# Commandes utiles

# □□ Commandes en ligne de commande

> Référence personnelle — Linux, macOS, Windows

---

# □□ Linux & macOS

## □□ Fichiers & Texte

**`nano +25 fichier`** Ouvre un fichier dans l'éditeur nano directement à la ligne 25.

```
nano +25 /etc/hosts
```

---

**`tar -czvf archive.tar.gz dossier/`** Crée une archive compressée gzip. Options : `c` = créer, `z` = gzip, `v` = verbose, `f` = fichier de sortie.

```
tar -czvf backup.tar.gz /home/user/documents/
```

**`tar -xzvf archive.tar.gz -C /chemin/`** Extrait une archive. Sans `-z` pour les `.tar` non compressés. `-C` = dossier de destination.

Avec compression (`.tar.gz`) :

```
tar -xzvf backup.tar.gz -C /tmp/restore/
```

Sans compression (`.tar`) :

```
tar -xvf backup.tar -C /tmp/restore/
```

---

**`gzip -d fichier.gz`** Décompresse un fichier `.gz`.

```
gzip -d archive.gz
```

---

**`zip -r archive.zip dossier/`** Crée une archive ZIP. `-r` = récursif (inclut les sous-dossiers).

```
zip -r projet.zip ./mon_projet/
```

**`unzip archive.zip`** Extrait une archive ZIP.

Dossier courant :

```
unzip projet.zip
```

Vers un dossier spécifique :

```
unzip projet.zip -d /destination/
```

---

**`tail -n 10 fichier`** Affiche les 10 dernières lignes d'un fichier.

```
tail -n 10 /var/log/syslog
```

**`tail -f fichier`** Suit un fichier en temps réel — idéal pour surveiller des logs.

```
tail -f /var/log/nginx/access.log
```

---

**`du -sh dossier/`** Affiche la taille d'un dossier. `s` = somme totale, `h` = format lisible (Ko, Mo, Go).

Taille d'un dossier :

```
du -sh /home/user/
```

Taille de chaque dossier à la racine :

```
du -sh /*
```

---

**`chmod +x fichier`** Rend un fichier exécutable.

```
chmod +x script.sh
./script.sh
```

---

`>> fichier` Redirige et **ajoute** la sortie à la fin d'un fichier, sans écraser. `>` écrase, `>>` ajoute.

Ajouter à la fin :

```
echo "nouvelle ligne" >> journal.txt
ls -la >> liste_fichiers.txt
```

Différence :

```
echo "test" > fichier.txt
echo "test" >> fichier.txt
```

---

`find chemin -type d -iname "*mot*"` Recherche des **dossiers** dont le nom contient "mot". `-i` = insensible à la casse.

```
find /home -type d -iname "*projet*"
```

`find chemin -type f -iname "*mot*"` Recherche des **fichiers** dont le nom contient "mot".

Glob d'extension :

```
find / -type f -iname "*.conf"
```

Dans le dossier courant :

```
find . -type f -iname "*.log"
```

---

`grep -i "terme" fichier` Recherche un terme dans un fichier, insensible à la casse.

```
grep -i "error" /var/log/syslog
grep -i "failed" auth.log
```

`grep -ci "terme" fichier` **Compte** le nombre de lignes contenant le terme. Affiche uniquement un nombre.

```
grep -ci "error" /var/log/syslog
```

---

**`ls | wc -l`** Compte le nombre d'éléments dans le dossier courant.

```
ls | wc -l
```

---

**`shutdown -h now`** Éteint immédiatement la machine. `-h` = halt.

Éteindre maintenant :

```
shutdown -h now
```

Redémarrer maintenant :

```
shutdown -r now
```

Éteindre dans 10 minutes :

```
shutdown -h +10
```

---

**`uname -a`** Affiche les informations complètes sur le noyau et le système (architecture, version, hostname...).

```
uname -a
```

---

# □□ Réseau

**`scp -P port source@serveur:chemin destination`** Copie un fichier via SSH entre deux machines. `-P` pour spécifier le port SSH.

