d'un dossier :

```
rsync {{-r|--recursive}} {{chemin/vers/origine}} {{chemin/vers/destination}}
```

- Transférer le contenu d'un dossier (mais pas le dossier lui-même) :

```
rsync {{-r|--recursive}} {{chemin/vers/origine}}/ {{chemin/vers/destination}}
```

- Utiliser le mode archive, résolvant les liens symboliques et ignorant les fichiers déjà transférés sauf si plus récents :

```
rsync {{-auL|--archive --update --copy-links}} {{chemin/vers/origine}} {{chemin/vers/destination}}
```

- Transférer un fichier vers un hôte distant exécutant `rsyncd` et supprimez les fichiers sur la destination qui n'existent pas sur l'hôte distant :

```
rsync {{-r|--recursive}} --delete rsync://{{hote_distant}}:{{chemin/vers/origine}} {{chemin/vers/destination}}
```

- Transférer un fichier par SSH et afficher l'avancement global du transfert :

```
rsync {{-e|--rsh}} 'ssh -p {{port}}' --info=progress2 {{hote_distant}}:{{chemin/vers/origine}} {{chemin/vers/destination}}
```

# rubocop

Lint les fichiers Ruby.

Plus d'informations : [https://docs.rubocop.org/rubocop/usage/basic\\_usage.html](https://docs.rubocop.org/rubocop/usage/basic_usage.html).

- Vérifie tous les fichiers du répertoire actuel (y compris les sous-répertoires) :

```
rubocop
```

- Vérifie un ou plusieurs fichiers ou répertoires spécifiques :

```
rubocop {{chemin/vers/fichier_ou_dossier1 chemin/vers/
fichier_ou_dossier2 ...}}
```

- Écrit la sortie dans un fichier :

```
rubocop --out {{chemin/vers/fichier}}
```

- Affiche la liste des cops (règles de lint) :

```
rubocop --show-cops
```

- Exclue un cop :

```
rubocop --except {{cop_1 cop_2 ...}}
```

- Exécute uniquement les cops spécifiés :

```
rubocop --only {{cop_1 cop_2 ...}}
```

- Corrige automatiquement les fichiers (expérimental) :

```
rubocop --auto-correct
```

# ruby

Interpréteur du langage de programmation Ruby.

Voir aussi : **gem**, **bundler**, **rake**, **irb**.

Plus d'informations : <https://www.ruby-lang.org>.

- Exécute un script Ruby :

```
ruby {{script.rb}}
```

- Exécute une seule commande Ruby dans la ligne de commande :

```
ruby -e {{commande}}
```

- Vérifie les erreurs de syntaxe d'un script Ruby donné :

```
ruby -c {{script.rb}}
```

- Démarre le serveur HTTP intégré sur le port 8080 dans le répertoire actuel :

```
ruby -run -e httpd
```

- Exécute localement un binaire Ruby sans installer la bibliothèque requise dont il dépend :

```
ruby -I {{chemin/vers/dossier_bibliothèque}} -r  
{{nom_chargement_bibliothèque}} {{chemin/vers/dossier_bin/  
nom_bin}}
```

- Affiche la version de Ruby utilisée :

```
ruby -v
```

# sh

Bourne SHell, l'interpréteur de ligne de commande classique.

Voir aussi **histexpand** pour l'expansion de l'historique.

Plus d'informations : <https://manned.org/sh>.

- Démarre une session shell interactive :

```
sh
```

- Exécute une commande, puis termine la session :

```
sh -c "{{commande}}"
```

- Exécute un script :

```
sh {{chemin/vers/le/script.sh}}
```

- Lit et exécute des commandes depuis l'entrée standard `stdin` :

```
sh -s
```



# shards

Outil de gestion des dépendances pour le langage Crystal.

Plus d'informations : [https://crystal-lang.org/reference/the\\_shards\\_command](https://crystal-lang.org/reference/the_shards_command).

- Créé un fichier `shard.yml` squelette :

```
shards init
```

- Installe les dépendances à partir d'un fichier `shard.yml` :

```
shards install
```

- Met à jour toutes les dépendances :

```
shards update
```

- Liste toutes les dépendances installées :

```
shards list
```

- Liste la version d'une dépendance ayant un chemin donné :

```
shards version {{path/to/dependency_directory}}
```

# ssh-add

Gère les clés SSH enregistrées dans l'agent SSH **ssh-agent**.

S'assurer que **ssh-agent** est en fonctionnement pour enregistrer des clés.

Plus d'informations : <https://man.openbsd.org/ssh-add>.

- Ajoute les clés présentes dans `~/.ssh` (clés par défaut) à l'agent SSH :

```
ssh-add
```

- Ajoute une clé spécifique à l'agent SSH :

```
ssh-add {{chemin/vers/clé_privée}}
```

- Liste les empreintes des clés actuellement enregistrées :

```
ssh-add -l
```

- Supprime une clé de l'agent SSH :

```
ssh-add -d {{chemin/vers/clé_privée}}
```

- Supprime toutes les clés enregistrées de l'agent SSH :

```
ssh-add -D
```

- Ajoute une clé spécifique à l'agent SSH et au trousseau de clés :

```
ssh-add -K {{chemin/vers/clé_privée}}
```

# ssh-agent

Lance un agent SSH.

Un agent SSH permet de stocker des clés SSH déchiffrées, jusqu'à ce qu'elle soient retirées ou que l'agent soit arrêté.

Voir également **ssh-add**, qui permet d'ajouter et de gérer les clés enregistrées par l'agent SSH.

Plus d'informations : <https://man.openbsd.org/ssh-agent>.

- Démarre un agent SSH pour le shell actuel :

```
eval $(ssh-agent)
```

- Arrête l'agent actuellement en fonctionnement :

```
ssh-agent -k
```

# ssh-copy-id

Dépose une clé publique sur une machine distante, dans les clés autorisées **authorized\_keys**.

Plus d'informations : <https://manned.org/ssh-copy-id>.

- Dépose toutes les clés publiques sur la machine distante :

```
ssh-copy-id {{nom_utilisateur}}@{{hote_distant}}
```

- Dépose une clé publique spécifique sur la machine distante :

```
ssh-copy-id -i {{chemin/vers/certificat}} {{nom_utilisateur}}  
@{{hote_distant}}
```

- Dépose une clé publique spécifique sur la machine distante en utilisant un port particulier :

```
ssh-copy-id -i {{chemin/vers/certificat}} -p {{port}}  
{{nom_utilisateur}}@{{hote_distant}}
```

# ssh-keygen

Génère des clés SSH. Utilisées entre autres pour l'authentification ou la connexion sans utiliser de mot de passe.

Plus d'informations : <https://man.openbsd.org/ssh-keygen>.

- Génère une clé de manière interactive :

```
ssh-keygen
```

- Génère une clé ed25519, avec 32 passages de fonction de dérivation de clé et enrigster dans un fichier spécifique :

```
ssh-keygen -t {{ed25519}} -a {{32}} -f {{~/.ssh/fichier}}
```

- Génère une clé RSA de 4096 bits, avec l'adresse électronique en commentaire :

```
ssh-keygen -t {{rsa}} -b {{4096}} -C "{{commentaire|email}}"
```

- Retire les clés d'une machine donnée du fichier `known_hosts` des hôtes connus (utile lorsque un hôte déjà enregistré change de clé) :

```
ssh-keygen -R {{hote_distant}}
```

- Affiche l'empreinte d'une clé sous format d'un hash MD5 :

```
ssh-keygen -l -E {{md5}} -f {{~/.ssh/fichier}}
```

- Change le mot de passe d'une clé :

```
ssh-keygen -p -f {{~/.ssh/fichier}}
```

- Change le format d'une clé (par exemple du format OPENSSH vers PEM), le fichier étant réécrit :

```
ssh-keygen -p -N "" -m {{PEM}} -f {{~/.ssh/cle_privee_OpenSSH}}
```

# ssh-keyscan

Récupère les clés SSH publiques de machines distantes.

Plus d'informations : <https://man.openbsd.org/ssh-keyscan>.

- Récupère toutes les clés d'une machine distante :

```
ssh-keyscan {{hote_distant}}
```

- Récupère toutes les clés d'une machine distante en écoutant sur un port particulier :

```
ssh-keyscan -p {{port}} {{hote_distant}}
```

- Récupère un type particulier de clés d'une machine distante :

```
ssh-keyscan -t {{rsa,dsa,ecdsa,ed25519}} {{hote_distant}}
```

- Met à jour manuellement le fichier `known_hosts` des hôtes connus avec l'empreinte d'une :

```
ssh-keyscan -H {{hote_distant}} >> ~/.ssh/known_hosts
```

# ssh

Secure Shell est un protocole utilisé pour se connecter de façon sécurisée à des systèmes distants.

On peut l'utiliser pour se connecter ou exécuter des commandes sur un serveur distant.

Plus d'informations : <https://man.openbsd.org/ssh>.

- Se connecter à un serveur distant :

```
ssh {{utilisateur}}@{{hote_distant}}
```

- Se connecter à un serveur distant en utilisant une identité spécifique (clé privée) :

```
ssh -i {{chemin/vers/fichier_clef}} {{utilisateur}}  
@{{hote_distant}}
```

- Se connecter à un serveur distant en utilisant un port spécifique :

```
ssh {{utilisateur}}@{{hote_distant}} -p {{2222}}
```

- Exécuter une commande sur un serveur distant :

```
ssh {{utilisateur}}@{{hote_distant}} -t {{commande}}  
{{commande_options}}
```

- Tunnel SSH : Transfert par port dynamique (le SOCKS proxy se trouve sur localhost:1080) :

```
ssh -D {{1080}} {{utilisateur}}@{{hote_distant}}
```

- Tunnel SSH : Transfère un port spécifique (localhost:9999 vers exemple.org:80) en désactivant l'allocation de pseudo-[t]ty et l'exécution de commandes distantes :

```
ssh -L {{9999}}:{{exemple.org}}:{{80}} -N -T {{utilisateur}}  
@{{hote_distant}}
```

- Saut SSH : Se connecter sur un serveur distant à travers une machine de rebond (plusieurs machines de rebond peuvent être définies en les séparant par des virgules) :

```
ssh -J {{utilisateur}}@{{hote_de_rebond}} {{utilisateur}}@{{hote_distant}}
```

- Transfert d'agent : Transfère les informations d'authentification vers la machine distante (voir `man ssh_config` pour les options disponibles) :

```
ssh -A {{utilisateur}}@{{hote_distant}}
```



# sudo

Exécute une commande unique en tant que super-utilisateur (super-user) ou un autre utilisateur.

Plus d'informations : <https://www.sudo.ws/sudo.html>.

- Exécute une commande en tant que super-utilisateur :

```
sudo {{less /var/log/syslog}}
```

- Édite un fichier en tant que super-utilisateur avec votre éditeur par défaut :

```
sudo --edit {{/etc/fstab}}
```

- Exécute une commande en tant qu'un autre utilisateur et/ou groupe :

```
sudo --user={{utilisateur}} --group={{groupe}} {{id -a}}
```

- Répète la dernière commande préfixée de `sudo` (uniquement dans Bash, Zsh, etc.) :

```
sudo !!
```

- Lance le terminal par défaut avec des privilèges de super-utilisateur et exécuter des fichiers à profil spécifique (`.profile`, `.bash_profile`, etc.) :

```
sudo --login
```

- Lance le terminal par défaut avec des privilèges de super-utilisateur sans modifier l'environnement :

```
sudo --shell
```

- Lance le terminal par défaut en tant que l'utilisateur spécifié, en chargeant l'environnement de cet utilisateur et en lisant les fichiers à profil spécifique de cet utilisateur (`.profile`, `.bash_profile`, etc.) :

```
sudo --login --user={{utilisateur}}
```

- Liste les commandes autorisées (et interdites) pour l'utilisateur courant :

```
sudo --list
```

# tar

Utilitaire d'archivage.

Souvent combiné avec une méthode de compression, telle que gzip ou bzip2.

Plus d'informations : <https://www.gnu.org/software/tar>.

- Crée une archive à partir de fichiers :

```
tar cf {{chemin/vers/cible.tar}} {{chemin/vers/fichier1  
chemin/vers/fichier2 ...}}
```

- Crée une archive gzip à partir de fichiers :

```
tar czf {{chemin/vers/cible.tar.gz}} {{chemin/vers/fichier1  
chemin/vers/fichier2 ...}}
```

- Crée une archive gzip à partir d'un répertoire en utilisant son chemin relatif :

```
tar czf {{chemin/vers/cible.tar.gz}} --directory={{chemin/  
vers/répertoire}} .
```

- Extrait une archive (compressée) dans le dossier courant en affichant la liste des fichiers traités :

```
tar xvf {{chemin/vers/source.tar[.gz|.bz2|.xz]}}
```

- Extrait une archive (compressée) dans un répertoire cible :

```
tar xf {{chemin/vers/source.tar[.gz|.bz2|.xz]}} --  
directory={{répertoire}}
```

- Crée une archive compressée, en utilisant le suffixe de l'archive pour déterminer le programme de compression :

```
tar caf {{chemin/vers/cible.tar.xz}} {{chemin/vers/fichier1  
chemin/vers/fichier2 ...}}
```

- Liste les fichiers contenus dans une archive tar :

```
tar tvf {{chemin/vers/source.tar}}
```

- Extrait les fichiers correspondant au motif :

```
tar xf {{source.tar}} --wildcards "{{*.html}}"
```

# tldr

Affiche des pages d'aide simples pour les outils en ligne de commande, depuis le projet **tldr-pages**.

Plus d'informations : <https://github.com/tldr-pages/tldr/blob/main/CLIENT-SPECIFICATION.md#command-line-interface>.

- Affiche la page tldr d'une commande (indice : c'est comme ça que vous êtes arrivé ici !):

```
tldr {{commande}}
```

- Affiche la page tldr de `cd`, en forçant la plateforme par défaut :

```
tldr -p {{android|linux|osx|sunos|windows}} {{cd}}
```

- Affiche la page tldr d'une sous-commande :

```
tldr {{git-checkout}}
```

- Met à jour les pages enregistrées localement (si le client supporte la mise en cache):

```
tldr -u
```

# tldr

Cette commande est un alias de **tldr-lint**.

Plus d'informations : <https://github.com/tldr-pages/tldr-lint>.

- Voir la documentation de la commande originale :

```
tldr tldr-lint
```

# tlmgr arch

Cette commande est un alias de **tlmgr platform**.

Plus d'informations : <https://www.tug.org/texlive/tlmgr.html>.

- Voir la documentation de la commande originale :

```
tldr tlmgr platform
```

# tmux

Multiplexeur de terminaux. Permet plusieurs sessions avec fenêtres, panneaux, et plus encore.

Plus d'informations : <https://github.com/tmux/tmux>.

- Démarrer une nouvelle session :

```
tmux
```

- Démarrer une nouvelle session nommée :

```
tmux new -s {{nom}}
```

- Lister les sessions existantes :

```
tmux ls
```

- S'attacher à la session utilisée la plus récemment :

```
tmux attach
```

- Se détacher de la session actuelle (avec le préfixe Ctrl-B) :

```
<Ctrl>-B d
```

- Détruire une session nommée :

```
tmux kill-session -t {{nom}}
```

# topydo

Une application de liste de choses à faire qui utilise le format todo.txt.

Plus d'informations : <https://github.com/topydo/topydo>.

- Ajouter une tâche à un projet spécifique avec un contexte donné :

```
topydo add "{{todo_message}}" +{{projet_nom}}  
@{{context_nom}}"
```

- Ajouter une tâche à faire avec une date d'échéance de demain et une priorité de A :

```
topydo add "(A) {{todo_message}} due:{{1d}}"
```

- Ajouter une tâche à faire dont la date d'échéance est le vendredi :

```
topydo add "{{todo_message}} due:{{fri}}"
```

- Ajouter une tâche répétitive non stricte (jour + récurrence) :

```
topydo add "water flowers due:{{mon}} rec:{{1w}}"
```

- Ajouter une tâche répétitive stricte (prochaine échéance = date + récurrence) :

```
topydo add "{{todo_message}} due:{{2020-01-01}} rec:{{+1m}}"
```

- Revenir sur la dernière commande topydo exécutée :

```
topydo revert
```

# tput

Accède et modifie les paramètres du terminal.

