🐧 Linux Command Cheat Sheet 2026

250+ Essential Commands for DevOps Engineers

By Tobias Weiss | tobias-weiss.org | @tobiasweiss_ai

> 💡 **Pro Tip**: Bookmark this cheat sheet. You'll reference it daily.

📖 Table of Contents

🗂️ File System Navigation

Command	Description	Example
`pwd`	Print working directory	`pwd`
`cd`	Change directory	`cd /var/www`
`cd ~`	Go to home directory	`cd ~`
`cd -`	Go to previous directory	`cd -`
`ls`	List files and directories	`ls -la`
`ls -l`	Long listing format	`ls -l`
`ls -a`	Show hidden files	`ls -a`
`ls -lh`	Human-readable sizes	`ls -lh`
`ls -R`	Recursive listing	`ls -R`
`tree`	Display directory tree	`tree -L 2`
`pushd`	Push directory to stack	`pushd /tmp`
`popd`	Pop directory from stack	`popd`

🔥 **Daily Use:** `ll` (alias for `ls -la`) | `cd ..` | `cd ~`

📄 File Operations

Command	Description	Example
`touch`	Create empty file	`touch file.txt`
`cat`	Display file content	`cat file.txt`

Command	Description	Example
`less`	View file page by page	`less file.txt`
`more`	View file page by page	`more file.txt`
`head`	Show first 10 lines	`head file.txt`
`head -n 20`	Show first 20 lines	`head -n 20 file.txt`
`tail`	Show last 10 lines	`tail file.txt`
`tail -f`	Follow file (live updates)	`tail -f /var/log/syslog`
`tail -n 20`	Show last 20 lines	`tail -n 20 file.txt`
`cp`	Copy file	`cp file.txt /backup/`
`cp -r`	Copy directory recursively	`cp -r dir/ /backup/`
`cp -i`	Interactive copy (confirm)	`cp -i file.txt /backup/`
`mv`	Move/rename file	`mv file.txt newfile.txt`
`mv -i`	Interactive move	`mv -i file.txt /backup/`
`rm`	Remove file	`rm file.txt`
`rm -r`	Remove directory recursively	`rm -r dir/`
`rm -i`	Interactive remove	`rm -i file.txt`
`rm -rf`	Force remove (DANGEROUS)	`rm -rf dir/`

🔥 **Daily Use:** `cat`, `less`, `cp -r`, `mv`, `rm -rf`

🔐 File Permissions

Permission Numbers

Number	Permission	Description
0	---	No permissions
1	--x	Execute only
2	-w-	Write only
3	-wx	Write + Execute
4	r--	Read only
5	r-x	Read + Execute
6	rw-	Read + Write
7	rwx	Read + Write + Execute

Permission Commands

Command	Description	Example
`chmod`	Change file permissions	`chmod 755 file.txt`
`chmod +x`	Add execute permission	`chmod +x script.sh`
`chmod -x`	Remove execute permission	`chmod -x script.sh`
`chmod -R`	Recursive permission change	`chmod -R 755 /var/www/`
`chown`	Change owner	`chown user:group file.txt`
`chown -R`	Recursive owner change	`chown -R user:group /var/www/`
`chgrp`	Change group	`chgrp www-data file.txt`
`umask`	Set default permissions	`umask 022`
`ls -l`	View permissions	`ls -l file.txt`

Common Permission Settings

Setting	Meaning	Use Case
`755`	rwxr-xr-x	Scripts, directories
`644`	rw-r--r--	Files

Setting	Meaning	Use Case
`600`	rw-------	Private files (SSH keys)
`700`	rwx------	Private directories
`640`	rw-r-----	Group-readable files
`750`	rwxr-x---	Group-writable directories

