

Les commandes du Dojo, chapitre par chapitre

Le pense-bête à garder sous la main. 21 chapitres (dont un bonus) + la page 📖 RHCSA · les commandes essentielles avec une explication courte.

Notation : `<f>` = un fichier · `<nom>` = à remplacer · `[..]` = optionnel · `|` = « ou ». Beaucoup de commandes système demandent `sudo` (droits admin).

CH 1 Les fondations

<code>pwd</code>	affiche le dossier courant (« où suis-je ? »)
<code>whoami</code>	le nom de l'utilisateur courant
<code>id</code>	identité complète : uid, gid, groupes
<code>ls</code>	liste le contenu d'un dossier
<code>ls -la</code>	liste tout, fichiers cachés + détails
<code>man <cmd></code>	le manuel d'une commande (q pour quitter)
<code><cmd> --help</code>	l'aide rapide d'une commande

CH 2 Navigation

<code>cd <dossier></code>	entre dans un dossier
<code>cd ..</code>	remonte d'un niveau
<code>cd ~</code>	retour au dossier personnel (ou <code>cd</code> seul)
<code>cd /</code>	va à la racine
<code>tree</code>	affiche l'arborescence
absolu vs relatif	<code>/home/user</code> (partout) vs <code>Documents</code> (depuis ici)

CH 3 Fichiers & dossiers

`cat <f>`

affiche le contenu d'un fichier

`mkdir <nom>`

crée un dossier (`-p` crée les parents)

`touch <f>`

crée un fichier vide

`cp source dest`

copie un fichier

`mv anc nouv`

déplace OU renomme

`rm <f>`

supprime (définitif !) · `rm -r` pour un dossier

jokers `*` `?`

`rm *.log` agit sur plusieurs fichiers

`find <où> -name`

retrouve des fichiers par nom, en profondeur

`ln -s cible lien`

crée un raccourci (lien symbolique)

`mkdir -p d/{a,b,c}`

plusieurs dossiers d'un coup (accolades)

CH 4 Utilisateurs & permissions

`ls -l`

lit les droits (ex. `-rwxr-xr--`)

`chmod +x <f>`

rend un fichier exécutable

`chmod 755 <f>`

droits en chiffres (r=4, w=2, x=1)

`chown user:grp <f>`

change le propriétaire (sudo)

`sudo <cmd>`

exécute en administrateur

`chgrp groupe <f>`

change le groupe d'un fichier

`umask 077`

droits par défaut des nouveaux fichiers

CH 5 Éditeurs (nano / vim)

`nano <f>`

éditeur simple (Ctrl+O enregistrer, Ctrl+X quitter)

`vim <f>`

éditeur à modes (`i` insérer, Échap, `:wq` enregistrer+quitter)

`echo "txt" > f`

écrit dans un fichier (écrase)

`echo "txt" >> f`

ajoute à la fin (sans écraser)

`cmd 2> erreurs.txt`

les erreurs à part

`cmd > f 2>&1`

sortie + erreurs dans un fichier

`cmd | tee f`

affiche ET enregistre

CH 6 Recherche & filtrage



`head / tail -n <f>`

le début / la fin d'un fichier

`tail -f <f>`

suit un journal en direct

`grep <motif> <f>`

cherche un texte (`-i -n -c -v -r`)

`cmd1 | cmd2`

le « pipe » : enchaîne les commandes

`wc -l`

compte les lignes

`cmd > f` · `>> f`

redirige la sortie vers un fichier

`less <f>`

lit page par page (q = quitter)

`tac <f>`

le fichier à l'envers

`locate <nom>`

retrouve un fichier (sudo updatedb)

`sed 's/a/b/g' <f>`

remplace du texte

`awk -F: '{print $1}' <f>`

extrait des colonnes

`... | xargs <cmd>`

passse une liste en arguments

CH 7 Processus système



`ps` · `ps aux`

liste les processus

`top`

activité en temps réel (q pour quitter)

`kill <PID>`

arrête un processus (`kill -9` force)

