

The Dojo's commands, chapter by chapter

The reminder to keep at hand. 21 chapters (including a bonus one) + the 📖 RHCSA page · the essential commands with a short explanation.

Notation: `<f>` = a file · `<name>` = to replace · `[..]` = optional · `|` = "or". Many system commands require `sudo` (admin rights).

CH 1 The foundations

<code>pwd</code>	prints the current folder ("where am I?")
<code>whoami</code>	the current user's name
<code>id</code>	full identity: uid, gid, groups
<code>ls</code>	lists the content of a folder
<code>ls -la</code>	lists everything, hidden files + details
<code>man <cmd></code>	a command's manual (q to quit)
<code><cmd> --help</code>	a command's quick help

CH 2 Navigation

<code>cd <folder></code>	enters a folder
<code>cd ..</code>	goes up one level
<code>cd ~</code>	back to the home folder (or plain <code>cd</code>)
<code>cd /</code>	goes to the root
<code>tree</code>	displays the tree
absolute vs relative	<code>/home/user</code> (anywhere) vs <code>Documents</code> (from here)

CH 3 Files & folders

`cat <f>`

displays a file's content

`mkdir <name>`

creates a folder (`-p` creates the parents)

`touch <f>`

creates an empty file

`cp source dest`

copies a file

`mv old new`

moves OR renames

`rm <f>`

deletes (for good!) · `rm -r` for a folder

wildcards `*` `?`

`rm *.log` act on several files

`find <where> -name`

finds files by name, in depth

`ln -s target link`

creates a shortcut (symbolic link)

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

several folders at once (braces)

CH 4 Users & permissions

`ls -l`

reads the permissions (e.g. `-rwxr-xr--`)

`chmod +x <f>`

makes a file executable

`chmod 755 <f>`

permissions in digits (r=4, w=2, x=1)

`chown user:grp <f>`

changes the owner (sudo)

`sudo <cmd>`

runs as administrator

`chgrp group <f>`

changes a file's group

`umask 077`

default permissions of new files

CH 5 Editors (nano / vim)

`nano <f>`

simple editor (Ctrl+O save, Ctrl+X quit)

`vim <f>`

modal editor (`i` insert, Esc, `:wq` save+quit)

`echo "txt" > f`

writes into a file (overwrites)

`echo "txt" >> f`

appends at the end (without overwriting)

`cmd 2> errors.txt`

errors apart

`cmd > f 2>&1`

output + errors in one file

`cmd | tee f`

shows AND saves

CH 6 Search & filtering



`head / tail -n <f>`

the beginning / the end of a file

`tail -f <f>`

follows a log live

`grep <pattern> <f>`

searches for text (`-i -n -c -v -r`)

`cmd1 | cmd2`

the "pipe": chains commands

`wc -l`

counts lines

`cmd > f · >> f`

redirects the output into a file

`less <f>`

reads page by page (q = quit)

`tac <f>`

the file backwards

`locate <name>`

finds a file (sudo updatedb)

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

replaces text

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

extracts columns

`... | xargs <cmd>`

passes a list as arguments

CH 7 System processes



`ps · ps aux`

lists the processes

`top`

live activity (q to quit)

`kill <PID>`

stops a process (`kill -9` forces)

`cmd & · jobs`

background · list of jobs

`fg %1 · bg %1`

to the foreground · to the background

`nohup cmd &`

survives logging out

`kill -15 / -9 <PID>`

politely (TERM) / by force (KILL)

`pstree -p`

the process tree

`pgrep · pkill <name>`

finds / stops by name

CH 8 User management

<code>cat /etc/passwd</code>	the list of accounts
<code>sudo useradd -m <name></code>	creates a user
<code>sudo passwd <name></code>	sets their password
<code>groups</code>	a user's groups
<code>sudo groupadd <name></code>	creates a group
<code>sudo usermod -aG grp user</code>	adds a user to a group
<code>sudo userdel -r <name></code>	deletes a user
<code>sudo -i · sudo su -</code>	root session (exit to come back)
<code>su - <n></code>	takes another account's session
<code>sudo -u <n> cmd</code>	runs a command as <n>
<code>sudo -l · sudo !!</code>	your sudo rights · redo with sudo

CH 9 Services & systemd

<code>systemctl status <svc></code>	a service's status
<code>sudo systemctl start/stop</code>	starts / stops a service
<code>sudo systemctl enable</code>	launches it at boot
files <code>.service</code>	in <code>/etc/systemd/system/</code> (ExecStart, After)
<code>journalctl -u <svc> -n 20</code>	a service's journal

CH 10 Deploy a site (nginx)

<code>systemctl status nginx</code>	checks that nginx is running
<code>/etc/nginx/</code>	the web server's config
<code>/var/www/html/</code>	the site's files
<code>tail -f ../access.log</code>	follows the visits live
<code>curl http://localhost</code>	tests the site from the command line

CH 11 Network & SSH



```
ip a
```

IP address / network config

```
ping <host>
```

tests whether a machine answers

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

the DNS servers

```
ssh-keygen
```

generates an SSH key (`~/.ssh/`)

```
ssh user@ip
```

logs into a remote machine

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

creates your key pair

```
ssh-copy-id user@host
```

installs your key on the server

```
ssh user@host
```

logs in (exit to come back)

```
scp f user@host:/folder
```

copies over SSH

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

server nicknames (Host ...)