🔥 **Daily Use:** `chmod 755`, `chmod +x`, `chown user:group`

📝 Text Processing

Command	Description	Example
`grep`	Search for pattern	`grep "error" log.txt`
`grep -i`	Case-insensitive search	`grep -i "error" log.txt`
`grep -r`	Recursive search	`grep -r "error" /var/log/`
`grep -v`	Invert match	`grep -v "success" log.txt`
`grep -n`	Show line numbers	`grep -n "error" log.txt`
`grep -c`	Count matches	`grep -c "error" log.txt`
`grep -A 3`	Include 3 lines after	`grep -A 3 "error" log.txt`
`grep -B 3`	Include 3 lines before	`grep -B 3 "error" log.txt`
`grep -E`	Extended regex	`grep -E "error
`egrep`	Same as grep -E	`egrep "error
`awk`	Pattern scanning & processing	`awk '{print $1}' file.txt`
`awk -F`	Set field separator	`awk -F',' '{print $1}' file.csv`
`sed`	Stream editor	`sed 's/old/new/g' file.txt`
`sed -i`	Edit file in place	`sed -i 's/old/new/g' file.txt`
`cut`	Extract sections	`cut -d',' -f1 file.csv`
`sort`	Sort lines	`sort file.txt`
`sort -r`	Reverse sort	`sort -r file.txt`
`sort -n`	Numeric sort	`sort -n file.txt`
`sort -u`	Unique sort	`sort -u file.txt`
`uniq`	Remove duplicates	`uniq file.txt`
`uniq -c`	Count occurrences	`uniq -c file.txt`
`tr`	Translate/Delete characters	`tr 'a-z' 'A-Z'`
`tee`	Write to file & stdout	`command
`wc`	Count words/lines	`wc -l file.txt`
`wc -l`	Count lines	`wc -l file.txt`
`wc -w`	Count words	`wc -w file.txt`
`wc -c`	Count characters	`wc -c file.txt`

Common Patterns

```
# Find all files containing "error"
grep -rl "error" /var/log/

# Count occurrences of "error" in syslog
grep -c "error" /var/log/syslog

# Show lines 10-20 of a file
sed -n '10,20p' file.txt

# Replace all occurrences of "old" with "new"
sed -i 's/old/new/g' file.txt
```

```
# Extract column 1 from CSV
cut -d',' -f1 data.csv

# Show unique IPs from access log
awk '{print $1}' /var/log/nginx/access.log | sort | uniq -c | sort -nr
```

🔥 **Daily Use:** `grep`, `awk`, `sed`, `cut`, `sort`, `uniq`

⚙️ Process Management

Command	Description	Example
`ps`	List processes	`ps aux`
`ps aux`	Show all processes	`ps aux`
`ps -ef`	Full format listing	`ps -ef`
`top`	Interactive process viewer	`top`
`htop`	Improved top	`htop`
`pstree`	Show process tree	`pstree`
`pstree -p`	Show PIDs	`pstree -p`
`kill`	Terminate process	`kill 1234`
`kill -9`	Force terminate	`kill -9 1234`
`killall`	Kill all processes by name	`killall nginx`
`pkill`	Kill by name pattern	`pkill -f "python"`
`pgrep`	Find PIDs by name	`pgrep nginx`
`nice`	Set process priority	`nice -n 10 command`
`renice`	Change priority	`renice 10 1234`
`nohup`	Run immune to hangups	`nohup command &`
`disown`	Remove from shell jobs	`disown -h %1`
`jobs`	List background jobs	`jobs`
`fg`	Bring to foreground	`fg %1`
`bg`	Send to background	`bg %1`
`wait`	Wait for background jobs	`wait`

Process Signals

Signal	Number	Description	command
SIGHUP	1	Hangup	`kill -1 1234`
SIGINT	2	Interrupt (Ctrl+C)	`kill -2 1234`
SIGKILL	9	Force kill	`kill -9 1234`
SIGTERM	15	Termination	`kill 1234`
SIGSTOP	19	Stop process	`kill -19 1234`