Plus d'informations : <https://manned.org/tput>.

- Déplace le curseur à un endroit donné sur l'écran :

```
tput cup {{coordonnée_y}} {{coordonnée_x}}
```

- Règle la couleur de l'avant-plan (af) ou de l'arrière-plan (ab) :

```
tput {{setaf|setab}} {{code_de_couleur_ANSI}}
```

- Affiche le numéro de colonnes, de rangées, ou de couleurs :

```
tput {{cols|lines|colors}}
```

- Active la sonnerie du terminal :

```
tput bel
```

- Réinitialise tous les attributs du terminal :

```
tput sgr0
```

- Active ou désactive le retour automatique à la ligne :

```
tput {{smam|rmam}}
```



# tr

Convertisseur de caractères : exécute des remplacements basés sur des caractères uniques et des jeux de caractères.

Plus d'informations : <https://www.gnu.org/software/coreutils/tr>.

- Remplace toutes les occurrences d'un caractère dans un fichier, et affiche le résultat :

```
tr {{caractère_recherché}} {{caractère_replacé}} <
{{fichier}}
```

- Remplace toutes les occurrences d'un caractère dans la sortie d'une autre commande :

```
echo {{texte}} | tr {{caractère_recherché}}
{{caractère_replacé}}
```

- Fais correspondre chaque caractère du premier ensemble au caractère correspondant du second ensemble :

```
tr '{{abcd}}' '{{jkmn}}' < {{fichier}}
```

- Supprime toutes les occurrences de l'ensemble de caractères spécifié dans l'entrée :

```
tr -d '{{caractères_en_entrée}}' < {{fichier}}
```

- Comprime une série de caractères identiques en un seul caractère :

```
tr -s '{{caractères_en_entrée}}' < {{fichier}}
```

- Traduis le contenu d'un fichier en majuscules :

```
tr "[:lower:]" "[:upper:]" < {{fichier}}
```

- Supprime les caractères non imprimables d'un fichier :

```
tr -cd "[:print:]" < {{fichier}}
```

# type

Affiche le type de la commande que le terminal exécuterait.

Plus d'informations : <https://manned.org/type>.

- Affiche le type d'une commande :

```
type {{commande}}
```

- Affiche tous les emplacements contenant l'exécutable spécifié :

```
type -a {{commande}}
```

- Affiche le nom du fichier qui serait exécuté :

```
type -p {{commande}}
```

# unlzma

Cette commande est un alias de **xz**.

Plus d'informations : <https://manned.org/unlzma>.

- Voir la documentation de la commande originale :

```
tldr xz
```

# unxz

Cette commande est un alias de **xz**.

Plus d'informations : <https://manned.org/unxz>.

- Voir la documentation de la commande originale :

```
tldr xz
```

# vegeta

Un utilitaire de ligne de commande et une bibliothèque pour les tests de charge HTTP.

Voir aussi **ab**.

Plus d'informations : <https://github.com/tsenart/vegeta>.

- Lancer une attaque d'une durée de 30 secondes :

```
echo "{{GET https://exemple.com}}" | vegeta attack -  
duration={{30s}}
```

- Lancez une attaque sur un serveur avec un certificat HTTPS auto-signé :

```
echo "{{GET https://exemple.com}}" | vegeta attack -insecure  
-duration={{30s}}
```

- Lancer une attaque avec un taux de 10 demandes par seconde :

```
echo "{{GET https://exemple.com}}" | vegeta attack -  
duration={{30s}} -rate={{10}}
```

- Lancer une attaque et afficher un rapport :

```
echo "{{GET https://exemple.com}}" | vegeta attack -  
duration={{30s}} | vegeta report
```

- Lancer une attaque et reporter les résultats sur un graphique (latence en fonction du temps) :

```
echo "{{GET https://exemple.com}}" | vegeta attack -  
duration={{30s}} | vegeta plot > {{chemin/au/results.html}}
```

- Lancer une attaque contre plusieurs URL à partir d'un fichier :

```
vegeta attack -duration={{30s}} -targets={{requetes.txt}} |  
vegeta report
```

# vi

Cette commande est un alias de **vim**.

- Voir la documentation de la commande originale :

`tldr vim`

# vim

Vim (Vi IMproved), un éditeur de texte en ligne de commandes, fournit plusieurs modes pour différentes manipulations de texte.

Pressez **i** pour passer en mode édition. **<Esc>** revient au mode normal, qui ne permet pas l'insertion de code.

Plus d'informations : <https://www.vim.org>.

- Ouvrir un fichier :

```
vim {{chemin/vers/fichier}}
```

- Ouvrir un fichier à une ligne spécifiée :

```
vim +{{numero_ligne}} {{chemin/vers/fichier}}
```

- Consulter le manuel utilisateur :

```
:help<Entrée>
```

- Sauvegarder et fermer :

```
{{<Esc>ZZ|<Esc>:x<Enter>|<Esc>:wq<Enter>}}
```

- Annuler la dernière opération :

```
<Esc>u
```

- Rechercher un motif dans un fichier (appuyez **n/N** pour aller à la prochaine / précédente occurrence) :

```
/{{motif_recherché}}<Entrée>
```

- Effectuer une substitution par expression régulière dans tout le fichier :

```
:%s/{{motif}}/{{remplacement}}/g<Entrée>
```

- Afficher les numéros de ligne :

```
:set nu<Entrée>
```

# WC

Compte les lignes, les mots ou les octets.

Plus d'informations : <https://www.gnu.org/software/coreutils/wc>.

- Compte les lignes d'un fichier :

```
wc --lines {{chemin/vers/fichier}}
```

- Compte les mots d'un fichier :

```
wc --words {{chemin/vers/fichier}}
```

- Compte les octets d'un fichier :

```
wc --bytes {{chemin/vers/fichier}}
```

- Compte les caractères d'un fichier (en prenant en compte l'ensemble des caractères multi-octets) :

```
wc --chars {{chemin/vers/fichier}}
```

- Compte les lignes, les mots et les caractères depuis l'entrée standard `stdin` :

```
{{find .}} | wc
```

- Compte la longueur en nombre de caractères de la plus grande ligne d'un fichier :

```
wc --max-line-length {{chemin/vers/fichier}}
```



# which

Localise un programme dans le chemin de l'utilisateur.

Plus d'informations : <https://manned.org/which>.

- Fouille la variable d'environnement « PATH » et affiche l'emplacement des programmes exécutables correspondants à la requête :

```
which {{exécutable}}
```

- Affiche tous les exécutables correspondants à la requête, s'il y en a plus qu'un :

```
which -a {{exécutable}}
```

# xzcat

Cette commande est un alias de **xz**.

Plus d'informations : <https://manned.org/xzcat>.

- Voir la documentation de la commande originale :

```
tldr xz
```

# yes

Envoie un message à répétition en sortie console.

Cette commande est souvent utilisée pour éviter de devoir accepter des opérations successives (par exemple des installations via la commande **apt-get**).

Plus d'informations : <https://www.gnu.org/software/coreutils/yes>.

- Envoyer « message » à répétition :

```
yes {{message}}
```

- Envoyer « y » à répétition :

```
yes
```

- Répondre « oui » à toutes les questions posées par la commande **apt-get** :

```
yes | sudo apt-get install {{program}}
```

# Z

Recherche les répertoires les plus utilisés et permet une navigation rapide à l'aide de chaînes de caractères ou d'expressions régulières.

Plus d'informations : <https://github.com/rupa/z>.

- Aller dans un répertoire qui contient "foo" dans son nom :

```
z {{foo}}
```

- Aller dans un répertoire qui contient "foo" et "bar" dans son nom :

```
z {{foo}} {{bar}}
```

- Aller dans le répertoire le mieux classé parmi ceux qui contiennent "foo" dans leurs noms :

```
z -r {{foo}}
```

- Aller dans le répertoire accédé le plus récemment parmi ceux qui contiennent "foo" dans leurs noms :

```
z -t {{foo}}
```

- Lis l'ensemble des répertoires dans la base de données z qui contiennent "foo" dans leurs noms :

```
z -l {{foo}}
```

- Supprime le répertoire courant de la base de données de z :

```
z -x .
```

# zip

Empaquette et compresse (archive) les fichiers en un fichier zip.

Plus d'informations : <https://manned.org/zip>.

- Empaquette et compresse [r]écurivement un répertoire et son contenu :

```
zip -r {{archive.zip}} {{chemin/du/répertoire}}
```

- E[x]clure des fichiers de l'archive :

```
zip -r {{archive.zip}} {{chemin/vers/le/répertoire}} -x  
{{chemin/des/fichiers/exclus}}
```

- Archive un répertoire et son contenu avec le plus haut niveau [9] de compression :

```
zip -r -{{9}} {{archive.zip}} {{chemin/du/répertoire}}
```

- Empaquette et compresse plusieurs répertoires et fichiers :

```
zip -r {{archive.zip}} {{chemin/du/répertoire1 chemin/du/  
répertoire2 chemin/du/fichier}}
```

- Crée une archive chiffrée (l'utilisateur sera sollicité pour saisir le mot de passe) :

```
zip -e -r {{archive.zip}} {{chemin/du/répertoire}}
```

- Ajoute des fichiers à une archive existante :

```
zip {{archive.zip}} {{chemin/du/fichier}}
```

- Supprime des fichiers d'une archive existante :

```
zip -d {{archive.zip}} "{{foo/*.tmp}}"
```

- Archive un répertoire et son contenu en plusieurs fichiers zip [s]cindés (ex : des fichiers de 3 Go) :

```
zip -r -s {{3g}} {{archive.zip}} {{chemin/du/répertoire}}
```

# zless

Lire des fichiers compressés.

Plus d'informations : <https://manned.org/zless>.

- Parcourir une archive compressée avec `less` :

```
zless {{fichier.txt.gz}}
```

# zola

Un générateur de site statique à partir d'un unique binaire sans dépendance.

Plus d'informations : <https://www.getzola.org/documentation/getting-started/cli-usage/>.

- Créer la structure du répertoire utilisé par Zola dans un répertoire donné :

```
zola init {{mon_site}}
```

- Construit la totalité du site dans le répertoire `public` (si le répertoire existe, il est supprimé) :

```
zola build
```

- Construit la totalité du site dans un répertoire différent :

```
zola build --output-dir {{chemin/du/répertoire_de_sortie/}}
```

- Construit et met à disposition le site à partir d'un serveur local (l'adresse par défaut est `127.0.0.1:1111`) :

```
zola serve
```

- Construit l'ensemble des pages comme la commande `build`, sans écrire le résultat sur le disque :

```
zola check
```

# zopflipng

Utilitaire de compression d'images PNG.

Plus d'informations : <https://github.com/google/zopfli>.

- Optimise une image PNG :

```
zopflipng {{entrée.png}} {{sortie.png}}
```

- Optimise plusieurs images PNG et sauvegarde avec préfixe donné :

```
zopflipng --prefix={{prefix}} {{image1.png}} {{image2.png}}  
{{image3.png}}
```



# zoxide

Garde une trace des répertoires les plus utilisés.

Utilise un algorithme de classement pour identifier le meilleur résultat.

Plus d'informations : <https://github.com/ajeetsouza/zoxide>.

- Aller au répertoire avec le meilleur classement qui contient "foo" dans son nom :

```
zoxide query {{foo}}
```

- Aller au répertoire avec le meilleur classement qui contient "foo" et "bar" dans son nom :

```
zoxide query {{foo}} {{bar}}
```

- Démarre une recherche de répertoire interactive (nécessite fzf) :

```
zoxide query --interactive
```

- Ajoute un répertoire ou incrémente son classement :

```
zoxide add {{chemin/du/répertoire}}
```

- Supprime un répertoire interactive de la base de données de zoxide :

```
zoxide remove {{chemin/du/répertoire}} --interactive
```

- Génère la configuration du shell pour la mise en place des alias de commandes (z, za, zi, zq, zr) :

```
zoxide init {{bash|fish|zsh}}
```

# zsh

Z SHell, un interpréteur de ligne de commande compatible avec Bash.

Voir aussi **histexpand** pour l'expansion de l'historique.

Plus d'informations : <https://www.zsh.org>.

- Démarre une session shell interactive :

```
zsh
```

- Exécute une commande, puis termine la session :

```
zsh -c "{{commande}}"
```

- Exécute un script :

```
zsh {{chemin/vers/script.zsh}}
```

- Exécute un script en affichant chaque commande avant de l'exécuter :

```
zsh --xtrace {{chemin/vers/script.zsh}}
```

- Démarre une session shell interactive en mode verbeux, qui affiche chaque commande avant de l'exécuter :

```
zsh --verbose
```

- Exécute une commande spécifique dans Zsh sans motifs génériques d'expansion des noms de fichier :

```
noglob "{{commande}}"
```

Linux

# a2disconf

Désactive un fichier de configuration sur une distribution Debian.

Plus d'informations : <https://manned.org/a2disconf.8>.

- Désactive un fichier de configuration :

```
sudo a2disconf {{fichier_de_configuration}}
```

- N'affiche aucun message (mode silencieux) :

```
sudo a2disconf --quiet {{fichier_de_configuration}}
```

# a2dismod

Désactive un module Apache sur une distribution Debian.

Plus d'informations : <https://manned.org/a2dismod.8>.

- Désactive un module :

```
sudo a2dismod {{module}}
```

- N'affiche aucun message (mode silencieux) :

```
sudo a2dismod --quiet {{module}}
```

# a2dissite

Désactive un hôte virtuel Apache sur une distribution Debian.

Plus d'informations : <https://manned.org/a2dissite.8>.

- Désactive un hôte virtuel :

```
sudo a2dissite {{virtual_host}}
```

- N'affiche aucun message (mode silencieux) :

```
sudo a2dissite --quiet {{virtual_host}}
```

# a2enconf

Active un fichier de configuration sur une distribution Debian.

Plus d'informations : <https://manned.org/a2enconf.8>.

- Active un fichier de configuration :

```
sudo a2enconf {{fichier_de_configuration}}
```

- N'affiche aucun message (mode silencieux) :

```
sudo a2enconf --quiet {{fichier_de_configuration}}
```

# a2enmod

Active un module Apache sur une distribution Debian.

Plus d'informations : <https://manned.org/a2enmod.8>.

- Active un module :

```
sudo a2enmod {{module}}
```

- N'affiche aucun message (mode silencieux) :

```
sudo a2enmod --quiet {{module}}
```



# a2ensite

Active un hôte virtuel Apache sur des systèmes d'exploitation (SE) basés sur Debian.

Plus d'informations : <https://manned.org/a2ensite.8>.

- Active un hôte virtuel :

```
sudo a2ensite {{hote_virtuel}}
```

- N'affiche aucun message (mode silencieux) :

```
sudo a2ensite --quiet {{hote_virtuel}}
```

# a2query

Retourne la configuration d'exécution d'Apache sur une distribution Debian.

Plus d'informations : <https://manned.org/a2query>.

- Liste les [m]odules Apache actifs :

```
sudo a2query -m
```

- Vérifie si un module spécifique est installé :

```
sudo a2query -m {{nom_module}}
```

- Liste les hôtes virtuels actifs :

```
sudo a2query -s
```

- Affiche le [M]odule de traitement multiple actif :

```
sudo a2query -M
```

- Affiche la [v]ersion d'Apache :

```
sudo a2query -v
```

# abbr

Gère les abréviations pour le shell fish.

Les mots définis par l'utilisateur sont remplacés par des phrases plus longues après leur saisie.

Plus d'informations : <https://fishshell.com/docs/current/cmds/abbr.html>.