Télécharger depuis un serveur :

```
scp -P 22 user@192.168.1.10:/home/user/fichier.txt ./local/
```

Envoyer vers un serveur :

```
scp -P 22 ./local/fichier.txt user@192.168.1.10:/home/user/
```

---

**`nmap -P0 -sS -p port ip`** Scanner de ports. `-P0` = pas de ping, `-sS` = scan SYN (semi-furtif), `-p` = port(s) ciblé(s).

```
```

> ⚠ À utiliser uniquement sur des réseaux dont vous êtes responsable.

Ports spécifiques :

```
nmap -P0 -sS -p 80,443 192.168.1.1
```

Plage de ports :

```
nmap -P0 -sS -p 1-1000 192.168.1.1
```

Réseau entier :

```
nmap -P0 -sS 192.168.1.0/24
```

---

**`curl https://ipinfo.io/8.8.8.8`** Obtient des informations sur une adresse IP (localisation, FAI, ASN…).

```
curl https://ipinfo.io/8.8.8.8
```

---

**`ssh-keygen -t rsa`** Génère une paire de clés SSH (privée + publique).

- Crée `~/.ssh/id_rsa` — clé privée (ne jamais partager)
- Crée `~/.ssh/id_rsa.pub` — clé publique (à copier sur le serveur)

```
ssh-keygen -t rsa -b 4096 -C "mon@email.com"
```

**`ssh-copy-id user@serveur`** Copie la clé publique sur un serveur pour se connecter sans mot de passe.

Clé par défaut :

```
ssh-copy-id user@192.168.1.10
```

Clé spécifique :

```
ssh-copy-id -i ~/.ssh/ma_cle.pub user@192.168.1.10
```

---

**`eval "$(ssh-agent -s)"`** + **`ssh-add /chemin/cle`** Lance l'agent SSH et charge une clé privée. À ajouter dans `~/.zshrc`.

```
eval "$(ssh-agent -s)"
ssh-add /chemin/vers/ma_cle_privee
```

---

**Config SSH —** `~/.ssh/config` Fichier de raccourcis SSH : permet de se connecter par un alias simple.

```
Host monserveur
    HostName hostname.com
    User monuser
    IdentityFile /chemin/vers/ma_cle_privee
```

Connexion simplifiée :

```
ssh monserveur
```

---

**`python -m http.server 8080`** Lance un serveur HTTP minimal sur le port 8080, servant le dossier courant. Accessible sur `http://localhost:8080` ou `http://IP_machine:8080`.

```
cd /var/www/
python -m http.server 8080
```

---

**`openssl aes-256-cbc -a --in fichier --out fichier.enc`** Chiffre un fichier avec AES-256. `-a` = encodage base64.

```
openssl aes-256-cbc -a --in secret.txt --out secret.enc
```

**`openssl aes-256-cbc -d -a --in fichier.enc --out fichier`** Déchiffre un fichier. `-d` = decrypt.

Standard :

```
openssl aes-256-cbc -d -a --in secret.enc --out secret.txt
```

Variante avec md5 (pour clés générées via certains outils) :

```
openssl enc -aes-256-cbc -d -md md5 -in id_rsa.enc -out id_rsa
```

---

**`masscan -p 1-65535,U:1-65535 cible --rate=10000`** Scanner de ports très rapide sur tous les ports TCP et UDP.

> ⚠ À utiliser uniquement sur des réseaux dont vous êtes responsable.

```
masscan -p 1-65535,U:1-65535 192.168.1.0/24 --rate=10000
```

---

**`nmap --script=mysql-brute -p 3306 IP`** Tente une attaque par dictionnaire sur un service MySQL via NSE.

> ⚠ Usage légal uniquement (pentest autorisé, CTF, lab perso).

```
nmap --script=mysql-brute -p 3306 192.168.1.100
```

---

**Scan réseau + filtre port ouvert (boucle bash)** Détecte tous les hôtes actifs, puis filtre ceux avec le port 80 ouvert.

```
for ip in $(nmap -n -sn 192.168.1.0/24 -oG - | awk '/Up$/{print $2}'); do
  nmap -p 80 $ip | grep open > /dev/null
  if [[ $? -eq 0 ]]; then echo $ip; fi
done
```