🔥 **Daily Use:** `ps aux`, `top`, `kill`, `kill -9`, `nohup`

💻 System Information

Command	Description	Example
`uname`	System information	`uname -a`
`uname -a`	All information	`uname -a`
`uname -r`	Kernel release	`uname -r`
`uname -s`	Kernel name	`uname -s`
`uname -n`	Node name (hostname)	`uname -n`
`hostname`	Show/set hostname	`hostname`

Command	Description	Example
`hostnamectl`	Control hostname	`hostnamectl set-hostname server1`
`uptime`	System uptime	`uptime`
`w`	Who is logged in	`w`
`who`	Show logged in users	`who`
`whoami`	Current user	`whoami`
`id`	User/group IDs	`id`
`date`	Show/set date	`date`
`cal`	Calendar	`cal`
`cal 2026`	Year calendar	`cal 2026`
`timedatectl`	Time/date control	`timedatectl`
`free`	Memory usage	`free -h`
`free -h`	Human-readable	`free -h`
`vmstat`	Virtual memory stats	`vmstat 1`
`iostat`	CPU/I/O stats	`iostat`
`mpstat`	CPU stats	`mpstat -P ALL`
`sar`	System activity report	`sar -u 1 3`
`nproc`	CPU cores	`nproc`
`lscpu`	CPU information	`lscpu`
`lshw`	Hardware info	`sudo lshw`
`lspci`	PCI devices	`lspci`
`lsusb`	USB devices	`lsusb`
`lsblk`	Block devices	`lsblk`
`lsblk -f`	Filesystem info	`lsblk -f`
`df`	Disk space usage	`df -h`
`df -h`	Human-readable	`df -h`
`df -i`	Inode usage	`df -i`

🔥 **Daily Use:** `uname -a`, `uptime`, `free -h`, `df -h`, `lscpu`

🌐 Networking

Command	Description	Example
`ip`	IP configuration	`ip a`
`ip a`	Show all interfaces	`ip a`
`ip addr`	Same as ip a	`ip addr`
`ip route`	Show routing table	`ip route`
`ip link`	Show interfaces	`ip link`
`ip neighbor`	Show ARP cache	`ip neighbor`
`ifconfig`	Legacy network config	`ifconfig`
`ifconfig -a`	Show all interfaces	`ifconfig -a`
`ifup`	Bring interface up	`sudo ifup eth0`
`ifdown`	Bring interface down	`sudo ifdown eth0`
`ethtool`	Ethernet tool	`ethtool eth0`
`ping`	Test connectivity	`ping google.com`
`ping -c 4`	Ping 4 times	`ping -c 4 google.com`
`ping -i 2`	Ping every 2 seconds	`ping -i 2 google.com`
`ss`	Socket statistics	`ss -tulnp`
`ss -t`	TCP connections	`ss -t`
`ss -u`	UDP connections	`ss -u`
`ss -l`	Listening ports	`ss -l`

Command	Description	Example
`ss -tulnp`	All listening ports	`ss -tulnp`
`netstat`	Network statistics	`netstat -tulnp`
`netstat -t`	TCP connections	`netstat -t`
`route`	Routing table	`route -n`
`route -n`	Numeric output	`route -n`
`traceroute`	Trace route	`traceroute google.com`
`mtr`	My traceroute	`mtr google.com`
`nslookup`	DNS lookup	`nslookup google.com`
`dig`	DNS query	`dig google.com`
`dig MX`	Mail exchange records	`dig google.com MX`
`dig TXT`	TXT records	`dig google.com TXT`
`host`	DNS lookup	`host google.com`
`curl`	Transfer URL	`curl https://google.com`
`curl -O`	Save to file	`curl -O https://example.com/file.txt`
`curl -o`	Save with name	`curl -o output.txt https://example.com`
`curl -L`	Follow redirects	`curl -L https://google.com`
`curl -I`	Headers only	`curl -I https://google.com`
`wget`	Download files	`wget https://example.com/file.txt`
`wget -r`	Recursive download	`wget -r https://example.com/`
`wget -O`	Save as	`wget -O output.txt https://example.com`