- Ajoute une nouvelle abréviation :

```
abbr --add {{nom_abrégé}} {{commande}}  
{{arguments_de_la_commande}}
```

- Renomme une abréviation existante :

```
abbr --rename {{ancien_nom}} {{nouveau_nom}}
```

- Supprime une abréviation existante :

```
abbr --erase {{nom_abrégé}}
```

- Importe les abréviations définies sur un autre hôte via SSH :

```
ssh {{nom_de_l_hôte}} abbr --show | source
```

# ac

Affiche les statistiques concernant la durée de connexion des utilisateurs.

Plus d'informations : <https://www.gnu.org/software/acct/manual/accounting.html#ac>.

- Affiche pendant combien de temps l'utilisateur actuel a été connecté, en heures :

```
ac
```

- Affiche pendant combien de temps les utilisateurs ont été connectés, en heures :

```
ac --individual-totals
```

- Affiche pendant combien de temps un utilisateur particulier a été connecté, en heures :

```
ac --individual-totals {{nom_d_utilisateur}}
```

- Affiche pendant combien de temps un utilisateur particulier a été connecté, en heures par jour (avec le total) :

```
ac --daily-totals --individual-totals {{nom_d_utilisateur}}
```

- Affiche des détails supplémentaires :

```
ac --compatibility
```

# acpi

Affiche l'état de la batterie ou des renseignements sur la température.

Plus d'informations : <https://sourceforge.net/projects/acpiclient/files/acpiclient/>.

- Affiche les informations sur la batterie :

```
acpi
```

- Affiche les informations sur la température :

```
acpi -t
```

- Afficher les informations sur le dispositif de refroidissement :

```
acpi -c
```

- Afficher les informations sur le dispositif de refroidissement en Fahrenheit :

```
acpi -tf
```

- Afficher toutes les informations :

```
acpi -V
```

- Extraye les informations depuis `/proc` au lieu de `/sys` :

```
acpi -p
```

# adduser

Outil d'ajout d'utilisateurs.

Plus d'informations : <https://manned.org/adduser>.

- Crée un nouvel utilisateur avec un répertoire personnel générique et demande interactivement un mot de passe :

```
adduser {{nom_d_utilisateur}}
```

- Crée un nouvel utilisateur sans répertoire personnel :

```
adduser --no-create-home {{nom_d_utilisateur}}
```

- Crée un nouvel utilisateur avec un répertoire personnel correspondant au dossier spécifié :

```
adduser --home {{chemin/vers/dossier}} {{nom_d_utilisateur}}
```

- Crée un nouvel utilisateur avec l'interpréteur de commandes spécifié comme interpréteur de commandes de connexion :

```
adduser --shell {{chemin/vers/shell}} {{nom_d_utilisateur}}
```

- Crée un nouvel utilisateur appartenant au groupe donné :

```
adduser --ingroup {{groupe}} {{nom_d_utilisateur}}
```

# alien

Convertit différents paquets d'installation vers d'autres formats.

Plus d'informations : <https://manned.org/alien>.

- Convertit un fichier d'installation spécifique vers le format Debian (extension .deb) :

```
sudo alien --to-deb {{chemin/vers/fichier}}
```

- Convertit un fichier d'installation spécifique vers le format Red Hat (extension .rpm) :

```
sudo alien --to-rpm {{chemin/vers/fichier}}
```

- Convertit un fichier d'installation spécifique en un fichier d'installation Slackware (extension .tgz) :

```
sudo alien --to-tgz {{chemin/vers/fichier}}
```

- Convertit un fichier d'installation spécifique vers le format Debian et l'installe sur le système :

```
sudo alien --to-deb --install {{chemin/vers/fichier}}
```

# alternatives

Cette commande est un alias de **update-alternatives**.

Plus d'informations : <https://manned.org/alternatives>.

- Voir la documentation de la commande originale :

```
tldr update-alternatives
```



# apache2ctl

L'outil d'Interface en Lignes de Commandes (ILC) pour administrer le serveur web HTTP Apache.

Cette commande est disponible sur une distribution Debian. Pour les distributions basées Red Hat, voir **httpd**.

Plus d'informations : <https://manned.org/apache2ctl.8>.

- Démarre le démon Apache. Envoie un message s'il est déjà actif :

```
sudo apache2ctl start
```

- Arrête le démon Apache :

```
sudo apache2ctl stop
```

- Re-démarre le démon Apache :

```
sudo apache2ctl restart
```

- Teste la syntaxe du fichier de configuration :

```
sudo apache2ctl -t
```

- Liste les modules chargés :

```
sudo apache2ctl -M
```

# apk

Gestionnaire de paquet d'Alpine Linux.

Plus d'informations : <https://manned.org/apk>.

- Mets à jour les indexes de tous les dépôts distants :

```
apk update
```

- Installe un nouveau paquet :

```
apk add {{paquet}}
```

- Désinstalle un paquet :

```
apk del {{paquet}}
```

- Essaye de réparer un paquet ou de mettre à jour un paquet sans ses dépendances :

```
apk fix {{paquet}}
```

- Recherche des paquets à partir d'un mot-clé :

```
apk search {{mot_cle}}
```

- Obtiens des information à propos d'un paquet précis :

```
apk info {{paquet}}
```

# apt-add-repository

Gère la définition des dépôts APT.

Plus d'informations : <https://manned.org/apt-add-repository.1>.

- Ajout d'un nouveau dépôt :

```
apt-add-repository {{repository_spec}}
```

- Suppression d'un dépôt :

```
apt-add-repository --remove {{repository_spec}}
```

- Mise à jour du cache des paquets après ajout d'un dépôt :

```
apt-add-repository --update {{repository_spec}}
```

- Activation pour les paquets source :

```
apt-add-repository --enable-source {{repository_spec}}
```

# apt-cache

Outil de recherche de paquets Debian et Ubuntu.

Plus d'informations : <https://manned.org/apt-cache.8>.

- Recherche un paquet dans vos sources actuelles :

```
apt-cache search {{query}}
```

- Affiche des informations sur un paquet :

```
apt-cache show {{package}}
```

- Indique si un paquet est installé et à jour :

```
apt-cache policy {{package}}
```

- Affiche les dépendances d'un paquet :

```
apt-cache depends {{package}}
```

- Affiche les paquets qui dépendent d'un paquet particulier :

```
apt-cache rdepends {{package}}
```

# apt-file

Recherche de fichiers dans les paquets APT, y compris ceux qui ne sont pas encore installés.

Plus d'informations : <https://manned.org/apt-file.1>.

- Mise à jour la base de données des métadonnées :

```
sudo apt update
```

- Recherche des paquets qui contiennent le fichier ou le chemin d'accès spécifié :

```
apt-file search {{part/of/filename}}
```

- Énumère le contenu d'un paquet spécifique :

```
apt-file list {{package_name}}
```

# apt-get

Utilitaire de gestion des paquets Debian et Ubuntu.

Recherche des paquets en utilisant **apt-cache**.

Plus d'informations : <https://manned.org/apt-get.8>.

- Mise à jour de la liste des paquets et des versions disponibles (il est recommandé de l'exécuter avant les autres commandes **apt-get**) :

```
apt-get update
```

- Installation d'un paquet, ou mise à jour avec la dernière version disponible :

```
apt-get install {{package}}
```

- Suppression d'un paquet :

```
apt-get remove {{package}}
```

- Suppression d'un paquet et de ses fichiers de configuration :

```
apt-get purge {{package}}
```

- Mise à jour de tous les paquets installés vers les dernières versions disponibles :

```
apt-get upgrade
```

- Nettoyage du dépôt local - supprime les fichiers de paquets (.deb) des téléchargements interrompus qui ne peuvent plus être téléchargés :

```
apt-get autoclean
```

- Suppression de tous les paquets qui ne sont plus nécessaires :

```
apt-get autoremove
```

- Mise à jour des paquets installés (comme la commande **upgrade**), mais avec suppression des paquets obsolètes et installation des paquets supplémentaires pour répondre aux nouvelles dépendances :

```
apt-get dist-upgrade
```

# apt-mark

Utilitaire permettant de modifier l'état des paquets installés.

Plus d'informations : <https://manned.org/apt-mark.8>.

- Marquer un paquet comme étant automatiquement installé :

```
sudo apt-mark auto {{package_name}}
```

- Maintenir un paquet à sa version actuelle et empêcher les mises à jour :

```
sudo apt-mark hold {{package_name}}
```

- Permettre une nouvelle mise à jour d'un paquet :

```
sudo apt-mark unhold {{package_name}}
```

- Afficher les paquets installés manuellement :

```
apt-mark showmanual
```

- Afficher les paquets détenus qui ne sont pas mis à jour :

```
apt-mark showhold
```

# apt

Utilitaire de gestion des paquets pour les distributions basées sur Debian.

Remplacement recommandé pour **apt-get** lorsqu'il est utilisé de manière interactive dans les versions 16.04 et ultérieures d'Ubuntu.

Plus d'informations : <https://manned.org/apt.8>.

- Mettre à jour la liste des paquets et des versions disponibles (il est recommandé de l'exécuter avant les autres commandes **apt**) :  

```
sudo apt update
```
- Recherche d'un paquet donné :  

```
apt search {{package}}
```
- Afficher les informations pour un paquet :  

```
apt show {{package}}
```
- Installer un paquet, ou le mettre à jour avec la dernière version disponible :  

```
sudo apt install {{package}}
```
- Supprimer un paquet (utiliser **purge** à la place supprime également ses fichiers de configuration) :  

```
sudo apt remove {{package}}
```
- Mettre à jour tous les paquets installés vers les dernières versions disponibles :  

```
sudo apt upgrade
```
- Lister tous les paquets :  

```
apt list
```
- Lister les paquets installés :  

```
apt list --installed
```



# ark

Outil d'archive de KDE.

Plus d'informations : <https://docs.kde.org/stable5/en/ark/ark/>.

- Extrait une archive dans le répertoire courant :

```
ark --batch {{chemin/vers/archive}}
```

- Change le répertoire d'extraction :

```
ark --batch --destination {{chemin/vers/dossier}} {{chemin/vers/archive}}
```

- Crée une archive si elle n'existe pas et y ajouter des fichiers :

```
ark --add-to {{chemin/vers/archive}} {{chemin/vers/fichier1  
chemin/vers/fichier2 ...}}
```

# as

Assembleur GNU portable. Principalement destiné pour assembler la sortie de **gcc** pour être utilisé par **ld**.

Plus d'informations : <https://manned.org/as>.

- Assemble un fichier, en écrivant la sortie dans le fichier `a.out` :

```
as {{chemin/vers/fichier.s}}
```

- Assemble la sortie vers un fichier donné :

```
as {{chemin/vers/fichier.s}} -o {{chemin/vers/sortie.o}}
```

- Génère la sortie plus vite en évitant le preprocess des espaces et des commentaires (doit seulement être utilisé sur des compilateurs sûrs) :

```
as -f {{chemin/vers/fichier.s}}
```

- Inclut un chemin donné à la liste des répertoires dans lesquels chercher les fichiers spécifiés dans les directives `.include` :

```
as -I {{chemin/vers/le/répertoire}} {{chemin/vers/fichier.s}}
```

# ascii

Affiche les alias de caractères ASCII.

Plus d'informations : <http://www.catb.org/~esr/ascii/>.

- Affiche les alias ASCII d'un caractère :

```
ascii {{c}}
```

- Affiche les alias ASCII dans un format court, adapté pour les scripts :

```
ascii -t {{c}}
```

- Affiche les alias ASCII de plusieurs caractères :

```
ascii -s {{tldr}}
```

- Affiche la table ASCII en décimal :

```
ascii -d
```

- Affiche la table ASCII en hexadécimal :

```
ascii -x
```

- Affiche la table ASCII en octal :

```
ascii -o
```

- Affiche la table ASCII en binaire :

```
ascii -b
```

- Affiche le résumé des options et une table ASCII complète :

```
ascii
```

# asciiart

Convertit des images en ASCII.

Plus d'informations : <https://github.com/nodanaonlyzuul/asciiart>.

- Lit une image depuis un fichier et l'affiche en ASCII :

```
asciiart {{chemin/vers/image.jpg}}
```

- Lit une image depuis une URL et l'affiche en ASCII :

```
asciiart {{www.example.com/image.jpg}}
```

- Choisit la largeur de sortie (valeur par défaut : 100) :

```
asciiart --width {{50}} {{chemin/vers/image.jpg}}
```

- Colorise la sortie ASCII :

```
asciiart --color {{chemin/vers/image.jpg}}
```

- Choisit le format de sortie (format par défaut : textuel) :

```
asciiart --format {{text|html}} {{chemin/vers/image.jpg}}
```

- Inverse la table de caractères :

```
asciiart --invert-chars {{chemin/vers/image.jpg}}
```

# at

Exécute des commandes à des temps déterminés.

Plus d'informations : <https://manned.org/at.1>.

- Ouvre une invite `at` afin de créer un nouvel ensemble de commandes programmées, presser `Ctrl + D` pour sauvegarder et quitter :

```
at {{hh:mm}}
```

- Exécute les commandes et envoie les résultats par mail en utilisant un programme de messagerie local comme Sendmail :

```
at {{hh:mm}} -m
```

- Exécute un script à un temps donné :

```
at {{hh:mm}} -f {{chemin/vers/le/script}}
```

- Affiche une notification système à 23 heures le 18 Février :

```
echo "notify-send '{{Debout !}}'" | at {{11pm}} {{Feb 18}}
```

# batcat

Cette commande est un alias de **bat**.

Plus d'informations : <https://github.com/sharkdp/bat>.

- Voir la documentation de la commande originale :

```
tldr bat
```

# blight

Utilitaire pour changer la luminosité de l'affichage.

Plus d'informations : <https://github.com/gutjuri/blight>.

- Fixe la luminosité de l'affichage à 50% :

```
blight set {{50}} -r
```

- Affiche la luminosité actuelle de l'affichage :

```
blight show
```

- Affiche la luminosité maximale de l'affichage :

```
blight max
```

- Augmente la luminosité de l'affichage en pourcentage :

```
blight inc {{nombre}} -r
```

- Réduit la luminosité de l'affichage en unités internes :

```
blight dec {{nombre}}
```

# btrfs check

Vérifier l'état, ou réparer un système de fichiers de type btrfs.

Plus d'informations : <https://btrfs.readthedocs.io/en/latest/btrfs-check.html>.

- Vérifier l'état d'un système de fichiers btrfs :

```
sudo btrfs check {{chemin/vers/partition}}
```

- Vérifier l'état et réparer d'un système de fichiers btrfs (dangereux) :

```
sudo btrfs check --repair {{chemin/vers/partition}}
```

- Afficher la progression de vérification en cours :

```
sudo btrfs check --progress {{chemin/vers/partition}}
```

- Vérifier la somme de contrôle de chaque bloc de données (si le système de fichiers a été correctement vérifié) :

```
sudo btrfs check --check-data-csum {{chemin/vers/partition}}
```

- Utiliser le n-ième super-bloc (n peut-être 0, 1 ou 2) :

```
sudo btrfs check --super {{n}} {{chemin/vers/partition}}
```

- Reconstruire l'arbre des sommes de contrôle (checksum tree) :

```
sudo btrfs check --repair --init-csum-tree {{chemin/vers/partition}}
```

- Reconstruire l'arbre des domaines (extent tree) :

```
sudo btrfs check --repair --init-extent-tree {{chemin/vers/partition}}
```



# btrfs device

Gestion des partitions dans un système de fichiers BTRFS.

Plus d'informations : <https://btrfs.readthedocs.io/en/latest/btrfs-device.html>.