`cmd &` · `jobs`

arrière-plan · liste des tâches

`fg %1` · `bg %1`

au premier plan · en arrière-plan

`nohup cmd &`

survit à la déconnexion

`kill -15 / -9 <PID>`

poliment (TERM) / de force (KILL)

`pstree -p`

l'arbre des processus

`pgrep` · `pkill <nom>`

trouve / arrête par le nom

CH 8 Gestion des utilisateurs



```
cat /etc/passwd
```

la liste des comptes

```
sudo useradd -m <nom>
```

crée un utilisateur

```
sudo passwd <nom>
```

définit son mot de passe

```
groups
```

les groupes d'un utilisateur

```
sudo groupadd <nom>
```

crée un groupe

```
sudo usermod -aG grp user
```

ajoute un user à un groupe

```
sudo userdel -r <nom>
```

supprime un utilisateur

```
sudo -i · sudo su -
```

session root (exit pour revenir)

```
su - <n>
```

prend la session d'un autre compte

```
sudo -u <n> cmd
```

lance une commande en tant que <n>

```
sudo -l · sudo !!
```

tes droits sudo · refaire avec sudo

CH 9 Services & systemd



```
systemctl status <svc>
```

l'état d'un service

```
sudo systemctl start/stop
```

démarre / arrête un service

```
sudo systemctl enable
```

le lance au démarrage

fichiers `.service`

dans `/etc/systemd/system/` (ExecStart, After)

```
journalctl -u <svc> -n 20
```

le journal d'un service

CH 10 Déployer un site (nginx)



```
systemctl status nginx
```

vérifie que nginx tourne

```
/etc/nginx/
```

la config du serveur web

```
/var/www/html/
```

les fichiers du site

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

suit les visites en direct

```
curl http://localhost
```

teste le site en ligne de commande

CH 11 Réseau & SSH

```
ip a
```

adresse IP / config réseau

```
ping <hôte>
```

teste si une machine répond

```
cat /etc/resolv.conf
```

les serveurs DNS

```
ssh-keygen
```

génère une clé SSH (`~/.ssh/`)

```
ssh user@ip
```

connexion à une machine distante

```
ssh-keygen -t ed25519
```

crée ta paire de clés

```
ssh-copy-id user@hôte
```

installe ta clé sur le serveur

```
ssh user@hôte
```

se connecter (exit pour revenir)

```
scp f user@hôte:/dossier
```

copie par SSH

```
~/.ssh/config
```

surnoms de serveurs (Host ...)

CH 12 Sécurité & hardening

```
sudo ufw status
```

l'état du pare-feu

```
sudo ufw enable
```

active le pare-feu · `ufw allow <port>`

```
grep Failed .../auth.log
```

repère les tentatives SSH ratées

```
sudo fail2ban-client status sshd
```

fail2ban : les IP bannies de la prison SSH

```
/etc/fail2ban/jail.local
```

tes réglages (`maxretry` , `bantime`) · jamais jail.conf

```
fail2ban-client set sshd unbanip <IP>
```

débannit une IP bloquée par erreur

```
/etc/ssh/sshd_config
```

durcir la connexion SSH

```
sudo visudo
```

qui a le droit d'utiliser sudo

CH 13 Système & disque

<code>df -h</code>	l'espace disque disponible
<code>du -sh <dossier></code>	la taille d'un dossier
<code>lscpu</code>	les infos du processeur
<code>free -h</code>	la mémoire (regarde <code>available</code>)
<code>uname -a</code>	la version du système
<code>apt search · apt show</code>	chercher · détails d'un paquet
<code>sudo apt install / remove</code>	installer / désinstaller
<code>dpkg -l · dpkg -L <p></code>	paquets installés · leurs fichiers
<code>snap find · snap install</code>	les paquets snap (--classic)

CH 14 Monitoring système

<code>uptime</code>	depuis quand + charge système
<code>w · who</code>	qui est connecté
<code>last</code>	l'historique des connexions
<code>htop</code>	un <code>top</code> en couleur, plus lisible
<code>vmstat</code>	CPU / mémoire / disque en un coup d'œil

CH 15 Planification (cron)