🔥 **Daily Use:** `ip a`, `ss -tulnp`, `ping`, `curl`, `dig`

📦 Package Management

Debian/Ubuntu (APT)

Command	Description	Example
`apt update`	Update package lists	`sudo apt update`
`apt upgrade`	Upgrade packages	`sudo apt upgrade`
`apt full-upgrade`	Full upgrade	`sudo apt full-upgrade`
`apt install`	Install package	`sudo apt install nginx`
`apt remove`	Remove package	`sudo apt remove nginx`
`apt purge`	Remove + config	`sudo apt purge nginx`
`apt autoremove`	Remove unused	`sudo apt autoremove`
`apt clean`	Clean cache	`sudo apt clean`
`apt autoclean`	Auto clean	`sudo apt autoclean`
`apt search`	Search packages	`apt search nginx`
`apt show`	Show package info	`apt show nginx`
`apt list --installed`	List installed	`apt list --installed`
`apt-cache policy`	Show versions	`apt-cache policy nginx`
`dpkg -l`	List packages	`dpkg -l`
`dpkg -i`	Install .deb	`sudo dpkg -i package.deb`
`dpkg -r`	Remove package	`sudo dpkg -r nginx`
`dpkg -L`	List files	`dpkg -L nginx`
`dpkg -s`	Package status	`dpkg -s nginx`

Red Hat/CentOS (YUM/DNF)

Command	Description	Example
`dnf install`	Install package	`sudo dnf install nginx`

Command	Description	Example
`dnf remove`	Remove package	`sudo dnf remove nginx`
`dnf update`	Update all	`sudo dnf update`
`dnf upgrade`	Upgrade packages	`sudo dnf upgrade`
`dnf search`	Search packages	`dnf search nginx`
`dnf info`	Package info	`dnf info nginx`
`dnf list installed`	List installed	`dnf list installed`
`dnf clean`	Clean cache	`sudo dnf clean all`
`yum`	Legacy DNF	`sudo yum install nginx`

Snap

Command	Description	Example
`snap install`	Install snap	`sudo snap install lxd`
`snap remove`	Remove snap	`sudo snap remove lxd`
`snap list`	List snaps	`snap list`
`snap refresh`	Update snaps	`sudo snap refresh`
`snap find`	Search snaps	`snap find docker`

Flatpak

Command	Description	Example
`flatpak install`	Install flatpak	`flatpak install flathub org.gnome.Calculator`
`flatpak run`	Run flatpak	`flatpak run org.gnome.Calculator`
`flatpak list`	List flatpaks	`flatpak list`
`flatpak update`	Update all	`flatpak update`

🔥 **Daily Use:** `apt update && apt upgrade`, `apt install`, `dpkg -l`

💽 Disk & Storage

Command	Description	Example
`du`	Disk usage	`du -sh /var`
`du -sh`	Human-readable	`du -sh /var`
`du -h --max-depth=1`	Summary	`du -h --max-depth=1 /var`
`df`	Disk space	`df -h`
`fdisk`	Partition table	`sudo fdisk -l`
`fdisk /dev/sda`	Edit partition	`sudo fdisk /dev/sda`
`parted`	Partition editor	`sudo parted /dev/sda`
`mkfs`	Make filesystem	`sudo mkfs.ext4 /dev/sda1`
`mkfs.ext4`	Ext4 filesystem	`sudo mkfs.ext4 /dev/sda1`
`mkfs.xfs`	XFS filesystem	`sudo mkfs.xfs /dev/sda1`
`fsck`	Filesystem check	`sudo fsck /dev/sda1`
`mount`	Mount filesystem	`sudo mount /dev/sda1 /mnt`
`mount -a`	Mount all	`sudo mount -a`
`umount`	Unmount	`sudo umount /mnt`
`umount -l`	Lazy unmount	`sudo umount -l /mnt`
`blkid`	Block device IDs	`sudo blkid`
`lsblk`	List block devices	`lsblk`
`lsblk -f`	With filesystem	`lsblk -f`
`lvm`	LVM commands	`sudo pvdisplay, vgdisplay, lvdisplay`
`pvdisplay`	Physical volumes	`sudo pvdisplay`