- Ajouter un ou plusieurs périphériques à un système de fichiers btrfs :

```
sudo btrfs device add {{chemin/vers/block_device1}}  
[{{chemin/vers/block_device2}}] {{chemin/vers/  
systeme_de_fichiers_btrfs}}
```

- Retirer un périphérique d'un système de fichiers btrfs :

```
sudo btrfs device remove {{chemin/vers/peripherique|  
identifiant_peripherique}} [{{...}}]
```

- Afficher les statistiques d'erreurs :

```
sudo btrfs device stats {{chemin/vers/  
systeme_de_fichiers_btrfs}}
```

- Scanner tous les disques et informer le noyau de tous les systèmes de fichiers btrfs détectés :

```
sudo btrfs device scan --all-devices
```

- Afficher les statistiques détaillées d'allocation par disque :

```
sudo btrfs device usage {{chemin/vers/  
systeme_de_fichiers_btrfs}}
```

# btrfs filesystem

Gérer les systèmes de fichiers btrfs.

Plus d'informations : <https://btrfs.readthedocs.io/en/latest/btrfs-filesystem.html>.

- Afficher l'utilisation du système de fichiers (affiche les informations détaillées si exécuté en tant que `root`) :

```
btrfs filesystem usage {{chemin/vers/point_de_montage_btrfs}}
```

- Afficher l'utilisation individuellement pour chaque périphérique :

```
sudo btrfs filesystem show {{chemin/vers/point_de_montage_btrfs}}
```

- Défragmenter un fichier unique sur un système de fichiers btrfs (à éviter lorsqu'un agent de dé-duplication est en cours d'exécution) :

```
sudo btrfs filesystem defragment -v {{chemin/vers/fichier}}
```

- Défragmenter récursivement un dossier (ne franchit pas la limite de sous-volume) :

```
sudo btrfs filesystem defragment -v -r {{chemin/vers/dossier}}
```

- Force la resynchronisation des blocs de données non écrits sur le ou les disques :

```
sudo btrfs filesystem sync {{chemin/vers/point_de_montage_btrfs}}
```

- Afficher un sommaire d'utilisation des disques pour les fichiers dans un dossier, récursivement :

```
sudo btrfs filesystem du --summarize {{chemin/vers/dossier}}
```

# btrfs inspect-internal

Recherche des informations internes concernant un système de fichier btrfs.

Plus d'informations : <https://btrfs.readthedocs.io/en/latest/btrfs-inspect-internal.html>.

- Afficher les informations du superbloc :

```
sudo btrfs inspect-internal dump-super {{chemin/vers/  
partition}}
```

- Afficher les informations sur les superblocs et toutes ses copies :

```
sudo btrfs inspect-internal dump-super --all {{chemin/vers/  
partition}}
```

- Afficher les meta-informations du système de fichiers :

```
sudo btrfs inspect-internal dump-tree {{chemin/vers/  
partition}}
```

- Afficher la liste des fichiers dans le n-ième inode :

```
sudo btrfs inspect-internal inode-resolve {{n}} {{chemin/  
vers/point_de_montage_btrfs}}
```

- Afficher la liste des fichiers à une adresse logique donnée :

```
sudo btrfs inspect-internal logical-resolve  
{{adresse_logique}} {{chemin/vers/point_de_montage_btrfs}}
```

- Afficher les statistiques concernant les arbres de racines, de domaines (extent), de sommes de contrôle (csum) et de système de fichiers :

```
sudo btrfs inspect-internal tree-stats {{chemin/vers/  
partition}}
```

# btrfs rescue

Essayer de récupérer un système de fichiers btrfs endommagé.

Plus d'informations : <https://btrfs.readthedocs.io/en/latest/btrfs-rescue.html>.

- Reconstruire les méta-données du système de fichiers (très lent) :

```
sudo btrfs rescue chunk-recover {{chemin/vers/partition}}
```

- Corriger les problèmes d'alignement de taille de périphérique (e.g. incohérence entre la taille du système de fichiers et le nombre total d'octets empêchant de monter la partition) :

```
sudo btrfs rescue fix-device-size {{chemin/vers/partition}}
```

- Restaurer un superbloc corrompu depuis ses copies correctes (restauration de la racine de l'arbre du système de fichiers) :

```
sudo btrfs rescue super-recover {{chemin/vers/partition}}
```

- Restaurer depuis des transactions interrompues (correction des problèmes de re-exécution des messages de journaux) :

```
sudo btrfs rescue zero-log {{chemin/vers/partition}}
```

- Créer un périphérique de contrôle sous `/dev/btrfs-control` quand l'outil `mknod` n'est pas installé :

```
sudo btrfs rescue create-control-device
```

# btrfs restore

Tenter de récupérer des fichiers depuis un système de fichiers btrfs endommagé.

Plus d'informations : <https://btrfs.readthedocs.io/en/latest/btrfs-restore.html>.

- Restaurer tout les fichiers depuis un système de fichier btrfs vers un répertoire cible indiqué :

```
sudo btrfs restore {{chemin/vers/peripherique_btrfs}}  
{{chemin/vers/dossier}}
```

- Lister (sans écriture) les fichiers qui peuvent être récupérés depuis un système de fichiers btrfs :

```
sudo btrfs restore --dry-run {{chemin/du/device/btrfs}}  
{{chemin/du/dossier}}
```

- Restaurer les fichiers correspondants à une expression régulière donnée (non sensible à la casse) à restaurer depuis un système de fichiers btrfs (tous les répertoires parents des fichiers doivent correspondre également à l'expression régulière) :

```
sudo btrfs restore --path-regex {{expression_reguliere}} -c  
{{chemin/vers/peripherique_btrfs}} {{chemin/vers/dossier}}
```

- Restaurer les fichiers depuis un système de fichiers btrfs en utilisant un arbre racine spécifique bytenr (voir `btrfs - find - root`) :

```
sudo btrfs restore -t {{bytenr}} {{chemin/vers/  
peripherique_btrfs}} {{chemin/vers/dossier}}
```

- Restaurer les fichiers depuis un système de fichiers btrfs (avec métadonnées, attributs étendus, et liens symboliques) en écrivant par dessus les fichiers déjà existants dans le répertoire cible :

```
sudo btrfs restore --metadata --xattr --symlinks --overwrite  
{{chemin/vers/peripherique_btrfs}} {{chemin/vers/dossier}}
```

# btrfs scrub

Examiner un système de fichiers btrfs pour vérifier l'intégrité de ses données.

Il est recommandé de faire tourner une vérification tous les mois.

Plus d'informations : <https://btrfs.readthedocs.io/en/latest/btrfs-scrub.html>.

- Démarrer un examen :

```
sudo btrfs scrub start {{chemin/vers/point_de_montage_btrfs}}
```

- Afficher le statut d'un examen en cours, ou du dernier examen complété :

```
sudo btrfs scrub status {{chemin/vers/  
point_de_montage_btrfs}}
```

- Stopper un examen en cours :

```
sudo btrfs scrub cancel {{chemin/vers/  
point_de_montage_btrfs}}
```

- Reprendre un examen précédemment stoppé :

```
sudo btrfs scrub resume {{chemin/vers/  
point_de_montage_btrfs}}
```

- Démarrer un examen, mais attendre qu'il termine avant de rendre la main :

```
sudo btrfs scrub start -B {{chemin/vers/  
point_de_montage_btrfs}}
```

- Démarrer un examen en mode silencieux (n'affiche ni erreurs ni statistiques) :

```
sudo btrfs scrub start -q {{chemin/vers/le/  
point_de_montage_btrfs}}
```

# btrfs subvolume

Gestion des sous-volumes et instantanés btrfs.

Plus d'informations : <https://btrfs.readthedocs.io/en/latest/btrfs-subvolume.html>.

- Créer un nouveau sous-volume vide :

```
sudo btrfs subvolume create {{chemin/vers/  
nouveau_sous_volume}}
```

- Lister tous les sous-volumes et instantanés du système de fichiers indiqué :

```
sudo btrfs subvolume list {{chemin/vers/  
systeme_de_fichiers_btrfs}}
```

- Supprimer un sous-volume :

```
sudo btrfs subvolume delete {{chemin/vers/sous_volume}}
```

- Créer un instantané en lecture seule d'un sous-volume existant :

```
sudo btrfs subvolume snapshot -r {{chemin/vers/  
sous_volume_source}} {{chemin/vers/sous_volume_cible}}
```

- Créer un instantané en lecture et écriture d'un sous-volume existant :

```
sudo btrfs subvolume snapshot {{chemin/vers/  
sous_volume_source}} {{chemin/vers/sous_volume_cible}}
```

- Afficher les informations détaillées d'un sous-volume :

```
sudo btrfs subvolume show {{chemin/vers/sous_volume}}
```

# btrfs version

Afficher les informations de version des outils btrfs, et accéder aux pages d'aide.

Plus d'informations : <https://btrfs.readthedocs.io/en/latest/btrfs.html>.

- Afficher l'aide :

```
btrfs version --help
```

- Afficher les informations de version des outils btrfs :

```
btrfs version
```



# btrfs

Système de fichiers basé sur le principe de copie à l'écriture ("copy-on-write", souvent désigné par son sigle anglais COW) pour Linux.

Certaines sous-commandes comme **device** ont leur propre documentation.

Plus d'informations : <https://btrfs.readthedocs.io/en/latest/btrfs.html>.

- Créer un sous-volume :

```
sudo btrfs subvolume create {{chemin/vers/sous_volume}}
```

- Lister les sous-volumes :

```
sudo btrfs subvolume list {{chemin/vers/point_de_montage}}
```

- Afficher les informations d'utilisation d'espace :

```
sudo btrfs filesystem df {{chemin/vers/point_de_montage}}
```

- Activer les quotas :

```
sudo btrfs quota enable {{chemin/vers/sous_volume}}
```

- Afficher les quotas :

```
sudo btrfs qgroup show {{chemin/vers/sous_volume}}
```

# cal

Affiche un calendrier, en surlignant la date d'aujourd'hui.

Plus d'informations : <https://manned.org/cal>.

- Affiche un calendrier pour le mois en cours :

```
cal
```

- Affiche les mois précédent, courant et suivant :

```
cal -3
```

- Utilise le Lundi comme le premier jour de la semaine :

```
cal --monday
```

- Affiche un calendrier pour une année en particulier (4 chiffres) :

```
cal {{année}}
```

- Affiche un calendrier pour un mois et une année en particulier :

```
cal {{mois}} {{année}}
```

# CC

Cette commande est un alias de **gcc**.

Plus d'informations : <https://gcc.gnu.org>.

- Voir la documentation de la commande originale :

```
tldr gcc
```

# cfdisk

Un programme pour gérer les tables de partitions et les partitions sur un disque dur en utilisant une interface utilisateur de type "curses".

Plus d'informations : <https://manned.org/cfdisk>.

- Lance le manipulateur de partitions sur un appareil spécifique :

```
cfdisk {{/dev/sdX}}
```

- Crée une nouvelle table de partitions pour un appareil spécifique et la gère :

```
cfdisk --zero {{/dev/sdX}}
```

# chcon

Change le contexte de sécurité de SELinux d'un ou plusieurs fichiers/dossiers.

Plus d'informations : <https://www.gnu.org/software/coreutils/chcon>.

- Affiche le contexte de sécurité d'un fichier :

```
ls -lZ {{chemin/vers/fichier}}
```

- Change le contexte de sécurité d'un fichier cible, en utilisant un fichier de référence :

```
chcon --reference={{fichier_référence}} {{fichier_cible}}
```

- Change le contexte de sécurité SELinux complet d'un fichier :

```
chcon {{utilisateur}}:{{role}}:{{type}}:{{range/level}}  
{{fichier}}
```

- Change seulement la partie utilisateur du contexte de sécurité SELinux :

```
chcon -u {{utilisateur}} {{fichier}}
```

- Change seulement la partie role du contexte de sécurité SELinux :

```
chcon -r {{role}} {{fichier}}
```

- Change seulement la partie type du contexte de sécurité SELinux :

```
chcon -t {{type}} {{fichier}}
```

- Change seulement la partie plage/niveau du contexte de sécurité SELinux :

```
chcon -l {{plage/niveau}} {{fichier}}
```

# cmus

Lecteur de musique en ligne de commande.

Utiliser les flèches pour naviguer, **<enter/return>** pour sélectionner, et les nombres 1-8 pour se déplacer dans les différentes vues.

Plus d'informations : <https://cmus.github.io>.

- Ouvre cmus dans le dossier spécifié (celui-ci deviendra le dossier de travail) :

```
cmus {{chemin/vers/dossier}}
```

- Ajoute un fichier/dossier à la bibliothèque :

```
:add {{chemin/vers/fichier_ou_dossier}}
```

- Mets en pause/relancer la musique actuelle :

```
c
```

- Active/désactive le mode aléatoire :

```
s
```

- Quitte cmus :

```
q
```

# cp

Copier fichiers et répertoires.

Plus d'informations : <https://www.gnu.org/software/coreutils/cp>.

- Copier un fichier vers un autre emplacement :

```
cp {{chemin/vers/fichier_original.ext}} {{chemin/vers/fichier_cible.ext}}
```

- Copier un fichier d'un répertoire vers un autre en conservant le nom du fichier :

```
cp {{chemin/vers/fichier_original.ext}} {{chemin/vers/repertoire_cible}}
```

- Copier récursivement le contenu d'un répertoire vers un autre emplacement (si la destination existe, le répertoire est copié dans celle-ci) :

```
cp -r {{chemin/vers/repertoire_source}} {{chemin/vers/repertoire_cible}}
```

- Copier récursivement le contenu d'un répertoire vers un autre emplacement en mode verbeux (affichage des noms fichiers à mesure de leur copie) :

```
cp -vr {{chemin/vers/repertoire_source}} {{chemin/vers/repertoire_cible}}
```

- Copier les fichiers textes vers un autre emplacement, en mode interactive (demande une confirmation avant d'écrire par dessus un fichier du même nom) :

```
cp -i {{*.txt}} {{chemin/vers/repertoire_cible}}
```

- Suivre le lien symbolique avant de copier (copie le fichier lié, et non le lien) :

```
cp -L {{link}} {{chemin/vers/repertoire_cible}}
```

- Utiliser le chemin complet du fichier source, créant tout répertoire manquant lors de la copie :

```
cp --parents {{chemin/vers/fichier_source}} {{chemin/vers/fichier_cible}}
```

# ctr

Gère les conteneurs et images de **containerd**.

Plus d'informations : <https://containerd.io>.

- Liste tous les conteneurs (en marche et arrêtés) :

```
ctr containers list
```

- Liste toutes les images :

```
ctr images list
```

- Télécharge une image :

```
ctr images pull {{image}}
```

- Étiquette une image :

```
ctr images tag {{image_source}}:{{étiquette_source}}  
{{image_destination}}:{{étiquette_destination}}
```



# curlie

Curlie est un frontend pour curl qui ajoute la facilité d'utilisation de httpie.

Plus d'informations : <https://github.com/rs/curlie>.

- Envoie une requête GET :

```
curlie {{httpbin.org/get}}
```

- Envoie une requête POST :

```
curlie post {{httpbin.org/post}} {{name=john}} {{age:=25}}
```

- Envoie une requête GET avec des paramètres de requête (par ex : premier\_param=5&deuxième\_param=true):

```
curlie get {{httpbin.org/get}} {{premier_param=5}}  
{{second_param=true}}
```

- Envoie une requête GET avec un en-tête personnalisé :

```
curlie get {{httpbin.org/get}} {{clef-d-en-tête:valeur-d-en-tête}}
```

# daemonize

Lance une commande (qui ne se "démonise" pas elle-même) comme démon UNIX.

Plus d'informations : <https://software.clapper.org/daemonize/>.

- Lance une commande comme démon :

```
daemonize {{commande}} {{arguments_de_commande}}
```

- Écrit le PID dans le fichier spécifié :

```
daemonize -p {{chemin/vers/le/fichier/pid}} {{commande}}  
{{arguments_de_commande}}
```

- Utilise un fichier verrou pour s'assurer que seulement une instance fonctionne à la fois :

```
daemonize -l {{chemin/vers/le/fichier/verrou}} {{commande}}  
{{arguments_de_commande}}
```

- Utilise le compte utilisateur spécifié :

```
sudo daemonize -u {{utilisateur}} {{commande}}  
{{arguments_de_commande}}
```

# deluser

Supprime un utilisateur du système.

