SkillNewton

Linux Commands Cheat Sheet

Essential Linux commands for files, permissions, users, processes, services, apt and dnf packages, networking, searching, archives and disk space.

> This cheat sheet covers the Linux commands that network and server administrators use daily, from moving around the file system to managing services, packages, users and remote connections. Commands work on Ubuntu and Debian as well as Fedora, RHEL, Rocky and AlmaLinux, with differences noted. Use it alongside a terminal on a lab machine or virtual machine, and test anything that deletes or overwrites files on a copy first.

How to use it

1. **Open a practice machine.** Use a virtual machine, WSL on Windows or a cloud lab rather than a production server while you learn.
2. **Find the section for your task.** Sections follow everyday jobs: files, permissions, processes, services, packages, networking and disk space.
3. **Read the manual for options.** Run `man command` or `command --help` to see every option for a command you use often.
4. **Check before deleting.** Run `ls`, `find` without `-delete`, or `rsync -n` first to see exactly what a destructive command will touch.

Navigation and files

`pwd`	Print the current folder (working directory).
`ls -lah`	List all files, including hidden ones, with permissions and human-readable sizes.
`cd /var/www`	Change to an absolute path.
`cd ..`	Go up one level. `cd ~` goes to your home folder, `cd -` to the previous folder.
`mkdir -p projects/site/css`	Create a folder, including any missing parent folders.
`touch notes.txt`	Create an empty file, or update the timestamp of an existing one.
`cp config.php config.php.bak`	Copy a file.
`cp -r site/ site-backup/`	Copy a folder and everything in it.
`mv old.txt new.txt`	Rename or move a file or folder.
`rm file.txt`	Delete a file. There is no recycle bin.
`rm -r old-folder/`	Delete a folder and its contents. Double-check the path; never run it as root on a path you have not checked.
`ln -s /var/www/site/current /var/www/html`	Create a symbolic link (shortcut) at the second path pointing to the first.

Viewing files

`cat file.txt`	Print a whole file. Best for short files.
`less /var/log/syslog`	Page through a file. Space for next page, / to search, q to quit.
`head -n 20 access.log`	Show the first 20 lines.
`tail -n 50 error.log`	Show the last 50 lines.
`tail -f /var/log/nginx/error.log`	Follow a file as new lines are written. Ctrl+C to stop.
`wc -l access.log`	Count lines in a file.
`diff -u old.conf new.conf`	Show the differences between two files.
`file backup.tar.gz`	Identify a file type from its contents.
`stat index.php`	Show size, permissions, owner and timestamps.

Permissions and ownership

`chmod 644 index.php`	Owner read/write, group and others read only. Typical for web files.
`chmod 755 deploy.sh`	Owner full access, group and others read and execute. Typical for scripts and folders.
`chmod 600 ~/.ssh/id_ed25519`	Owner read/write only. SSH refuses private keys with looser permissions.
`chmod u+x deploy.sh`	Symbolic mode: add execute for the owner (u).
`chmod g+w shared.txt`	Add write for the group (g).
`chmod o-rwx secrets.txt`	Remove all permissions for others (o).
`chmod -R u=rwX,go=rX site/`	Recursive: capital X adds execute only to folders and files that are already executable.
`chown priya file.txt`	Change the owner of a file (needs sudo for files you do not own).
`sudo chown -R www-data:www-data /var/www/site`	Change owner and group recursively, here to the web server user on Debian and Ubuntu.
`chgrp developers report.txt`	Change only the group.
`umask`	Show the default permission mask for new files (often 0022, giving 644 files and 755 folders).

```
r = 4   w = 2   x = 1      (add them for each of: owner, group, others)
7 = rwx   6 = rw-   5 = r-x   4 = r--   0 = ---

-rw-r--r--  644  owner read/write; group and others read
-rwxr-xr-x  755  owner full; group and others read and execute
-rw-------  600  owner only (SSH private keys, .env files)
d at the start of the line means a directory, l a symbolic link
```

Users and sudo

`whoami`	Show the user you are logged in as.

`id`	Show your user ID, group ID and every group you belong to.
`sudo apt update`	Run one command as root. You enter your own password.
`sudo -i`	Open a root shell. Type `exit` as soon as you are done.
`sudo adduser priya`	Create a user interactively with a home folder (Debian and Ubuntu).
`sudo useradd -m -s /bin/bash priya`	Create a user with a home folder and Bash shell (works on all distributions).
`sudo passwd priya`	Set or reset a user's password.
`sudo usermod -aG sudo priya`	Add a user to the sudo group on Debian and Ubuntu. On Fedora and RHEL the group is `wheel`. Keep -a or other groups are removed.
`sudo userdel -r priya`	Delete a user and their home folder.
`sudo visudo`	Edit sudo rules safely; it checks the syntax before saving.
`w`	Show who is logged in and what they are running.

Processes

`ps aux`	List every running process with its user, PID, CPU and memory use.
`ps aux | grep nginx`	Find processes whose command line contains a word.
`pgrep -a php-fpm`	List matching process IDs with their command lines.
`top`	Live view of processes, sorted by CPU. Press q to quit, M to sort by memory.
`htop`	Friendlier, colourful version of top. Install it with apt or dnf if missing.
`kill 1234`	Ask process 1234 to stop (SIGTERM). It can save work and exit cleanly.
`kill -9 1234`	Force-stop a process (SIGKILL). Last resort: no clean-up happens.
`pkill -f "php artisan queue"`	Stop processes whose full command line matches the text.
`nohup ./backup.sh &`	Run a command in the background and keep it running after you log out.
`jobs`	List background jobs in this shell; `fg` brings one back to the foreground.