Command	Description	Example
`vgdisplay`	Volume groups	`sudo vgdisplay`
`lvdisplay`	Logical volumes	`sudo lvdisplay`
`pvcreate`	Create PV	`sudo pvcreate /dev/sda1`
`vgcreate`	Create VG	`sudo vgcreate vg1 /dev/sda1`
`lvcreate`	Create LV	`sudo lvcreate -n lv1 -L 10G vg1`
`dmesg`	Kernel messages	`dmesg`
`dmesg	grep -i error`	Error messages	
`smartctl`	SMART info	`sudo smartctl -a /dev/sda`
`hdparm`	Disk performance	`sudo hdparm -Tt /dev/sda`
`badblocks`	Check bad blocks	`sudo badblocks /dev/sda1`
`dd`	Data duplication	`dd if=/dev/sda of=/dev/sdb`
`dd if=/dev/zero`	Create file	`dd if=/dev/zero of=file bs=1M count=100`
`shred`	Secure delete	`shred -u file.txt`

🔥 **Daily Use:** `df -h`, `du -sh`, `lsblk`, `mount`, `umount`

🗜️ Compression & Archives

Command	Description	Example
`gzip`	Compress	`gzip file.txt`
`gunzip`	Decompress	`gunzip file.txt.gz`
`zcat`	View compressed	`zcat file.txt.gz`
`bzip2`	Bzip2 compress	`bzip2 file.txt`
`bunzip2`	Decompress	`bunzip2 file.txt.bz2`
`xz`	XZ compress	`xz file.txt`
`unxz`	Decompress	`unxz file.txt.xz`
`tar`	Create archive	`tar -cvf archive.tar /path/`
`tar -xvf`	Extract	`tar -xvf archive.tar`
`tar -czvf`	Gzip tar	`tar -czvf archive.tar.gz /path/`
`tar -cjvf`	Bzip2 tar	`tar -cjvf archive.tar.bz2 /path/`
`tar -cJvf`	XZ tar	`tar -cJvf archive.tar.xz /path/`
`tar -tvf`	List contents	`tar -tvf archive.tar.gz`
`unzip`	Extract zip	`unzip archive.zip`
`zip`	Create zip	`zip archive.zip file.txt`
`7z`	7-Zip	`7z a archive.7z file.txt`
`7z x`	Extract 7z	`7z x archive.7z`

Common Examples

```
# Create gzipped tar archive
tar -czvf backup.tar.gz /var/www/

# Extract gzipped tar
tar -xzvf backup.tar.gz -C /restore/

# Compress directory with best compression
tar -cJvf backup.tar.xz /var/www/

# Create zip archive
zip -r backup.zip /var/www/
```

```
# Extract specific file from tar
tar -xzvf backup.tar.gz path/to/file.txt
```

🔥 **Daily Use:** `tar -czvf`, `tar -xzvf`, `unzip`

👥 User & Group Management

Command	Description	Example
`useradd`	Add user	`sudo useradd -m username`
`useradd -r`	Add system user	`sudo useradd -r -s /bin/false username`
`usermod`	Modify user	`sudo usermod -aG group username`
`userdel`	Delete user	`sudo userdel username`
`userdel -r`	Delete + home	`sudo userdel -r username`
`passwd`	Change password	`passwd`
`passwd username`	Change user password	`sudo passwd username`
`id`	Show user info	`id username`
`whoami`	Current user	`whoami`
`su`	Switch user	`su username`
`su -`	Switch with login	`su - username`
`sudo`	Run as root	`sudo command`
`sudo -i`	Interactive root	`sudo -i`
`sudo -u`	Run as user	`sudo -u username command`
`groupadd`	Add group	`sudo groupadd groupname`
`groupmod`	Modify group	`sudo groupmod -n newname groupname`
`groupdel`	Delete group	`sudo groupdel groupname`
`groups`	Show user groups	`groups username`
`grep user /etc/passwd`	Find user	`grep username /etc/passwd`
`getent passwd`	Query users	`getent passwd`
`vipw`	Edit users	`sudo vipw`
`vigr`	Edit groups	`sudo vigr`