Plus d'informations : <https://manned.org/deluser>.

- Supprime un utilisateur :

```
sudo deluser {{nom_dutilisateur}}
```

- Supprime un utilisateur et son répertoire "home" :

```
sudo deluser --remove-home {{nom_dutilisateur}}
```

- Supprime un utilisateur et son répertoire "home", mais sauvegarde ses fichiers dans une archive `.tar.gz` dans le répertoire spécifié :

```
sudo deluser --backup-to {{chemin/vers/le/répertoire/de/sauvegarde}} --remove-home {{nom_dutilisateur}}
```

- Supprime un utilisateur et tous les fichiers qu'il possède :

```
sudo deluser --remove-all-files {{nom_dutilisateur}}
```

# dex

DesktopEntry Execution est un programme servant à générer et à exécuter des fichiers DesktopEntry de type Application.

Plus d'informations : <https://github.com/jceb/dex>.

- Exécute tous les programmes dans les dossiers de démarrage automatique :

```
dex --autostart
```

- Exécute tous les programmes dans les dossiers spécifiés :

```
dex --autostart --search-paths {{chemin/vers/dossier1}}:{{chemin/vers/dossier2}}:{{chemin/vers/dossier3}}:
```

- Prévisualise les programmes qui seraient exécutés lors d'un démarrage automatique spécifique à GNOME :

```
dex --autostart --environment {{GNOME}}
```

- Prévisualise les programmes qui seraient exécutés lors d'un démarrage automatique standard :

```
dex --autostart --dry-run
```

- Prévisualise la valeur de la propriété Name de DesktopEntry :

```
dex --property {{Name}} {{chemin/vers/fichier.desktop}}
```

- Crée une DesktopEntry pour un programme dans le dossier courant :

```
dex --create {{chemin/vers/fichier.desktop}}
```

- Exécute un programme (avec Terminal=true dans le fichier Desktop) dans le terminal donné :

```
dex --term {{terminal}} {{chemin/vers/fichier.desktop}}
```

# dos2unix

Remplace les fins de lignes de style DOS par des fins de lignes de style Unix.

Remplace CRLF par LF.

Voir également **unix2dos**, **unix2mac**, et **mac2unix**.

Plus d'informations : <https://manned.org/dos2unix>.

- Remplace les fins de lignes d'un fichier :

```
dos2unix {{chemin/vers/fichier}}
```

- Crée une copie avec des fins de lignes de type Unix :

```
dos2unix {{-n|--newfile}} {{chemin/vers/fichier}} {{chemin/vers/nouveau_fichier}}
```

- Affiche les informations d'un fichier :

```
dos2unix {{-i|--info}} {{chemin/vers/fichier}}
```

- Conserve/Ecrit/Supprime la marque d'ordre des octets (BOM) :

```
dos2unix --{{keep-bom|add-bom|remove-bom}} {{chemin/vers/fichier}}
```

# edit

Un alias à l'action de modification de **run-mailcap**.

Originellement, **run-mailcap** est utilisé afin de modifier des fichiers mime-type.

Plus d'informations : <https://www.computerhope.com/unix/uedit.htm>.

- L'action de modification peut être utilisée pour voir n'importe quel fichier dans l'explorateur mailcap par défaut :

```
edit {{fichier}}
```

- Avec run-mailcap :

```
run-mailcap --action=edit {{fichier}}
```

# eject

Éjecte les CD, disquettes et les bandes magnétiques.

Plus d'informations : <https://manned.org/eject>.

- Affiche l'appareil par défaut :

```
eject -d
```

- Éjecte l'appareil par défaut :

```
eject
```

- Éjecte un appareil spécifique (l'ordre par défaut est CD-ROM, SCSI, Disquette puis bande magnétique) :

```
eject {{/dev/cdrom}}
```

- Bascule le support d'appareil en mode ouvert ou fermé :

```
eject -T {{/dev/cdrom}}
```

- Éjecte un CD :

```
eject -r {{/dev/cdrom}}
```

- Éjecte une disquette :

```
eject -f {{/mnt/disquette}}
```

- Éjecte une bande magnétique :

```
eject -q {{/mnt/bande}}
```

# flameshot

Outil de capture d'écran avec une interface graphique.

Ajoute du texte, des formes, des couleurs et envoie à imgur.

Plus d'informations : <https://flameshot.org>.

- Lancez Flameshot en mode graphique :

```
flameshot launcher
```

- Prenez une capture d'écran en cliquant et en faisant glisser :

```
flameshot gui
```

- Prenez une capture d'écran en plein écran :

```
flameshot full
```

- Définissez le chemin de sauvegarde :

```
flameshot full --path {{path/to/directory}}
```

- Retardez la capture d'écran de N millisecondes et la sortie dans le presse-papiers :

```
flameshot full --delay {{2000}} --clipboard
```



# ifdown

Désactive des interfaces réseau.

Plus d'informations : <https://manned.org/ifdown>.

- Désactive l'interface eth0 :

```
ifdown {{eth0}}
```

- Désactive toutes les interfaces déjà actives :

```
ifdown -a
```

# ifup

Outil utilisé pour activer des interfaces réseau.

Plus d'informations : <https://manned.org/ifup.8>.

- Active l'interface eth0 :

```
ifup {{eth0}}
```

- Active l'ensemble des interfaces réseau définies dans le fichier `/etc/network/interfaces` :

```
ifup -a
```

# ip address

Sous-commande de gestion des adresses IP.

Plus d'informations : <https://manned.org/ip-address>.

- Liste les interfaces réseau et leurs adresses IP associées :

```
ip address
```

- Filtre pour n'afficher que les interfaces réseau actives :

```
ip address show up
```

- Affiche les informations relatives à une interface réseau spécifique :

```
ip address show dev {{eth0}}
```

- Ajoute une adresse IP à une interface réseau :

```
ip address add {{ip_address}} dev {{eth0}}
```

- Supprimer une adresse réseau d'une interface réseau :

```
ip address delete {{ip_address}} dev {{eth0}}
```

- Supprime l'ensemble des adresses IP sur une portée donnée (scope) depuis une interface réseau :

```
ip address flush dev {{eth0}} scope {{global|host|link}}
```

# ip route list

Cette commande est un alias de **ip route show**.

- Voir la documentation de la commande originale :

`tldr ip-route-show`

# ip route show

Sous commande de gestion de l'affichage des tables de routage.

Plus d'informations : <https://manned.org/ip-route>.

- Affiche la table de routage :

```
ip route show
```

- Affiche la table de routage principale (identique au premier exemple) :

```
ip route show {{main|254}}
```

- Affiche la table de routage locale :

```
ip route show table {{local|255}}
```

- Affiche l'ensemble des tables de routage :

```
ip route show table {{all|unspec|0}}
```

- Affiche les routes d'un périphérique donné :

```
ip route show dev {{eth0}}
```

- Affiche les routes d'une portée donnée :

```
ip route show scope link
```

- Affiche le cache de routage :

```
ip route show cache
```

- N'affiche que les routes IPv6 ou IPv4 :

```
ip {{-6|-4}} route show
```

# ip

Affiche / manipule l'adressage, le routage, les interfaces et périphériques réseau, les règles de routage et les tunnels.

Certaines commandes comme **ip address** ont leur propre documentation.

Plus d'informations : <https://www.manned.org/ip.8>.

- Liste les interfaces avec des infos détaillées :

```
ip address
```

- Liste les interfaces sur la couche réseau de façon synthétique :

```
ip -brief address
```

- Liste les interfaces sur la couche liaison de façon synthétique :

```
ip -brief link
```

- Affiche la table de routage :

```
ip route
```

- Affiche les voisins (table ARP) :

```
ip neighbour
```

- Active/Désactive une interface :

```
ip link set {{interface}} {{up|down}}
```

- Ajoute/Supprime une adresse ip à une interface :

```
ip addr add/del {{ip}}/{{mask}} dev {{interface}}
```

- Ajoute une route par défaut :

```
ip route add default via {{ip}} dev {{interface}}
```

# iwctl

Un outil de ligne de commande pour gérer iwd.

Plus d'informations : <https://archive.kernel.org/oldwiki/iwd.wiki.kernel.org/gettingstarted.html>.

- Lancer le mode interactif, dans ce mode vous pouvez entrer les commandes directement, avec de l'auto-complétion :

```
iwctl
```

- Avoir l'aide générale :

```
iwctl --help
```

- Afficher vos stations wifi :

```
iwctl station list
```

- Lancer la recherche de réseaux avec une station :

```
iwctl station {{station}} scan
```

- Afficher les réseaux trouvés par une station :

```
iwctl station {{station}} get-networks
```

- Se connecter à un réseau avec une station, si des informations de connexion sont nécessaires elles seront demandées :

```
iwctl station {{station}} connect {{nom_du_réseau}}
```

# login

Démarre une session pour un utilisateur.

Plus d'informations : <https://manned.org/login>.

- Démarrer une session en tant qu'utilisateur :

```
login {{utilisateur}}
```

- Démarrer une session en tant qu'utilisateur sans authentification si jamais l'utilisateur est déjà pré-authentifié :

```
login -f {{utilisateur}}
```

- Démarrer une session en tant qu'utilisateur et en préservant l'environnement courant :

```
login -p {{utilisateur}}
```

- Démarrer une session en tant qu'utilisateur sur un hôte distant :

```
login -h {{hote}} {{utilisateur}}
```



# man

Interface de consultation des pages du manuel de référence.

Plus d'informations : <https://manned.org/man>.

- Affiche la page de manuel d'une commande :

```
man {{commande}}
```

- Affiche la page de manuel de la section 7 d'une commande :

```
man {{7}} {{commande}}
```

- Liste toutes les sections dans lesquelles se trouve une commande :

```
man --what-is {{commande}}
```

- Affiche tous les chemins où se trouvent les pages de manuel :

```
man --path
```

- Affiche l'emplacement d'une page de manuel plutôt que la page elle-même :

```
man --where {{commande}}
```

- Affiche la page de manuel dans une langue particulière :

```
man --locale={{fr_FR}} {{commande}}
```

- Cherche toutes les pages de manuel contenant la chaîne de caractères spécifiée :

```
man --apropos "{{chaîne_de_caractères}}"
```

# megadl

Cette commande est un alias de **megatools-dl**.

Plus d'informations : <https://megatools.megous.com/man/megatools-dl.html>.

- Voir la documentation de la commande originale :

```
tldr megatools-dl
```

# ncal

Cette commande est un alias de **cal**.

Plus d'informations : <https://manned.org/ncal>.

- Voir la documentation de la commande originale :

```
tldr cal
```

# pacman --database

Interagis avec les bases de données des paquets Arch Linux.

Modifie des attributs des paquets installés.

Voir aussi: **pacman**.

Plus d'informations : <https://manned.org/pacman.8>.

- Marque un paquet comme étant installé en tant que dépendance :  
`sudo pacman --database --asdeps {{paquet}}`
- Marque un paquet comme étant explicitement installé :  
`sudo pacman --database --asexplicit {{paquet}}`
- Vérifie les dépendances de tous les paquets installés :  
`pacman --database --check`
- Vérifie que toutes les dépendances des paquets installés sont disponibles dans les dépôts :  
`pacman --database --check --check`
- N'affiche que les messages d'erreur :  
`pacman --database --check --quiet`
- Affiche l'aide :  
`pacman --database --help`

# pacman --deptest

Vérifie la satisfaction des dépendances et renvoie celles qui ne le sont pas.

Voir aussi: **pacman**.

Plus d'informations : <https://manned.org/pacman.8>.

- Affiche les noms des paquets qui ne sont pas installés dans la liste :

```
pacman --deptest {{paquet1 paquet2 ...}}
```

- Vérifie que le paquet installé a une version supérieure ou égale :

```
pacman --deptest "{{bash>=5}}"
```

- Vérifie qu'une version ultérieure du paquet est installée :

```
pacman --deptest "{{bash>5}}"
```

- Affiche l'aide :

```
pacman --deptest --help
```

# pacman --files

Interagis avec les bases de données de fichiers.

Voir aussi: **pacman**, **pkgfile**.

Plus d'informations : <https://manned.org/pacman.8>.

- Mets à jour les bases de données des fichiers :

```
sudo pacman --files --refresh
```

- Trouve les paquets contenant un fichier spécifique :

```
pacman --files {{fichier}}
```

- Trouve les paquets contenant un fichier spécifique en utilisant une expression régulière :

```
pacman --files --regex '{{expression_reguliere}}'
```

- Liste uniquement les noms de paquets :

```
pacman --files --quiet {{fichier}}
```

- Liste les fichiers contenus dans un paquet :

```
pacman --files --list {{paquet}}
```

- Affiche l'aide :

```
pacman --files --help
```

# pacman-key

Script enrobeur pour GnuPG utilisé pour gérer le trousseau de clés de pacman.

Voir aussi: **pacman**.

Plus d'informations : <https://manned.org/pacman-key>.

- Initialise le trousseau de clés de pacman :

```
sudo pacman-key --init
```

- Ajoute les clés par défaut pour Arch Linux :

```
sudo pacman-key --populate {{archlinux}}
```

- Liste les clés publiques du trousseau de clés :

```
pacman-key --list-keys
```

- Ajoute les clés contenues dans le fichier spécifié :

```
sudo pacman-key --add {{chemin/vers/fichier}}
```

- Reçois une clé depuis un serveur de clés :

```
sudo pacman-key --recv-keys "{{uid|nom|email}}"
```

- Affiche l'empreinte d'une clé du trousseau de clés :

```
pacman-key --finger "{{uid|nom|email}}"
```

- Signe, localement, une clé du trousseau de clés :

```
sudo pacman-key --lsign-key "{{uid|nom|email}}"
```

- Supprime une clé :

```
sudo pacman-key --delete "{{uid|nom|email}}"
```

# pacman-mirrors

Génère une liste de miroirs pour pacman sur Manjaro Linux.

Tout appel à pacman-mirrors demande de synchroniser les bases de données et de mettre à jour le système en exécutant **sudo pacman -Syyu** en suivant.

Voir aussi: **pacman**.

Plus d'informations : <https://wiki.manjaro.org/index.php?title=Pacman-mirrors>.

- Génère une liste de miroirs avec les réglages par défaut :

```
sudo pacman-mirrors --fasttrack
```

- Obtiens le statut des miroirs actuels :

```
pacman-mirrors --status
```

- Affiche la branche courante :

```
pacman-mirrors --get-branch
```

- Utilise une branche différente :

```
sudo pacman-mirrors --api --set-branch {{stable|unstable|testing}}
```

- Génère une liste de miroirs se trouvant dans le pays associé à l'adresse IP :

```
sudo pacman-mirrors --geoip
```



# pacman --query

Fais des requêtes dans la base de données des paquets installés.

Voir aussi: **pacman**.

Plus d'informations : <https://manned.org/pacman.8>.

- Liste les paquets installés et leur version :

```
pacman --query
```

- Liste uniquement les paquets installés explicitement :

```
pacman --query --explicit
```

- Trouve le paquet propriétaire d'un fichier :

```
pacman --query --owns {{fichier}}
```

- Affiche des informations sur un paquet installé :

```
pacman --query --info {{paquet}}
```

- Liste les fichiers appartenant à un paquet :

```
pacman --query --list {{paquet}}
```

- Liste les paquets orphelins (installés en tant que dépendances mais requis par aucun paquet installé) :

```
pacman --query --unrequired --deps --quiet
```

- Liste les paquets installés ne se trouvant pas dans les dépôts :

```
pacman --query --foreign
```

- Liste les paquets périmés :

```
pacman --query --upgrades
```

# pacman --remove

Supprime des paquets.

Voir aussi: **pacman**.

Plus d'informations : <https://manned.org/pacman.8>.