Services and logs (systemd)

`systemctl status nginx`	Show whether a service is running, its PID and its latest log lines.
`sudo systemctl start nginx`	Start a service.
`sudo systemctl stop nginx`	Stop a service.
`sudo systemctl restart nginx`	Stop and start a service.
`sudo systemctl reload nginx`	Reload configuration without dropping connections (if the service supports it).

`sudo systemctl enable --now nginx`	Start a service now and at every boot.
`sudo systemctl disable nginx`	Stop a service from starting at boot.
`systemctl --failed`	List units that failed to start.
`journalctl -u nginx`	Show the logs for one service.
`journalctl -u nginx -f`	Follow a service's logs live.
`journalctl --since "1 hour ago"`	Show all logs from the last hour.
`journalctl -p err -b`	Show errors and worse since the last boot.

Packages: apt and dnf

`sudo apt update`	Debian/Ubuntu: refresh the list of available packages. Run before installing or upgrading.
`sudo apt upgrade`	Debian/Ubuntu: install available updates.
`sudo apt install nginx`	Debian/Ubuntu: install a package and its dependencies.
`sudo apt remove nginx`	Debian/Ubuntu: remove a package but keep its configuration files.
`sudo apt purge nginx`	Debian/Ubuntu: remove a package and its configuration files.
`sudo apt autoremove`	Debian/Ubuntu: remove dependencies that are no longer needed.
`apt search nginx`	Debian/Ubuntu: search package names and descriptions.
`apt show nginx`	Debian/Ubuntu: show details of a package.
`sudo dnf check-update`	Fedora/RHEL: list available updates.
`sudo dnf upgrade`	Fedora/RHEL: install available updates.
`sudo dnf install nginx`	Fedora/RHEL: install a package. `dnf search` and `dnf info` find and describe packages.
`sudo dnf remove nginx`	Fedora/RHEL: remove a package.

Networking and remote access

`ip a`	Show interfaces with their IPv4 and IPv6 addresses (short for `ip addr`).
`ip route`	Show the routing table; the default gateway is on the line starting `default via`.
`ss -tulpn`	List listening TCP and UDP ports with the owning process. Use sudo to see every process name.
`ping -c 4 example.com`	Send four echo requests to test reachability and latency.
`curl -I https://example.com`	Fetch only the HTTP response headers (status code, redirects, caching).
`curl -L -o file.zip https://example.com/file.zip`	Download a file, following redirects, and save it under a chosen name.

`dig example.com A +short`	Look up DNS records. Try `MX`, `TXT` or `@1.1.1.1` to query a specific server.
`ssh priya@203.0.113.10`	Log in to a remote server over SSH. Add `-p 2222` for a non-standard port.
`ssh-keygen -t ed25519`	Create an SSH key pair; `ssh-copy-id user@host` installs the public key on a server.
`scp report.pdf priya@203.0.113.10:/home/priya/`	Copy a file to a server over SSH. Use `-r` for folders.
`rsync -avz --delete site/ priya@203.0.113.10:/var/www/site/`	Sync a folder over SSH, sending only changes. --delete removes extra files at the destination; run with -n first for a dry run.

```
# rsync: the trailing slash on the source matters
rsync -av site/  backup/    # copies the CONTENTS of site into backup/
rsync -av site   backup/    # copies the folder itself, giving backup/site/
rsync -avn --delete site/ backup/   # -n = dry run: list changes, touch nothing
```

Searching

`find /var/www -name "*.php"`	Find files by name pattern under a folder.
`find . -type f -mtime -7`	Files changed in the last seven days.
`find / -type f -size +100M 2>/dev/null`	Files larger than 100 MB, hiding permission errors.
`find /tmp -name "*.tmp" -delete`	Delete matching files. Run without -delete first to check the list.
`find site/ -type d -exec chmod 755 {} \;`	Run a command on every match, here setting folder permissions.
`grep -rn "DB_PASSWORD" /var/www/site`	Search file contents recursively and show line numbers.
`grep -i "error" app.log`	Case-insensitive search.
`grep -v "^#" nginx.conf`	Show lines that do not match, here hiding comment lines.
`grep -rl "old-domain.com" .`	List only the names of files that contain the text.
`grep -E "404|500" access.log`	Extended regular expression: match either pattern.
`which php`	Show the full path of a command that would run.

Archives and disk space

`tar -czvf site-backup.tar.gz site/`	Create a gzip-compressed archive of a folder (c = create, z = gzip, f = file name).
`tar -xzvf site-backup.tar.gz`	Extract a .tar.gz archive into the current folder.
`tar -xzvf site-backup.tar.gz -C /tmp/restore`	Extract into another folder (it must already exist).
`tar -tzvf site-backup.tar.gz`	List the contents without extracting.
`zip -r site.zip site/`	Create a .zip archive of a folder.
`unzip site.zip`	Extract a .zip file. `unzip -l` lists contents first.
`df -h`	Free and used space on each mounted file system.

`df -i`	Inode use. A disk can be "full" with free space if it runs out of inodes.
`du -sh /var/log`	Total size of one folder.
`du -h --max-depth=1 /var | sort -h`	Size of each sub-folder, smallest to largest, to find what is filling a disk.
`lsblk`	List disks and partitions with sizes and mount points. `lsblk -f` adds file system types.
`free -h`	Show memory and swap use.

Online version, with the matching free guide: skillnewton.com/resources/cheat-sheets/linux-commands/
Learn this hands-on: Networking & Cybersecurity: Foundation & Practical Implementation – skillnewton.com/courses/networking-foundation-course/