🔥 **Daily Use:** `useradd`, `usermod -aG`, `passwd`, `sudo`, `groups`

🔑 SSH & Remote Access

Command	Description	Example
`ssh`	Connect to server	`ssh user@hostname`
`ssh -p`	Port specification	`ssh -p 2222 user@hostname`
`ssh -i`	Private key	`ssh -i ~/.ssh/id_rsa user@hostname`
`ssh -L`	Local tunnel	`ssh -L 8888:localhost:80 user@hostname`
`ssh -R`	Remote tunnel	`ssh -R 8888:localhost:80 user@hostname`
`ssh -D`	SOCKS proxy	`ssh -D 1080 user@hostname`
`scp`	Secure copy	`scp file.txt user@hostname:/path/`
`scp -r`	Recursive copy	`scp -r dir/ user@hostname:/path/`
`scp from remote`	Copy from server	`scp user@hostname:/path/file.txt .`
`sftp`	Secure FTP	`sftp user@hostname`
`ssh-keygen`	Generate key	`ssh-keygen -t ed25519`
`ssh-keygen -t rsa`	RSA key	`ssh-keygen -t rsa -b 4096`
`ssh-copy-id`	Copy public key	`ssh-copy-id user@hostname`
`ssh-add`	Add key to agent	`ssh-add ~/.ssh/id_rsa`
`ssh-agent`	Key agent	`eval $(ssh-agent)`

Command	Description	Example
`ssh-config`	SSH config	`~/.ssh/config`

SSH Config Example

```
Host myserver
    HostName server.example.com
    User myuser
    Port 22
    IdentityFile ~/.ssh/id_ed25519
    IdentitiesOnly yes
    ForwardAgent yes
    ServerAliveInterval 60
    ServerAliveCountMax 3
```

Common Examples

```
# Connect to server
ssh user@server.example.com

# Copy file to server
scp file.txt user@server.example.com:/remote/path/

# Copy directory from server
scp -r user@server.example.com:/remote/path/ ./local/path/

# Create SSH tunnel
ssh -L 3306:localhost:3306 user@server.example.com

# SOCKS proxy
ssh -D 1080 -C -N user@server.example.com

# Generate new SSH key
ssh-keygen -t ed25519 -C "your_email@example.com"
```

🔥 **Daily Use:** `ssh`, `scp`, `ssh-keygen`, `ssh-copy-id`

🛠️ Systemd & Services

Command	Description	Example
`systemctl`	Systemd control	`systemctl status nginx`
`systemctl start`	Start service	`sudo systemctl start nginx`
`systemctl stop`	Stop service	`sudo systemctl stop nginx`
`systemctl restart`	Restart service	`sudo systemctl restart nginx`
`systemctl reload`	Reload config	`sudo systemctl reload nginx`
`systemctl status`	Service status	`sudo systemctl status nginx`
`systemctl is-active`	Check if active	`systemctl is-active nginx`
`systemctl is-enabled`	Check if enabled	`systemctl is-enabled nginx`
`systemctl enable`	Enable at boot	`sudo systemctl enable nginx`
`systemctl disable`	Disable at boot	`sudo systemctl disable nginx`
`systemctl mask`	Prevent start	`sudo systemctl mask nginx`
`systemctl unmask`	Remove mask	`sudo systemctl unmask nginx`
`journalctl`	View logs	`journalctl -u nginx`
`journalctl -f`	Follow logs	`journalctl -f -u nginx`
`journalctl -xe`	Show errors	`sudo journalctl -xe`