- Supprime un paquet et ses dépendances :

```
sudo pacman --remove --recursive {{paquet}}
```

- Supprime un paquet, ses dépendances et ses fichiers de configuration :

```
sudo pacman --remove --recursive --nosave {{paquet}}
```

- Supprime un paquet silencieusement :

```
sudo pacman --remove --noconfirm {{paquet}}
```

- Supprime les paquets orphelins (installés en tant que dépendance mais requis par aucun paquet installé) :

```
sudo pacman --remove --recursive --nosave $(pacman --query --unrequired --deps --quiet)
```

- Supprime un paquet et les paquets qui en dépendent :

```
sudo pacman --remove --cascade {{paquet}}
```

- Affiche les paquets qui seraient affectés par la commande sans agir :

```
pacman --remove --print {{paquet}}
```

- Affiche l'aide :

```
pacman --remove --help
```

# pacman --sync

Synchronise les paquets.

Voir aussi: **pacman**.

Plus d'informations : <https://manned.org/pacman.8>.

- Installe un nouveau paquet :

```
sudo pacman --sync {{paquet}}
```

- Synchronise et mettre à jour :

```
sudo pacman --sync --refresh --sysupgrade
```

- Synchronise, mets à jour et installe un paquet sans demander de confirmation :

```
sudo pacman --sync --refresh --sysupgrade --noconfirm  
{{paquet}}
```

- Recherche un paquet en utilisant un nom ou une expression régulière :

```
pacman --sync --search "{{motif}}"
```

- Affiche des informations sur un paquet :

```
pacman --sync --info {{paquet}}
```

- Ecris par dessus des fichiers pendant une mise à jour :

```
sudo pacman --sync --refresh --sysupgrade --overwrite {{path/  
to/file}}
```

- Synchronise et mets à jour les paquets, en ignorant un paquet (peut être utilisé plusieurs fois) :

```
sudo pacman --sync --refresh --sysupgrade --ignore {{paquet}}
```

- Supprime les fichiers concernant des paquets non installés et les dépôts supprimés du cache de pacman :

```
sudo pacman --sync --clean
```

# pacman --upgrade

Mets à jour ou ajoute des paquets au système.

Voir aussi: **pacman**.

Plus d'informations : <https://manned.org/pacman.8>.

- Installe un ou des paquets depuis le système de fichiers :

```
sudo pacman --upgrade {{chemin/vers/paquet1.pkg.tar.zst}}  
{{chemin/vers/paquet2.pkg.tar.zst}}
```

- Installe un paquet silencieusement :

```
sudo pacman --upgrade --noconfirm {{chemin/vers/  
paquet.pkg.tar.zst}}
```

- Ecris par dessus les fichiers en conflit pendant l'installation du paquet :

```
sudo pacman --upgrade --overwrite {{chemin/vers/fichier}}  
{{chemin/vers/paquet.pkg.tar.zst}}
```

- Installe un paquet sans vérifier les dépendances :

```
sudo pacman --upgrade --nodeps {{chemin/vers/  
paquet.pkg.tar.zst}}
```

- Affiche ce qui se passerait si la commande était exécutée mais sans agir :

```
pacman --upgrade --print {{chemin/vers/paquet.pkg.tar.zst}}
```

- Affiche l'aide :

```
pacman --upgrade --help
```

# pacman

Outil de gestion de paquets sur Arch Linux.

Voir aussi: **pacman-database**, **pacman-deptest**, **pacman-files**, **pacman-key**, **pacman-mirrors**, **pacman-query**, **pacman-remove**, **pacman-sync**, **pacman-upgrade**.

Plus d'informations : <https://manned.org/pacman.8>.

- Synchronise et mets à jour tous les paquets :

```
sudo pacman -Syu
```

- Installe un nouveau paquet :

```
sudo pacman -S {{nom_paquet}}
```

- Efface un paquet et ses dépendances :

```
sudo pacman -Rs {{nom_paquet}}
```

- Recherche dans la base de données des paquets contenant un certain fichier :

```
pacman -F "{{nom_fichier}}"
```

- Liste les paquets installés et leurs versions :

```
pacman -Q
```

- Liste seulement les paquets installés explicitement et leurs versions :

```
pacman -Qe
```

- Vide le cache des paquets pour libérer de l'espace :

```
sudo pacman -Scc
```

# reboot

Redémarre le système.

Plus d'informations : <https://manned.org/reboot.8>.

- Redémarre le système :

```
reboot
```

- Éteint le système (identique à `poweroff`) :

```
reboot --poweroff
```

- Arrête (met fin à tous les processus et arrête le processeur) le système (identique à `halt`) :

```
reboot --halt
```

- Redémarre immédiatement sans contacter le gestionnaire du système :

```
reboot --force
```

- Écrit l'entrée `wtmp shutdown` sans redémarrer le système :

```
reboot --wtmp-only
```

# resolvectl

Résout les noms de domaine, les adresses IPV4 et IPV6, les enregistrements de ressources DNS et les services.

Inspecte et reconfigure le résolveur DNS.

Plus d'informations : <https://www.freedesktop.org/software/systemd/man/resolvectl.html>.

- Affiche les paramètres DNS :

```
resolvectl status
```

- Résout les adresses IPv4 et IPv6 pour un ou plusieurs domaines :

```
resolvectl query {{domaine1 domaine2 ...}}
```

- Récupère le domaine d'une IP spécifiée :

```
resolvectl query {{adresse_ip}}
```

- Vide tous les caches DNS locaux :

```
resolvectl flush-caches
```

- Affiche les statistiques DNS (transactions, cache et verdicts DNSSEC) :

```
resolvectl statistics
```

- Récupère un enregistrement MX du domaine :

```
resolvectl --legend={{no}} --type={{MX}} query {{domaine}}
```

- Résout un enregistrement SRV, par exemple \_xmpp-server.\_tcp gmail.com :

```
resolvectl service _{{service}}._{{protocole}} {{nom}}
```

- Récupère une clé TLS :

```
resolvectl tlsa tcp {{domain}}:443
```

# shutdown

Éteint et redémarre le système.

Plus d'informations : <https://manned.org/shutdown.8>.

- Éteint (arrête) immédiatement :

```
shutdown -h now
```

- Redémarre immédiatement :

```
shutdown -r now
```

- Redémarre dans 5 minutes :

```
shutdown -r +{{5}} &
```

- Éteint à 1:00 pm (Utilise un format 24h) :

```
shutdown -h 13:00
```

- Annule une opération d'arrêt ou de redémarrage du système en attente :

```
shutdown -c
```



# sysctl

Liste et modifie les variables d'exécution du noyau.

Plus d'informations : <https://manned.org/sysctl.8>.

- Affiche toutes les variables disponibles et leurs valeurs :

```
sysctl -a
```

- Définis une variable d'état modifiable du noyau :

```
sysctl -w {{section.modifiable}}={{valeur}}
```

- Obtiens les gestionnaires de fichiers (handlers) actuellement ouverts :

```
sysctl fs.file-nr
```

- Obtiens la limite de nombre de fichiers ouverts simultanément :

```
sysctl fs.file-max
```

- Applique les changements de `/etc/sysctl.conf` :

```
sysctl -p
```

# systemctl

Contrôle le système systemd et le gestionnaire de services.

Plus d'informations : <https://www.freedesktop.org/software/systemd/man/systemctl.html>.

- Liste des unités en échec :

```
systemctl --failed
```

- Démarre/arrête/redémarre/recharge un service :

```
systemctl {{start|stop|restart|reload}} {{unité}}
```

- Affiche le statut d'une unité :

```
systemctl status {{unité}}
```

- Active/désactive une unité à démarrer au démarrage :

```
systemctl {{enable|disable}} {{unité}}
```

- Masque/démasque une unité pour empêcher l'activation et l'activation manuelle :

```
systemctl {{mask|unmask}} {{unité}}
```

- Rechargement de systemd, recherche d'unités nouvelles ou modifiées :

```
systemctl daemon-reload
```

- Vérifie si une unité est activée :

```
systemctl is-enabled {{unité}}
```

# ubuntu-bug

Cette commande est un alias de **apport-bug**.

Plus d'informations : <https://manned.org/ubuntu-bug>.

- Voir la documentation de la commande originale :

```
tldr apport-bug
```

# useradd

Crée un nouvel utilisateur.

Voir aussi **users**, **userdel**, **usermod**.

Plus d'informations : <https://manned.org/useradd>.

- Crée un nouvel utilisateur :

```
sudo useradd {{nom_utilisateur}}
```

- Crée un nouvel utilisateur en spécifiant un identifiant numérique particulier :

```
sudo useradd --uid {{identifiant}} {{nom_utilisateur}}
```

- Crée un nouvel utilisateur avec le shell spécifié :

```
sudo useradd --shell {{chemin/vers/shell}}  
{{nom_utilisateur}}
```

- Crée un nouvel utilisateur qui appartient aux groupes supplémentaires (attention à l'omission des espaces) :

```
sudo useradd --groups {{groupe1,groupe2,...}}  
{{nom_utilisateur}}
```

- Crée un nouvel utilisateur avec le répertoire personnel par défaut :

```
sudo useradd --create-home {{nom_utilisateur}}
```

- Crée un nouvel utilisateur avec un répertoire personnel rempli par les fichiers et répertoires d'un répertoire squelette :

```
sudo useradd --skel {{chemin/vers/repertoire_squelette}} --  
create-home {{nom_utilisateur}}
```

- Crée un nouvel utilisateur système sans répertoire personnel :

```
sudo useradd --system {{nom_utilisateur}}
```

# userdel

Supprime un compte utilisateur ou supprime un utilisateur d'un groupe.

Voir aussi **users**, **useradd**, **usermod**.

Plus d'informations : <https://manned.org/userdel>.

- Supprime un utilisateur :

```
sudo userdel {{nom_utilisateur}}
```

- Supprime un utilisateur dans un autre répertoire racine :

```
sudo userdel --root {{chemin/vers/autre_racine}}  
{{nom_utilisateur}}
```

- Supprime un utilisateur, son répertoire personnel ainsi que son répertoire d'attente des courriels :

```
sudo userdel --remove {{nom_utilisateur}}
```

# usermod

Modifie le compte d'un utilisateur.

Voir aussi **users**, **useradd**, **userdel**.

Plus d'informations : <https://manned.org/usermod>.

- Change le nom d'un utilisateur :

```
sudo usermod --login {{nouveau_nom}} {{nom_utilisateur}}
```

- Modifie l'identifiant numérique d'un utilisateur :

```
sudo usermod --uid {{identifiant}} {{nom_utilisateur}}
```

- Change le shell d'un utilisateur :

```
sudo usermod --shell {{chemin/vers/shell}}  
{{nom_utilisateur}}
```

- Ajoute l'utilisateur à des groupes supplémentaires (attention à l'omission d'espaces) :

```
sudo usermod --append --groups {{groupe1,groupe2,...}}  
{{nom_utilisateur}}
```

- Change le répertoire personnel d'un utilisateur et déplace ses fichiers vers celui-ci :

```
sudo usermod --move-home --home {{chemin/vers/  
nouveau_répertoire}} {{nom_utilisateur}}
```

# wg

Gestion de la configuration des interfaces WireGuard.

Plus d'informations : <https://www.wireguard.com/quickstart/>.

- Vérifier l'état des interfaces actuellement actives :

```
sudo wg
```

- Générer une clé privée :

```
wg genkey
```

- Générer une clé publique à partir d'une clé privée :

```
wg pubkey < {{chemin/vers/clé_privée}} > {{chemin/vers/clé_publique}}
```

- Générer une clé publique et privée :

```
wg genkey | tee {{chemin/vers/clé_privée}} | wg pubkey > {{chemin/vers/clé_publique}}
```

- Afficher la configuration actuelle d'une interface wireguard :

```
sudo wg showconf {{wg0}}
```

# xbacklight

Outil pour ajuster la luminosité du rétroéclairage en utilisant l'extension RandR.

Plus d'informations : <https://gitlab.freedesktop.org/xorg/app/xbacklight>.

- Obtient le niveau de luminosité de l'écran actuel comme un pourcentage :

```
xbacklight
```

- Régle la luminosité de l'écran à 40% :

```
xbacklight -set {{40}}
```

- Augmente la luminosité de l'écran actuel de 25% :

```
xbacklight -inc {{25}}
```

- Diminue la luminosité de l'écran actuel de 75% :

```
xbacklight -dec {{75}}
```

- Augment la luminosité de l'écran à 100%, étendu sur 60 secondes (valeur donnée en ms), en 60 pas :

```
xbacklight -set {{100}} -time {{60000}} -steps {{60}}
```



# xclip

Outil de manipulation de presse-papiers X11, semblable à **xsel**.

Gère les sélections primaires et secondaires de X, en plus du presse-papier système (**Ctrl + C**/**Ctrl + V**).

Plus d'informations : <https://manned.org/xclip>.

- Copie la sortie d'une commande vers la zone de sélection primaire de X11 (presse-papiers) :

```
echo 123 | xclip
```

- Copie la sortie d'une commande vers une zone de sélection de X11 donnée :

```
echo 123 | xclip -selection {{primary|secondary|clipboard}}
```

- Copie le contenu d'un fichier vers le presse-papiers système, avec la notation courte :

```
echo 123 | xclip -sel clip
```

- Copie le contenu d'un fichier vers le presse-papiers système :

```
xclip -sel clip {{fichier_entrée.txt}}
```

- Copie le contenu d'une image PNG vers le presse-papiers système (peut être collé dans d'autres programmes correctement) :

```
xclip -sel clip -t image/png {{fichier_entrée.png}}
```

- Colle le contenu de la zone de sélection de X11 à la console :

```
xclip -o
```

- Colle le contenu du presse-papier système à la console :

```
xclip -o -sel clip
```

# xsel

Outil de sélection et de manipulation du presse-papiers X11.

Plus d'informations : <https://manned.org/xsel>.

- Utilise la sortie d'une commande comme entrée du presse-papiers (équivalent de `Ctrl + C`):

```
echo 123 | xsel -ib
```

- Utilise le contenu d'un fichier comme entrée du presse-papiers :

```
cat {{fichier}} | xsel -ib
```

- Affiche le contenu du presse-papiers dans le terminal (équivalent à `Ctrl + V`):

```
xsel -ob
```

- Sortie du contenu du presse-papiers dans un fichier :

```
xsel -ob > {{fichier}}
```

- Efface le presse-papiers :

```
xsel -cb
```

- Affiche le contenu de la sélection primaire X11 dans le terminal (équivalent à un clic du milieu de la souris) :

```
xsel -op
```

# yay

Yet Another Yogurt : Un outil pour Arch Linux pour construire et installer des paquets depuis le Arch User Repository.

À regarder : **pacman**.

Plus d'informations : <https://github.com/Jguer/yay>.

- Recherche interactivement et installe des paquets depuis les dépôts et l'AUR :

```
yay {{nom_paquet|terme_recherche}}
```

- Synchronise et met à jour tous les paquets depuis les dépôts et l'AUR :

```
yay
```

- Synchronise et met à jour seulement les paquets de l'AUR :

```
yay -Sua
```

- Installe un nouveau paquet depuis les dépôts et l'AUR :

```
yay -S {{nom_paquet}}
```

- Recherche dans la base de données de paquets un mot clé depuis les dépôts et l'AUR :

```
yay -Ss {{mot_clé}}
```

- Montre des statistiques sur les paquets installés et la santé du système :

```
yay -Ps
```

# zypper

SUSE & openSUSE utilitaire de gestion de paquets.

Plus d'informations : [https://en.opensuse.org/SDB:Zypper\\_manual](https://en.opensuse.org/SDB:Zypper_manual).

- Synchroniser la liste des paquets et versions disponibles :

```
zypper refresh
```

- Installer un nouveau paquet :

```
zypper install {{paquet}}
```

- Supprimer un paquet :

```
zypper remove {{paquet}}
```

- Mettre à jour un paquet installé vers la version la plus récente disponible :

```
zypper update
```

- Chercher un paquet par mot clef :

```
zypper search {{mot_clef}}
```

- Afficher les informations concernant les dépôts de paquets configurés :

```
zypper repos --sort-by-priority
```

Osx

# aa

Cette commande est un alias de **yaa**.