<code>crontab -l</code>	liste tes tâches planifiées
<code>crontab -e</code>	édite : <code>min h jour mois j-sem cmd</code>
<code>echo '...' crontab -</code>	installe (⚠️ REMPLACE toute la crontab)
<code>crontab -l > f</code> puis <code>crontab f</code>	ajouter une tâche sans rien perdre
<code>*/15 * * * *</code> · <code>0 7 * * 1-5</code>	toutes les 15 min · 7 h en semaine
<code>systemctl list-timers</code>	les timers systemd (le cron moderne)
<code>date</code>	la date et l'heure du système
<code>/etc/anacrontab</code>	rattrape les tâches manquées
<code>echo cmd at now + 5 minutes</code>	une tâche unique (atq, atrm)

CH 16 Scripting Bash

```
#!/bin/bash
```

```
VAR=x · $VAR
```

```
./script.sh
```

```
echo $?
```

```
$(command)
```

```
read -p "? " var
```

```
$(( a + b ))
```

```
if [ -f f ]; then ...; fi
```

```
[ $a -gt 5 ] · [ -z "$v" ]
```

```
case $v in a) ...; esac
```

```
for i in {1..5}; do ...; done
```

```
while read l; do ...; done < f
```

```
$1 $2 · $# · "$@"
```

```
f() { ...; }
```

```
a && b · a || b
```

```
bash -x script.sh
```

le « shebang » : 1^{re} ligne d'un script

variables (ex. `echo $PATH`)

lance le script (après `chmod +x`)

le code de retour (0 = tout va bien)

substitution : insère le résultat d'une commande

demande une saisie

calcul entier

une condition

tests : nombres / texte

plusieurs cas

une boucle

ligne par ligne

les arguments du script

une fonction

si ça marche · sinon

déboguer (trace)

CH 17 Git — le versioning

```
git init
```

```
git status
```

```
git add <f>
```

```
git commit -m "msg"
```

```
git log
```

crée un dépôt dans le dossier

l'état des fichiers (ta boussole)

ajoute à l'index (`git add .` = tout)

enregistre un instantané

relit l'historique des commits

CH 18 Virtualisation (KVM)

```
egrep -c 'vmx|svm' /proc/cpuinfo
```

CPU compatible virtualisation ? (> 0 = oui)

```
sudo apt install qemu-kvm
```

installe l'hyperviseur

```
sudo systemctl enable --now libvirtd
```

démarre le démon de virtualisation

```
virsh list --all
```

liste les machines virtuelles

```
virsh start <vm>
```

démarre une VM

CH 19 Archives & compression

```
tar czf a.tar.gz <dossier>
```

crée une archive compressée

```
tar xzf a.tar.gz
```

extraît une archive

les lettres

c = créer, **x** = extraire, **z** = gzip, **f** = fichier

```
gzip f · gunzip f.gz
```

compresser · décompresser

```
zip -r a.zip d · unzip a.zip
```

archive .zip

CH 21 Proxy & reverse proxy

```
ss -tlnp
```

les ports en écoute et leur programme

```
proxy_pass <url>;
```

nginx relaie vers l'app, ex. `http://127.0.0.1:3000`

```
ln -s <cible> <lien>
```

active un site dans `sites-enabled/` (cible en chemin absolu !)

```
sudo nginx -t
```

teste la config avant de la charger

```
systemctl reload nginx
```

recharge sans couper les visiteurs (sudo)

```
proxy_set_header
```

transmet `Host`, `X-Real-IP`, `X-Forwarded-For` à l'app

```
upstream <nom> { }
```

répartit la charge entre instances (`server ip:port;`)

erreur 502

l'app derrière ne répond pas →

`/var/log/nginx/error.log`

```
curl -x <proxy> <url>
```

passse par un proxy sortant (Squid, port 3128)

```
export http_proxy=...
```

proxy pour tout le système (+ `https_proxy`)

`cat /etc/redhat-release``dnf install / search / info``rpm -q <paquet>``systemctl enable --now``-G wheel``firewall-cmd --add-service=http``--permanent` puis `--reload``getenforce` · `ls -Z``restorecon -Rv <chemin>``semanage fcontext -a -t``setsebool -P <booléen> on`