CH 12 Security & hardening



```
sudo ufw status
```

the firewall's status

```
sudo ufw enable
```

enables the firewall · `ufw allow <port>`

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

spots failed SSH attempts

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

fail2ban: the banned IPs of the SSH jail

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

your settings (`maxretry` , `bantime`) · never jail.conf

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

unbans an IP blocked by mistake

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

harden SSH access

```
sudo visudo
```

who is allowed to use sudo

CH 13 System & disk

`df -h`

the available disk space

`du -sh <folder>`

the size of a folder

`lscpu`

the processor's info

`free -h`

memory (look at `available`)

`uname -a`

the system version

`apt search · apt show`

search · package details

`sudo apt install / remove`

install / uninstall

`dpkg -l · dpkg -L <p>`

installed packages · their files

`snap find · snap install`

snap packages (--classic)

CH 14 System monitoring

`uptime`

since when + system load

`w · who`

who is logged in

`last`

the login history

`htop`

a colored, more readable `top`

`vmstat`

CPU / memory / disk at a glance

CH 15 Scheduling (cron)

`crontab -l`

lists your scheduled tasks

`crontab -e`

edits: `min h day month weekday cmd`

`echo '...' | crontab -`

installs (⚠ REPLACES the whole crontab)

`crontab -l > f` then `crontab f`

add a task without losing anything

`*/15 * * * * · 0 7 * * 1-5`

every 15 min · 7:00 on weekdays

`systemctl list-timers`

systemd timers (the modern cron)

`date`

the system's date and time

`/etc/anacrontab`

catches up missed jobs

`echo cmd | at now + 5 minutes`

a one-off job (atq, atrm)

CH 16 Bash scripting

`#!/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`

the "shebang": 1st line of a script

variables (e.g. `echo $PATH`)

runs the script (after `chmod +x`)

the return code (0 = all good)

substitution: inserts a command's result

asks for input

integer arithmetic

a condition

tests: numbers / text

several cases

a loop

line by line

the script's arguments

a function

if it works · otherwise

debug (trace)

CH 17 Git — version control

`git init`

`git status`

`git add <f>`

`git commit -m "msg"`

`git log`

creates a repository in the folder

the state of the files (your compass)

adds to the index (`git add .` = everything)

records a snapshot

reads the commit history

CH 18 Virtualization (KVM)

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

CPU supports virtualization? (> 0 = yes)

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

installs the hypervisor

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

starts the virtualization daemon

```
virsh list --all
```

lists the virtual machines

```
virsh start <vm>
```

starts a VM

CH 19 Archives & compression

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

creates a compressed archive

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

extracts an archive

the letters

c = create, **x** = extract, **z** = gzip, **f** = file

```
gzip f · gunzip f.gz
```

compress · decompress

```
zip -r a.zip d · unzip a.zip
```

.zip archive

CH 21 Proxy & reverse proxy

```
ss -tlnp
```

the listening ports and their program

```
proxy_pass <url>;
```

nginx relays to the app, e.g. `http://127.0.0.1:3000`

```
ln -s <target> <link>
```

enables a site in `sites-enabled/` (target as an absolute path!)

```
sudo nginx -t
```

tests the config before loading it

```
systemctl reload nginx
```

reloads without cutting the visitors off (sudo)

```
proxy_set_header
```

passes **Host**, **X-Real-IP**, **X-Forwarded-For** on to the app

```
upstream <name> { }
```

spreads the load across instances (`server ip:port;`)

502 error

the app behind doesn't answer →

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

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

goes through an outgoing proxy (Squid, port 3128)

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

proxy for the whole system (+ `https_proxy`)

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

403 on Red Hat

which distribution of the Red Hat family

.rpm packages (the equivalent of apt)

is it installed? (`-qa` = all)

starts now AND at boot (nothing starts by itself)

the admins group (instead of `sudo`)

opens the web (firewalld)

otherwise the rule disappears at the reload

SELinux mode · file labels

puts back the right labels (after an `mv`)new labeling rule, then `restorecon`

turns on a planned SELinux permission

Unix permissions 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`

the cards · the active addresses

saves (`addresses` with /24, `gateway` , `dns` ,
`method manual`)applies; `+ipv4.dns` adds instead of replacing
permanent hostname (writes /etc/hostname)write into a system file (not `sudo echo >>`)

imposed GID / UID, secondary group

password lifetime · change at 1st login

service account, no login

shared folder: the set-GID keeps the group

one rule per file, in `chmod 440` , syntax checked`multi-user.target` (text) or `graphical.target`

a service's logs · the errors

+ restart systemd-journald = persistent logs

the recommended performance profile



```
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
```

then `mkpart`, `set 1 lvm on` (or `fdisk` + `w`)

imposed extent size; `lvcreate -l 50` = 50 extents

grows the LV AND the filesystem, live

permanent swap: line

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

test fstab BEFORE rebooting

NFS share; `showmount -e server`

NFS mount in fstab (waits for the network)

autofs; wildcard map `* -rw srv:/homes/&`

```
passwd root then touch /.autorelabel
```

rootless container, SELinux-labeled volume

Quadlet: the container becomes a user service

their services start at boot, without a login

arguments, a script's exit codes

repeated task · one-off task (full paths!)

BONUS



Toolbox



`cal`

the month's calendar

`seq 1 10`

a sequence of numbers

`echo "2+2" | bc`

the terminal as a calculator

`which <cmd>`

where a command lives

`stat <f>`

detailed info on a file

`file <f>`

the type of a file

`sort` · `uniq`

sorts / deduplicates lines

`cut -d: -f1`

extracts a column

`tr a b`

replaces characters

`rev`

reverses each line

`factor 60`

breaks into prime factors

`alias`

your shortcuts (e.g. `ll` = `ls -la`)

`cowsay txt`

the talking cow 🐮

`alias name='cmd'`

your own shortcut

`source ~/.bashrc`

reloads your config

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

customises the prompt

`printenv` · `export V=x`

the environment