- Voir la documentation de la commande originale :

```
tldr yaa
```

# g[

Cette commande est un alias de **-p linux** [.

- Voir la documentation de la commande originale :

```
tldr -p linux [
```

# gawk

Cette commande est un alias de **-p linux awk**.

- Voir la documentation de la commande originale :

```
tldr -p linux awk
```



# gb2sum

Cette commande est un alias de **-p linux b2sum**.

- Voir la documentation de la commande originale :

```
tldr -p linux b2sum
```

# gbase32

Cette commande est un alias de **-p linux base32**.

- Voir la documentation de la commande originale :

```
tldr -p linux base32
```

# gbase64

Cette commande est un alias de **-p linux base64**.

- Voir la documentation de la commande originale :

```
tldr -p linux base64
```

# gbasename

Cette commande est un alias de **-p linux basename**.

- Voir la documentation de la commande originale :

```
tldr -p linux basename
```

# gbasenc

Cette commande est un alias de **-p linux basenc**.

- Voir la documentation de la commande originale :

```
tldr -p linux basenc
```

# gcat

Cette commande est un alias de **-p linux cat**.

- Voir la documentation de la commande originale :

```
tldr -p linux cat
```

# gchcon

Cette commande est un alias de **-p linux chcon**.

- Voir la documentation de la commande originale :

```
tldr -p linux chcon
```

# gchgrp

Cette commande est un alias de **-p linux chgrp**.

- Voir la documentation de la commande originale :

```
tldr -p linux chgrp
```



# gchmod

Cette commande est un alias de **-p linux chmod**.

- Voir la documentation de la commande originale :

```
tldr -p linux chmod
```

# gchown

Cette commande est un alias de **-p linux chown**.

- Voir la documentation de la commande originale :

```
tldr -p linux chown
```

# gchroot

Cette commande est un alias de **-p linux chroot**.

- Voir la documentation de la commande originale :

```
tldr -p linux chroot
```

# gcksum

Cette commande est un alias de **-p linux cksum**.

- Voir la documentation de la commande originale :

```
tldr -p linux cksum
```

# gcomm

Cette commande est un alias de **-p linux comm**.

- Voir la documentation de la commande originale :

```
tldr -p linux comm
```

# gcp

Cette commande est un alias de **-p linux cp**.

- Voir la documentation de la commande originale :

```
tldr -p linux cp
```

# gcsplit

Cette commande est un alias de **-p linux csplit**.

- Voir la documentation de la commande originale :

```
tldr -p linux csplit
```

# gcut

Cette commande est un alias de **-p linux cut**.

- Voir la documentation de la commande originale :

```
tldr -p linux cut
```



# gdate

Cette commande est un alias de **-p linux date**.

- Voir la documentation de la commande originale :

```
tldr -p linux date
```

# gdd

Cette commande est un alias de **-p linux dd**.

- Voir la documentation de la commande originale :

```
tldr -p linux dd
```

# gdf

Cette commande est un alias de **-p linux df**.

- Voir la documentation de la commande originale :

```
tldr -p linux df
```

# gdir

Cette commande est un alias de **-p linux dir**.

- Voir la documentation de la commande originale :

```
tldr -p linux dir
```

# gdircolors

Cette commande est un alias de **-p linux dircolors**.

- Voir la documentation de la commande originale :

```
tldr -p linux dircolors
```

# gdirname

Cette commande est un alias de **-p linux dirname**.

- Voir la documentation de la commande originale :

```
tldr -p linux dirname
```

# gdnsdomainname

Cette commande est un alias de **-p linux dnsdomainname**.

- Voir la documentation de la commande originale :

```
tldr -p linux dnsdomainname
```

# gecho

Cette commande est un alias de **-p linux echo**.

- Voir la documentation de la commande originale :

```
tldr -p linux echo
```



# ged

Cette commande est un alias de **-p linux ed**.

- Voir la documentation de la commande originale :

```
tldr -p linux ed
```

# gegrep

Cette commande est un alias de **-p linux egrep**.

- Voir la documentation de la commande originale :

```
tldr -p linux egrep
```

# genv

Cette commande est un alias de **-p linux env**.

- Voir la documentation de la commande originale :

```
tldr -p linux env
```

# gexpand

Cette commande est un alias de **-p linux expand**.

- Voir la documentation de la commande originale :

```
tldr -p linux expand
```

# gexpr

Cette commande est un alias de **-p linux expr**.

- Voir la documentation de la commande originale :

```
tldr -p linux expr
```

# gfactor

Cette commande est un alias de **-p linux factor**.

- Voir la documentation de la commande originale :

```
tldr -p linux factor
```

# gfalse

Cette commande est un alias de **-p linux false**.

- Voir la documentation de la commande originale :

```
tldr -p linux false
```

# gfgrep

Cette commande est un alias de **-p linux fgrep**.

- Voir la documentation de la commande originale :

```
tldr -p linux fgrep
```



# gfind

Cette commande est un alias de **-p linux find**.

- Voir la documentation de la commande originale :

```
tldr -p linux find
```

# gfmt

Cette commande est un alias de **-p linux fmt**.

- Voir la documentation de la commande originale :

```
tldr -p linux fmt
```

# gfold

Cette commande est un alias de **-p linux fold**.

- Voir la documentation de la commande originale :

```
tldr -p linux fold
```

# gftp

Cette commande est un alias de **-p linux ftp**.

- Voir la documentation de la commande originale :

```
tldr -p linux ftp
```

# ggrep

Cette commande est un alias de **-p linux grep**.

- Voir la documentation de la commande originale :

```
tldr -p linux grep
```

# ggroups

Cette commande est un alias de **-p linux groups**.

- Voir la documentation de la commande originale :

```
tldr -p linux groups
```

# ghead

Cette commande est un alias de **-p linux head**.

- Voir la documentation de la commande originale :

```
tldr -p linux head
```

# ghostid

Cette commande est un alias de **-p linux ghostid**.

- Voir la documentation de la commande originale :

```
tldr -p linux ghostid
```



# ghostname

Cette commande est un alias de **-p linux hostname**.

- Voir la documentation de la commande originale :

```
tldr -p linux hostname
```

# gid

Cette commande est un alias de **-p linux id**.

- Voir la documentation de la commande originale :

```
tldr -p linux id
```

# gifconfig

Cette commande est un alias de **-p linux ifconfig**.

- Voir la documentation de la commande originale :

```
tldr -p linux ifconfig
```

# gindent

Cette commande est un alias de **-p linux indent**.

- Voir la documentation de la commande originale :

```
tldr -p linux indent
```

# ginstall

Cette commande est un alias de **-p linux install**.

- Voir la documentation de la commande originale :

```
tldr -p linux install
```

# gjoin

Cette commande est un alias de **-p linux join**.

- Voir la documentation de la commande originale :

```
tldr -p linux join
```

# gkill

Cette commande est un alias de `-p linux kill`.

- Voir la documentation de la commande originale :

```
tldr -p linux kill
```

# glibtool

Cette commande est un alias de **-p linux libtool**.

- Voir la documentation de la commande originale :

```
tldr -p linux libtool
```



# glibtoolize

Cette commande est un alias de **-p linux libtoolize**.

- Voir la documentation de la commande originale :

```
tldr -p linux libtoolize
```

# glink

Cette commande est un alias de **-p linux link**.

- Voir la documentation de la commande originale :

```
tldr -p linux link
```

# gln

Cette commande est un alias de **-p linux ln**.

- Voir la documentation de la commande originale :

```
tldr -p linux ln
```

# glocate

Cette commande est un alias de **-p linux locate**.

- Voir la documentation de la commande originale :

```
tldr -p linux locate
```

# glogger

Cette commande est un alias de **-p linux logger**.

- Voir la documentation de la commande originale :

```
tldr -p linux logger
```

# glogname

Cette commande est un alias de **-p linux logname**.

- Voir la documentation de la commande originale :

```
tldr -p linux logname
```

# gls

Cette commande est un alias de **-p linux ls**.

- Voir la documentation de la commande originale :

```
tldr -p linux ls
```

# gmake

Cette commande est un alias de **-p linux make**.

- Voir la documentation de la commande originale :

```
tldr -p linux make
```



# gmd5sum

Cette commande est un alias de **-p linux md5sum**.

- Voir la documentation de la commande originale :

```
tldr -p linux md5sum
```

# gmkdir

Cette commande est un alias de **-p linux mkdir**.

- Voir la documentation de la commande originale :

```
tldr -p linux mkdir
```

# gmkfifo

Cette commande est un alias de **-p linux mkfifo**.

- Voir la documentation de la commande originale :

```
tldr -p linux mkfifo
```

# gmknod

Cette commande est un alias de **-p linux mknod**.

- Voir la documentation de la commande originale :

```
tldr -p linux mknod
```

# gmkttemp

Cette commande est un alias de **-p linux mktemp**.

- Voir la documentation de la commande originale :

```
tldr -p linux mktemp
```

# gmv

Cette commande est un alias de **-p linux mv**.

- Voir la documentation de la commande originale :

```
tldr -p linux mv
```

# gnice

Cette commande est un alias de **-p linux nice**.

- Voir la documentation de la commande originale :

```
tldr -p linux nice
```

# gnl

Cette commande est un alias de **-p linux nl**.

- Voir la documentation de la commande originale :

```
tldr -p linux nl
```



# gnohup

Cette commande est un alias de **-p linux nohup**.

- Voir la documentation de la commande originale :

```
tldr -p linux nohup
```

# gnproc

Cette commande est un alias de **-p linux nproc**.

- Voir la documentation de la commande originale :

```
tldr -p linux nproc
```

# gnumfmt

Cette commande est un alias de **-p linux numfmt**.

- Voir la documentation de la commande originale :

```
tldr -p linux numfmt
```

# god

Cette commande est un alias de **-p linux od**.

- Voir la documentation de la commande originale :

```
tldr -p linux od
```

# gpaste

Cette commande est un alias de **-p linux paste**.

- Voir la documentation de la commande originale :

```
tldr -p linux paste
```

# gpathchk

Cette commande est un alias de **-p linux pathchk**.

- Voir la documentation de la commande originale :

```
tldr -p linux pathchk
```

# gping

Cette commande est un alias de **-p linux ping**.

- Voir la documentation de la commande originale :

```
tldr -p linux ping
```

# gping6

Cette commande est un alias de **-p linux ping6**.

- Voir la documentation de la commande originale :

```
tldr -p linux ping6
```



# gpinky

Cette commande est un alias de **-p linux pinky**.

- Voir la documentation de la commande originale :

```
tldr -p linux pinky
```

# gpr

Cette commande est un alias de **-p linux pr**.

- Voir la documentation de la commande originale :

```
tldr -p linux pr
```

# gprintenv

Cette commande est un alias de **-p linux printenv**.

- Voir la documentation de la commande originale :

```
tldr -p linux printenv
```

# gprintf

Cette commande est un alias de **-p linux printf**.

- Voir la documentation de la commande originale :

```
tldr -p linux printf
```

# gptx

Cette commande est un alias de **-p linux ptx**.

- Voir la documentation de la commande originale :

```
tldr -p linux ptx
```

# gpwd

Cette commande est un alias de **-p linux pwd**.

- Voir la documentation de la commande originale :

```
tldr -p linux pwd
```

# grcp

Cette commande est un alias de **-p linux rcp**.

- Voir la documentation de la commande originale :

```
tldr -p linux rcp
```

# greadlink

Cette commande est un alias de **-p linux readlink**.

- Voir la documentation de la commande originale :

```
tldr -p linux readlink
```



# grealpath

Cette commande est un alias de **-p linux realpath**.

- Voir la documentation de la commande originale :

```
tldr -p linux realpath
```

# grexec

Cette commande est un alias de **-p linux rexec**.

- Voir la documentation de la commande originale :

```
tldr -p linux rexec
```

# grlogin

Cette commande est un alias de **-p linux rlogin**.

- Voir la documentation de la commande originale :

```
tldr -p linux rlogin
```

# grm

Cette commande est un alias de **-p linux rm**.

- Voir la documentation de la commande originale :

```
tldr -p linux rm
```

# grmdir

Cette commande est un alias de **-p linux rmdir**.

- Voir la documentation de la commande originale :

```
tldr -p linux rmdir
```

# grsh

Cette commande est un alias de **-p linux rsh**.

- Voir la documentation de la commande originale :

```
tldr -p linux rsh
```

# gruncon

Cette commande est un alias de **-p linux runcon**.

- Voir la documentation de la commande originale :

```
tldr -p linux runcon
```

# gsed

Cette commande est un alias de **-p linux sed**.

- Voir la documentation de la commande originale :

```
tldr -p linux sed
```



# gseq

Cette commande est un alias de **-p linux seq**.

- Voir la documentation de la commande originale :

```
tldr -p linux seq
```

# gsha1sum

Cette commande est un alias de **-p linux sha1sum**.

- Voir la documentation de la commande originale :

```
tldr -p linux sha1sum
```

# gsha224sum

Cette commande est un alias de **-p linux sha224sum**.

- Voir la documentation de la commande originale :

```
tldr -p linux sha224sum
```

# gsha256sum

Cette commande est un alias de **-p linux sha256sum**.

- Voir la documentation de la commande originale :

```
tldr -p linux sha256sum
```

# gsha384sum

Cette commande est un alias de **-p linux sha384sum**.

- Voir la documentation de la commande originale :

```
tldr -p linux sha384sum
```

# gsha512sum

Cette commande est un alias de **-p linux sha512sum**.

- Voir la documentation de la commande originale :

```
tldr -p linux sha512sum
```

# gshred

Cette commande est un alias de **-p linux shred**.

- Voir la documentation de la commande originale :

```
tldr -p linux shred
```

# gshuf

Cette commande est un alias de **-p linux shuf**.

- Voir la documentation de la commande originale :

```
tldr -p linux shuf
```



# gsleep

Cette commande est un alias de **-p linux sleep**.

- Voir la documentation de la commande originale :

```
tldr -p linux sleep
```

# gsort

Cette commande est un alias de **-p linux sort**.

- Voir la documentation de la commande originale :

```
tldr -p linux sort
```

# gsplit

Cette commande est un alias de **-p linux split**.

- Voir la documentation de la commande originale :

```
tldr -p linux split
```

# gstat

Cette commande est un alias de **-p linux stat**.

- Voir la documentation de la commande originale :

```
tldr -p linux stat
```

# gstdbuf

Cette commande est un alias de **-p linux stdbuf**.

- Voir la documentation de la commande originale :

```
tldr -p linux stdbuf
```

# gstty

Cette commande est un alias de **-p linux stty**.

- Voir la documentation de la commande originale :

```
tldr -p linux stty
```

# gsum

Cette commande est un alias de **-p linux sum**.

- Voir la documentation de la commande originale :

```
tldr -p linux sum
```

# gsync

Cette commande est un alias de **-p linux sync**.

- Voir la documentation de la commande originale :

```
tldr -p linux sync
```



# gtac

Cette commande est un alias de **-p linux tac**.

- Voir la documentation de la commande originale :

```
tldr -p linux tac
```

# gtail

Cette commande est un alias de **-p linux tail**.

- Voir la documentation de la commande originale :

```
tldr -p linux tail
```

# gtalk

Cette commande est un alias de **-p linux talk**.

- Voir la documentation de la commande originale :

```
tldr -p linux talk
```

# gtar

Cette commande est un alias de **-p linux tar**.

- Voir la documentation de la commande originale :

```
tldr -p linux tar
```

# gtee

Cette commande est un alias de **-p linux tee**.

- Voir la documentation de la commande originale :

```
tldr -p linux tee
```

# gtelnet

Cette commande est un alias de **-p linux telnet**.