403 sur Red Hat

quelle distribution de la famille Red Hat

les paquets .rpm (équivalent d'apt)

est-il installé ? (`-qa` = tous)

démarré maintenant ET au boot (rien ne démarre seul)

le groupe des admins (au lieu de `sudo`)

ouvre le web (firewalld)

sinon la règle disparaît au rechargement

mode SELinux · étiquettes des fichiers

remet les bonnes étiquettes (après un `mv`)nouvelle règle d'étiquette, puis `restorecon`

active une permission SELinux prévue

droits Unix OK ? → `/var/log/audit/audit.log``nmcli device` · `ip a``nmcli con mod enp0s3 ipv4....``nmcli con up enp0s3``hostnamectl set-hostname``... | sudo tee -a /etc/hosts``groupadd -g` · `useradd -u -G``chage -M 90 -W 7` · `-d 0``useradd -r -s /sbin/nologin``chgrp` + `chmod 2770``/etc/sudoers.d/` · `visudo -c``systemctl set-default``journalctl -u sshd -p err``mkdir /var/log/journal``tuned-adm recommend / profile`

les cartes · les adresses actives

enregistre (`addresses` avec /24, `gateway`, `dns`, `method manual`)applique ; `+ipv4.dns` ajoute au lieu de remplacer nom d'hôte permanent (écrit /etc/hostname)écrire dans un fichier système (pas `sudo echo >>`)

GID / UID imposés, groupe secondaire

durée du mot de passe · changer au 1er login

compte de service, sans connexion

dossier partagé : le set-GID garde le groupe

une règle par fichier, en `chmod 440`, syntaxe vérifiée`multi-user.target` (texte) ou `graphical.target`

journaux d'un service · les erreurs

+ restart systemd-journald = journaux persistants

le profil de performance conseillé



```
parted /dev/sdb mklabel gpt
```

```
vgcreate -s 8M vg /dev/sdb1
```

```
lvextend -r -L 600M
```

```
mkswap · swapon -a
```

```
mount -a · findmnt --verify
```

```
/etc/exports · exportfs -rv
```

```
defaults,_netdev
```

```
/etc/auto.master.d/x.autofs
```

```
rd.break → chroot /sysroot
```

```
podman run -d -p 8080:8080 -v ~/web:/:Z
```

```
~/.config/containers/systemd/*.container
```

```
loginctl enable-linger kai
```

```
$# · "$@" · exit 1 · $?
```

```
crontab -e · at now + 1 minute
```

puis `mkpart` , `set 1 lvm on` (ou `fdisk` + `w`)

taille d'extent imposée ; `lvcreate -l 50` = 50 extents

agrandit le LV ET le système de fichiers, à chaud

swap permanent : ligne

```
UUID=... none swap defaults 0 0
```

tester fstab AVANT de redémarrer

partage NFS ; `showmount -e serveur`

montage NFS dans fstab (attend le réseau)

autofs ; carte joker `* -rw srv:/homes/&`

`passwd root` puis `touch /.autorelabel`

conteneur rootless, volume étiqueté SELinux

Quadlet : le conteneur devient un service utilisateur

ses services démarrent au boot, sans connexion

arguments, codes de sortie d'un script

tâche répétée · tâche unique (chemins complets !)

BONUS**Boîte à outils**`cal`

le calendrier du mois

`seq 1 10`

une suite de nombres

`echo "2+2" | bc`

le terminal comme calculatrice

`which <cmd>`

où se trouve une commande

`stat <f>`

infos détaillées d'un fichier

`file <f>`

le type d'un fichier

`sort` · `uniq`

trie / dédoublonne des lignes

`cut -d: -f1`

extraît une colonne

`tr a b`

remplace des caractères

`rev`

inverse chaque ligne

`factor 60`

décompose en facteurs premiers

`alias`tes raccourcis (ex. `ll` = `ls -la`)`cowsay txt`

la vache qui parle 🐮

`alias nom='cmd'`

ton propre raccourci

`source ~/.bashrc`

recharge ta config

`PS1='\u@\h:\w\$ '`

personnalise l'invite

`printenv` · `export V=x`

l'environnement