Command	Description	Example
`journalctl --since`	Logs since time	`journalctl --since "1 hour ago"`
`journalctl --until`	Logs until time	`journalctl --until "2026-01-01 00:00:00"`
`systemctl list-units`	List units	`systemctl list-units --type=service`
`systemctl list-unit-files`	List service files	`systemctl list-unit-files`
`systemd-analyze`	Analyze boot	`systemd-analyze`
`systemd-analyze blame`	Slow services	`systemd-analyze blame`
`systemd-analyze critical-chain`	Boot chain	`systemd-analyze critical-chain`

Creating a Service

```
# /etc/systemd/system/myapp.service
[Unit]
Description=My Application
After=network.target

[Service]
User=myuser
Group=mygroup
WorkingDirectory=/opt/myapp
ExecStart=/opt/myapp/bin/start.sh
Restart=always
RestartSec=5
Environment=PORT=8080

[Install]
WantedBy=multi-user.target
```

Commands:

```
# Create service
sudo systemctl daemon-reload

# Start and enable
sudo systemctl enable myapp
sudo systemctl start myapp

# Check status
sudo systemctl status myapp
```

🔥 **Daily Use:** `systemctl status`, `systemctl start/stop/restart`, `journalctl -u`

📜 Logging

Command	Description	Example
`logger`	Write to syslog	`logger "Test message"`
`logger -p`	Priority	`logger -p user.info "Test message"`
`dmesg`	Kernel messages	`dmesg`
`dmesg -H`	Human-readable	`dmesg -H`
`dmesg --follow`	Follow kernel logs	`dmesg --follow`
`journalctl`	Systemd journal	`journalctl`
`journalctl -f`	Follow journal	`journalctl -f`
`journalctl -u`	Unit logs	`journalctl -u nginx`
`journalctl -xe`	Show errors	`sudo journalctl -xe`
`journalctl --since`	Since time	`journalctl --since "1 hour ago"`

Command	Description	Example
`journalctl --disk-usage`	Disk usage	`journalctl --disk-usage`
`journalctl --vacuum-time`	Clean old logs	`sudo journalctl --vacuum-time=2weeks`
`tail /var/log/syslog`	Syslog	`tail -f /var/log/syslog`
`tail /var/log/messages`	Messages	`tail -f /var/log/messages`
`tail /var/log/auth.log`	Auth logs	`tail -f /var/log/auth.log`
`tail /var/log/kern.log`	Kernel logs	`tail -f /var/log/kern.log`
`cat /var/log/nginx/access.log`	Nginx access	`cat /var/log/nginx/access.log`
`cat /var/log/nginx/error.log`	Nginx errors	`cat /var/log/nginx/error.log`

Rotate Logs

```
# Rotate logs manually
sudo logrotate -f /etc/logrotate.conf

# Force nginx rotation
sudo systemctl reload nginx
```

🔥 **Daily Use:** `journalctl -u`, `tail -f /var/log/`, `dmesg | grep -i error`

🔍 Find & Search

Command	Description	Example
`find`	Find files	`find /var -name "*.log"`
`find -name`	Find by name	`find /home -name "file.txt"`
`find -iname`	Case insensitive	`find /home -iname "file.TXT"`
`find -type f`	Find files	`find /var -type f -name "*.conf"`
`find -type d`	Find directories	`find /var -type d`
`find -mtime -7`	Modified < 7 days	`find /var -mtime -7`
`find -mtime +30`	Modified > 30 days	`find /var -mtime +30`
`find -size +10M`	Size > 10MB	`find /var -size +10M`
`find -size -1k`	Size < 1KB	`find /var -size -1k`
`find -exec`	Execute command	`find /tmp -type f -mtime +7 -exec rm {} \;`
`find -delete`	Delete files	`find /tmp -type f -mtime +7 -delete`
`find -user`	Find by user	`find /home -user username`
`find -group`	Find by group	`find /var -group www-data`
`grep`	Search in files	`grep "pattern" file.txt`
`grep -r`	Recursive search	`grep -r "pattern" /var/`
`grep -l`	Show filenames	`grep -rl "pattern" /var/`
`grep -i`	Case insensitive	`grep -i "error" file.txt`
`grep -v`	Invert match	`grep -v "success" file.txt`
`grep -n`	Show line numbers	`grep -n "pattern" file.txt`
`grep -c`	Count matches	`grep -c "pattern" file.txt`
`locate`	Fast file find	`locate file.txt`
`updatedb`	Update locate DB	`sudo updatedb`
`whereis`	Find binary/source/man	`whereis nginx`
`which`	Find executable	`which python3`