---

# □□ Stockage

**`sudo dd if=image.iso of=/dev/diskX`** Copie bit-à-bit une image ISO sur un disque (clé USB bootable).

> ⚠ Vérifier **deux fois** le bon disque cible avant d'exécuter.

```
sudo dd if=/chemin/image.iso of=/dev/disk4 bs=4m status=progress
```

---

# □□ Linux uniquement

## □□ Informations système

**free -h** Affiche l'utilisation de la RAM et du swap. `-h` = format lisible (Go, Mo).

```
free -h
```

---

**lsblk** Liste tous les périphériques de blocs (disques, partitions, clés USB…) sous forme d'arborescence.

```
lsblk
```

---

**lscpu** Affiche les informations détaillées sur le processeur (architecture, cœurs, fréquence…).

```
lscpu
```

---

**htop** Moniteur de processus interactif. Raccourcis : `F9` = kill, `F6` = trier, `q` = quitter.

```
htop
```

---

**uptime -p** Affiche depuis combien de temps le système tourne. Ex : `up 2 days, 3 hours, 41 minutes`.

```
uptime -p
```

---

**lsb_release -a** Affiche la version de la distribution Linux (Distributor ID, Release, Codename…).

```
lsb_release -a
```

---

# □□ Réseau Linux

**ip addr add xx.xx.xx.xx/24 dev eth0** Assigne une adresse IP à une interface réseau (temporaire, perdu au redémarrage).

Assigner :

```
ip addr add 192.168.1.50/24 dev eth0
```

Vérifier :

```
ip addr show eth0
```

---

**`ip route add 192.168.30.0/24 via 192.168.7.1`** Ajoute une route statique.

Commande temporaire :

```
ip route add 192.168.30.0/24 via 192.168.7.1
```

Pour la rendre permanente — ajouter dans `/etc/network/interfaces` :

```
up ip route add 192.168.30.0/24 via 192.168.7.1
```

Vérifier la table de routage :

```
ip route show
```

---

**`sudo dhclient -r`** Libère puis renouvelle l'adresse IP DHCP.

Libérer l'IP (Windows : `ipconfig /release`) :

```
sudo dhclient -r
```

Demander une nouvelle IP (Windows : `ipconfig /renew`) :

```
sudo dhclient eth0
```

---

**`nano /etc/resolv.conf`** Modifie manuellement les serveurs DNS du système.

```
nameserver 192.168.1.1
nameserver 8.8.8.8
```

---

**Configuration réseau statique — `/etc/network/interfaces`** Assigne une IP fixe de façon permanente (Debian/Ubuntu).

```
auto eth0
iface eth0 inet static
    address 192.168.1.100
    netmask 255.255.255.0
    gateway 192.168.1.1
```

---

**`arp-scan --localnet`** / **`arp-scan -I eth0 -l`** Découvre tous les hôtes actifs sur le réseau local via requêtes ARP. Affiche IP + adresse MAC.

```
sudo arp-scan --localnet
sudo arp-scan -I eth0 -l
```

---

# □□ Stockage Linux

**`shred -n 5 -vz /dev/sda`** Efface définitivement un disque. `-n 5` = 5 passes aléatoires, `-z` = passe finale en zéros, `-v` = verbose.

> ⚠ Irréversible — vérifier le bon disque avec `lsblk` avant.

```
sudo shred -n 5 -vz /dev/sda
```

---

**`sudo hwclock --hctosys`** Synchronise l'horloge système depuis l'horloge matérielle (RTC/BIOS). `hc` = hardware clock → `sys` = system clock.

```
sudo hwclock --hctosys
```

---

# □□ Gestion des paquets (Debian/Ubuntu)

Routine complète de mise à jour :

```
sudo apt update
sudo apt upgrade
sudo apt dist-upgrade
sudo apt autoremove
sudo reboot
```

---

# □□ Utilisateurs & Permissions

**Ajouter un utilisateur au groupe sudo :**

```
apt install sudo
adduser nom_utilisateur sudo
reboot
```

**Fichier sudoers (`/etc/sudoers` — éditer via `visudo`) :**

```
# Donner tous les droits à root :
root ALL = (ALL) ALL

# Groupe admin sans mot de passe :
%admin ALL = (ALL) NOPASSWD: ALL
```

---

# □□ Bureau GNOME

**`sudo dconf reset -f /`** Réinitialise tous les paramètres GNOME (thème, raccourcis, extensions...) aux valeurs par défaut.

```
sudo dconf reset -f /
sudo reboot
```

---

# □□ curlFtpFs

Monte un serveur FTP comme un dossier local via FUSE.