- Voir la documentation de la commande originale :

```
tldr -p linux telnet
```

# gtest

Cette commande est un alias de **-p linux test**.

- Voir la documentation de la commande originale :

```
tldr -p linux test
```

# gtftp

Cette commande est un alias de **-p linux tftp**.

- Voir la documentation de la commande originale :

```
tldr -p linux tftp
```



# gtime

Cette commande est un alias de **-p linux time**.

- Voir la documentation de la commande originale :

```
tldr -p linux time
```

# gtimeout

Cette commande est un alias de **-p linux timeout**.

- Voir la documentation de la commande originale :

```
tldr -p linux timeout
```

# gtouch

Cette commande est un alias de **-p linux touch**.

- Voir la documentation de la commande originale :

```
tldr -p linux touch
```

# gtr

Cette commande est un alias de **-p linux tr**.

- Voir la documentation de la commande originale :

```
tldr -p linux tr
```

# gtracroute

Cette commande est un alias de **-p linux traceroute**.

- Voir la documentation de la commande originale :

```
tldr -p linux traceroute
```

# gtrue

Cette commande est un alias de **-p linux true**.

- Voir la documentation de la commande originale :

```
tldr -p linux true
```

# gtruncate

Cette commande est un alias de **-p linux truncate**.

- Voir la documentation de la commande originale :

```
tldr -p linux truncate
```

# gtsort

Cette commande est un alias de **-p linux tsort**.

- Voir la documentation de la commande originale :

```
tldr -p linux tsort
```



# gtty

Cette commande est un alias de **-p linux tty**.

- Voir la documentation de la commande originale :

```
tldr -p linux tty
```

# guname

Cette commande est un alias de **-p linux uname**.

- Voir la documentation de la commande originale :

```
tldr -p linux uname
```

# gunexpand

Cette commande est un alias de **-p linux unexpand**.

- Voir la documentation de la commande originale :

```
tldr -p linux unexpand
```

# guniq

Cette commande est un alias de **-p linux uniq**.

- Voir la documentation de la commande originale :

```
tldr -p linux uniq
```

# gunits

Cette commande est un alias de **-p linux units**.

- Voir la documentation de la commande originale :

```
tldr -p linux units
```

# gunlink

Cette commande est un alias de **-p linux unlink**.

- Voir la documentation de la commande originale :

```
tldr -p linux unlink
```

# gupdatedb

Cette commande est un alias de **-p linux updatedb**.

- Voir la documentation de la commande originale :

```
tldr -p linux updatedb
```

# guptime

Cette commande est un alias de **-p linux uptime**.

- Voir la documentation de la commande originale :

```
tldr -p linux uptime
```



# gusers

Cette commande est un alias de **-p linux users**.

- Voir la documentation de la commande originale :

```
tldr -p linux users
```

# gvdir

Cette commande est un alias de **-p linux vdir**.

- Voir la documentation de la commande originale :

```
tldr -p linux vdir
```

# gwc

Cette commande est un alias de **-p linux wc**.

- Voir la documentation de la commande originale :

```
tldr -p linux wc
```

# gwhich

Cette commande est un alias de **-p linux which**.

- Voir la documentation de la commande originale :

```
tldr -p linux which
```

# gwho

Cette commande est un alias de **-p linux who**.

- Voir la documentation de la commande originale :

```
tldr -p linux who
```

# gwhoami

Cette commande est un alias de **-p linux whoami**.

- Voir la documentation de la commande originale :

```
tldr -p linux whoami
```

# gwhois

Cette commande est un alias de **-p linux whois**.

- Voir la documentation de la commande originale :

```
tldr -p linux whois
```

# gxargs

Cette commande est un alias de **-p linux xargs**.

- Voir la documentation de la commande originale :

```
tldr -p linux xargs
```



# gyes

Cette commande est un alias de **-p linux yes**.

- Voir la documentation de la commande originale :

```
tldr -p linux yes
```

Sunos

# devfsadm

Commande d'administration pour **/dev**. Maintient le **/dev** espace de noms.

Plus d'informations : <https://www.unix.com/man-page/sunos/1m/devfsadm>.

- Rechercher de nouveaux disques :

```
devfsadm -c disk
```

- Nettoyez tout pendaison **/dev** liens et rechercher un nouvel appareil :

```
devfsadm -C -v
```

- Marche à sec - sortir ce qui serait changé mais ne faire aucune modification :

```
devfsadm -C -v -n
```

# dmesg

Écrire les messages du noyau sur la sortie standard.

Plus d'informations : <https://www.unix.com/man-page/sunos/1m/dmesg>.

- Afficher les messages du noyau :

```
dmesg
```

- Afficher la quantité de mémoire physique disponible sur ce système :

```
dmesg | grep -i memory
```

- Afficher les messages du noyau une page à la fois :

```
dmesg | less
```

# prctl

Obtenir ou définir les contrôles de ressources des processus, tâches et projets en cours d'exécution.

Plus d'informations : <https://www.unix.com/man-page/sunos/1/prctl>.

- Examiner les limites et les autorisations des processus :

```
prctl {{pid}}
```

- Examiner les limites et les autorisations de processus dans un format analysable par machine :

```
prctl -P {{pid}}
```

- Obtenir une limite spécifique pour un processus en cours d'exécution :

```
prctl -n process.max-file-descriptor {{pid}}
```

# prstat

Signaler les statistiques de processus actifs.

Plus d'informations : <https://www.unix.com/man-page/sunos/1m/prstat>.

- Examinez tous les processus et rapportez les statistiques triées par utilisation du processeur :

```
prstat
```

- Examinez tous les processus et rapportez les statistiques triées par utilisation de la mémoire :

```
prstat -s rss
```

- Rapporter le résumé de l'utilisation totale pour chaque utilisateur :

```
prstat -t
```

- Rapporter les informations comptables du processus de micro-état :

```
prstat -m
```

- Imprimez une liste des 5 meilleurs processeurs utilisant des processus chaque seconde :

```
prstat -c -n 5 -s cpu 1
```

# snoop

Renifleur de paquets réseau.

Équivalent SunOS de tcpdump.

Plus d'informations : <https://www.unix.com/man-page/sunos/1m/snoop>.

- Capturer des paquets sur une interface réseau spécifique :

```
snoop -d {{e1000g0}}
```

- Enregistrer les paquets capturés dans un fichier au lieu de les afficher :

```
snoop -o {{nom_de_fichier}}
```

- Afficher le résumé détaillé de la couche de protocole des paquets d'un fichier :

```
snoop -V -i {{nom_de_fichier}}
```

- Capturez les paquets réseau provenant d'un nom d'hôte et accédez à un port donné :

```
snoop to port {{port}} from host {{nom_d'hôte}}
```

- Capturez et affichez un vidage hexadécimal des paquets réseau échangés entre deux adresses IP :

```
snoop -x0 -p4 {{adresse_ip_1}} {{adresse_ip_2}}
```

# svcadm

Manipuler les instances de service.

Plus d'informations : <https://www.unix.com/man-page/linux/1m/svcadm>.

- Activer un service dans la base de données de service :

```
svcadm enable {{nom_du_service}}
```

- Désactiver le service :

```
svcadm disable {{nom_du_service}}
```

- Redémarrer un service en cours d'exécution :

```
svcadm restart {{nom_du_service}}
```

- Service de commande pour relire les fichiers de configuration :

```
svcadm refresh {{nom_du_service}}
```

- Effacer un service de l'état de maintenance et lui ordonner de démarrer :

```
svcadm clear {{nom_du_service}}
```



# SVCS

Répertorier les informations sur les services en cours d'exécution.

Plus d'informations : <https://www.unix.com/man-page/linux/1/svcs>.

- Lister tous les services en cours d'exécution :

```
svcs
```

- Lister les services qui ne fonctionnent pas :

```
svcs -vx
```

- Répertorier les informations sur un service :

```
svcs apache
```

- Afficher l'emplacement du fichier journal de service :

```
svcs -L apache
```

- Afficher la fin d'un fichier journal de service :

```
tail $(svcs -L apache)
```

# truss

Outil de débannage pour tracer les appels système.

Équivalent SunOS de strace.

Plus d'informations : <https://www.unix.com/man-page/linux/1/truss>.

- Commencez à tracer un programme en l'exécutant, en suivant tous les processus enfants :

```
truss -f {{programme}}
```

- Commencez à tracer un processus spécifique par son PID :

```
truss -p {{pid}}
```

- Commencez à tracer un programme en l'exécutant, en affichant les arguments et les variables d'environnement :

```
truss -a -e {{programme}}
```

- Comptez le temps, les appels et les erreurs pour chaque appel système et rapportez un résumé à la sortie du programme :

```
truss -c -p {{pid}}
```

- Tracez une sortie de filtrage de processus par appel système :

```
truss -p {{pid}} -t {{nom_d'appel_système}}
```

# Windows

# choco info

Afficher des informations détaillées sur un forfait avec Chocolatey.

Plus d'informations : <https://chocolatey.org/docs/commands-info>.

- Afficher des informations sur un package spécifique :

```
choco info {{paquet}}
```

- Afficher les informations pour un package local uniquement :

```
choco info {{paquet}} --local-only
```

- Spécifier une source personnalisée à partir de laquelle recevoir des informations sur les packages :

```
choco info {{paquet}} --source {{source_url|alias}}
```

- Fournir un nom d'utilisateur et un mot de passe pour l'authentification :

```
choco info {{paquet}} --user {{nom d'utilisateur}} --password {{mot de passe}}
```

# choco new

Générez de nouveaux fichiers de spécifications de package avec Chocolatey.

Plus d'informations : <https://chocolatey.org/docs/commands-new>.

- Créer un nouveau squelette de package :

```
choco new {{paquet}}
```

- Créer un nouveau package avec une version spécifique :

```
choco new {{paquet}} --version {{version}}
```

- Créer un nouveau package avec un nom de responsable spécifique :

```
choco new {{paquet}} --maintainer {{nom_mainteneur}}
```

- Créer un nouveau package dans un répertoire de sortie personnalisé :

```
choco new {{paquet}} --output-directory {{chemin/vers/
répertoire}}
```

- Créez un nouveau package avec des URL d'installation 32 bits et 64 bits spécifiques :

```
choco new {{paquet}} url="{{url}}" url64="{{url64}}"
```

# choco outdated

Vérifiez les packages obsolètes avec Chocolatey.

Plus d'informations : <https://chocolatey.org/docs/commands-outdated>.

- Afficher une liste des packages obsolètes sous forme de tableau :

```
choco outdated
```

- Ignorer les packages épinglés dans la sortie :

```
choco outdated --ignore-pinned
```

- Spécifiez une source personnalisée à partir de laquelle vérifier les packages :

```
choco outdated --source {{source_url|alias}}
```

- Fournir un nom d'utilisateur et un mot de passe pour l'authentification :

```
choco outdated --user {{nom_d_utilisateur}} --password  
{{mot_de_passe}}
```

# choco search

Recherchez un forfait local ou distant avec Chocolatey.

Plus d'informations : <https://chocolatey.org/docs/commands-search>.

- Rechercher un forfait :

```
choco search {{requête}}
```

- Rechercher un package localement :

```
choco search {{requête}} --local-only
```

- Inclure uniquement les correspondances exactes dans les résultats :

```
choco search {{requête}} --exact
```

- Confirmer automatiquement toutes les invites :

```
choco search {{requête}} --yes
```

- Spécifiez une source personnalisée dans laquelle rechercher des packages :

```
choco search {{requête}} --source {{source_url|alias}}
```

- Fournir un nom d'utilisateur et un mot de passe pour l'authentification :

```
choco search {{requête}} --user {{nom d'utilisateur}} --  
password {{mot de passe}}
```

# choco uninstall

Désinstallez un ou plusieurs packages avec Chocolatey.

Plus d'informations : <https://chocolatey.org/docs/commands-uninstall>.

- Désinstaller un ou plusieurs packages séparés par des espaces :

```
choco uninstall {{paquet1 paquet2 ...}}
```

- Désinstaller une version spécifique d'un package :

```
choco uninstall {{paquet}} --version {{version}}
```

- Confirmer automatiquement toutes les invites :

```
choco uninstall {{paquet}} --yes
```

- Supprimez toutes les dépendances lors de la désinstallation :

```
choco uninstall {{paquet}} --remove-dependencies
```

- Désinstaller tous les packages :

```
choco uninstall all
```



# choco upgrade

Surclassez un ou plusieurs forfaits avec Chocolatey.

Plus d'informations : <https://chocolatey.org/docs/commands-upgrade>.

- Mettre à niveau un ou plusieurs packages séparés par des espaces :  
`choco upgrade {{paquet1 paquet2 ...}}`
- Mise à niveau vers une version spécifique d'un package :  
`choco upgrade {{paquet}} --version {{version}}`
- Mettre à niveau tous les packages :  
`choco upgrade all`
- Mettre à niveau tous les packages sauf ceux spécifiés, séparés par des virgules :  
`choco upgrade all --except "{{paquet1 paquet2 ...}}"`
- Confirmer automatiquement toutes les invites :  
`choco upgrade {{paquet}} --yes`
- Spécifier une source personnalisée à partir de laquelle recevoir les packages :  
`choco upgrade {{paquet}} --source {{source_url|alias}}`
- Fournir un nom d'utilisateur et un mot de passe pour l'authentification :  
`choco upgrade {{paquet}} --user {{nom d'utilisateur}} --password {{mot de passe}}`

# chrome

Cette commande est un alias de **chromium**.

Plus d'informations : <https://chrome.google.com>.

- Voir la documentation de la commande originale :

```
tldr chromium
```

# cinst

Cette commande est un alias de **choco install**.

Plus d'informations : <https://docs.chocolatey.org/en-us/choco/commands/install>.

- Voir la documentation de la commande originale :

```
tldr choco install
```

# clist

Cette commande est un alias de **choco list**.

Plus d'informations : <https://docs.chocolatey.org/en-us/choco/commands/list>.

- Voir la documentation de la commande originale :

```
tldr choco list
```

# cpush

Cette commande est un alias de **choco-push**.

Plus d'informations : <https://docs.chocolatey.org/en-us/create/commands/push>.

- Voir la documentation de la commande originale :

```
tldr choco-push
```

# cuninst

Cette commande est un alias de **choco uninstall**.

Plus d'informations : <https://docs.chocolatey.org/en-us/choco/commands/uninstall>.

- Voir la documentation de la commande originale :

```
tldr choco uninstall
```

# iwr

Cette commande est un alias de **invoke-webrequest**.

Plus d'informations : <https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/invoke-webrequest>.

- Voir la documentation de la commande originale :

```
tldr invoke-webrequest
```

# pwsh where

Cette commande est un alias de **Where-Object**.

Plus d'informations : <https://learn.microsoft.com/powershell/module/microsoft.powershell.core/where-object>.

- Voir la documentation de la commande originale :

`tldr Where-Object`



# rd

Cette commande est un alias de **rmdir**.

Plus d'informations : <https://learn.microsoft.com/windows-server/administration/windows-commands/rd>.

- Voir la documentation de la commande originale :

```
tldr rmdir
```

# sls

Cette commande est un alias de **Select-String**.

Plus d'informations : <https://learn.microsoft.com/powershell/module/microsoft.powershell.utility/select-string>.

- Voir la documentation de la commande originale :

```
tldr select-string
```

# whoami

Affiche des détails sur l'utilisateur courant.

Plus d'informations : <https://learn.microsoft.com/windows-server/administration/windows-commands/whoami>.

- Affiche le nom de l'utilisateur courant :

```
whoami
```

- Affiche les groupes pour lesquels l'utilisateur courant est un membre :

```
whoami /groups
```

- Affiche les droits de l'utilisateur courant :

```
whoami /priv
```

- Affiche le nom principal d'utilisateur (UPN) de l'utilisateur courant :

```
whoami /upn
```

- Affiche l'identifiant de connexion de l'utilisateur courant :

```
whoami /logonid
```