Monter :

```
curlftpfs ftp://user:password@serveur.com /point/de/montage/
ls /point/de/montage/
```

Démonter :

```
fusermount -u /point/de/montage/
```

---

# □□ macOS uniquement

## □□ Sécurité & WiFi

**`security find-generic-password -wa "NomDuWifi"`** Affiche le mot de passe d'un réseau WiFi enregistré dans le Trousseau macOS. Demande le mot de passe admin.

```
security find-generic-password -wa "MonWifi"
```

---

# Gestion des disques — `diskutil`

**Créer une clé USB bootable :**

1. Identifier le disque cible :

```
diskutil list
```

2. Démonter :

```
diskutil unmountDisk /dev/disk2
```

3. Écrire l'image :

```
sudo dd if=/chemin/image.iso of=/dev/disk2 bs=1m
```

4. Éjecter :

```
diskutil eject /dev/disk2
```

**Formater une clé USB en exFAT :**

```
diskutil eraseDisk exFat USB /dev/disk2
```

---

# Réseau macOS

**`route -n get default`** Affiche la passerelle par défaut. `netstat -rn` pour la table de routage complète.

Passerelle par défaut :

```
route -n get default
```

Table de routage complète :

```
netstat -rn
```

---

**`sudo ifconfig en0 alias 192.168.7.99/24`** Ajoute une IP secondaire (alias) à l'interface `en0`.

Notation CIDR :

```
sudo ifconfig en0 alias 192.168.7.99/24
```

Notation avec netmask explicite :

```
sudo ifconfig en0 alias 192.168.8.245 netmask 255.255.255.0
```

**`sudo ifconfig en0 -alias 192.168.7.99`** Supprime un alias IP de l'interface `en0`.

```
sudo ifconfig en0 -alias 192.168.7.99
sudo ifconfig en0 -alias 192.168.8.245
```

---

# ⚡ Utilitaires macOS

**`caffeinate`** Empêche le Mac de se mettre en veille.

Jusqu'à `Ctrl+C` :

```
caffeinate
```

Pendant 1 heure :

```
caffeinate -t 3600
```

Pendant l'exécution d'un script :

```
caffeinate -i mon_script.sh
```

---

# □□ Windows uniquement

## □□ Réseau Windows

**`netsh wlan show profile name="NomWifi" key=clear`** Affiche les détails d'un profil WiFi, y compris le mot de passe en clair. Chercher la ligne "Contenu de la clé" (ou "Key Content").

```
netsh wlan show profile name="MonWifi" key=clear
```

**ipconfig /displaydns** Affiche le cache DNS local de Windows.

Afficher :

```
ipconfig /displaydns
```

Vider le cache :

```
ipconfig /flushdns
```

---

**ipconfig /release** / **ipconfig /renew** Libère et renouvelle l'adresse IP DHCP.

Libérer (Linux : `sudo dhclient -r`) :

```
ipconfig /release
```

Renouveler (Linux : `sudo dhclient`) :

```
ipconfig /renew
```

---

# ⚙ Système Windows

**shutdown /s /t 0** Éteindre/redémarrer Windows. `/s` = shutdown, `/r` = restart, `/t 0` = délai 0 s.

Éteindre :

```
shutdown /s /t 0
```

Redémarrer :

```
shutdown /r /t 0
```

Annuler un arrêt planifié :

```
shutdown /a
```

---

Revision #4
Created 31 May 2026 15:17:45 by Bibi
Updated 31 May 2026 21:06:42 by Bibi