Common Examples

```
# Find all .conf files
find /etc -type f -name "*.conf"
```

```
# Find large files (>100MB)
find / -type f -size +100M -exec ls -lh {} \;

# Find modified files in last 24 hours
find /var -type f -mtime -1

# Find empty files
find /tmp -type f -empty

# Find files owned by user
find /home -type f -user username

# Recursive grep
grep -r "db_password" /etc/

# Case-insensitive recursive search
grep -ri "error" /var/log/
```

🔥 **Daily Use:** `find /var -name`, `grep -r`, `locate`

🌐 Network Tools

Command	Description	Example
`ping`	Test connectivity	`ping google.com`
`ping -c 4`	Ping 4 times	`ping -c 4 google.com`
`ping -i 2`	Ping every 2s	`ping -i 2 google.com`
`traceroute`	Trace route	`traceroute google.com`
`mtr`	My traceroute	`mtr google.com`
`nslookup`	DNS lookup	`nslookup google.com`
`dig`	DNS query	`dig google.com`
`dig MX`	Mail records	`dig google.com MX`
`dig ANY`	All records	`dig google.com ANY`
`host`	DNS lookup	`host google.com`
`whois`	Domain info	`whois example.com`
`curl`	Transfer URL	`curl https://google.com`
`curl -I`	Headers only	`curl -I https://google.com`
`curl -L`	Follow redirects	`curl -L https://google.com`
`curl -O`	Save to file	`curl -O https://example.com/file`
`curl -d`	POST data	`curl -d "param=value" URL`
`curl -H`	Custom header	`curl -H "Authorization: Bearer X" URL`
`wget`	Download files	`wget https://example.com/file`
`wget -r`	Recursive	`wget -r https://example.com/`
`wget -O`	Save as	`wget -O output.txt URL`
`nc`	Netcat	`nc -zv google.com 80`
`nc -l`	Listen	`nc -l -p 8080`
`nc -z`	Port scan	`nc -zv localhost 1-1000`
`nmap`	Network scan	`nmap -sS localhost`
`nmap -p`	Scan ports	`nmap -p 80,443 google.com`
`nmap -O`	OS detection	`nmap -O localhost`
`ss`	Socket stats	`ss -tulnp`
`ss -t`	TCP connections	`ss -t`
`ss -u`	UDP connections	`ss -u`

Command	Description	Example
`netstat`	Network stats	`netstat -tulnp`

Port Scanning

```
# Check if port is open
nc -zv localhost 22

# Scan ports 1-1000
nmap -p 1-1000 localhost

# Scan specific ports
nmap -p 80,443,8080 google.com

# Show listening ports
ss -tulnp | grep LISTEN
```

Download Files

```
# Download file
wget https://example.com/package.tar.gz

# Download with custom name
wget -O myfile.tar.gz https://example.com/package.tar.gz

# Resume download
wget -c https://example.com/large-file.iso

# Download via curl
curl -O https://example.com/file.txt
```

🔥 **Daily Use:** `ping` , `curl` , `dig` , `nslookup` , `ss -tulnp`

🔒 Security

Command	Description	Example
`passwd`	Change password	`passwd`
`chage`	Password expiry	`sudo chage -M 90 username`
`shadow`	Password info	`sudo grep user /etc/shadow`
`groups`	User groups	`groups username`
`sudo`	Root access	`sudo command`
`visudo`	Edit sudoers	`sudo visudo`
`su`	Switch user	`su - username`
`chmod`	Change permissions	`chmod 755 file`
`chown`	Change owner	`sudo chown user:group file`
`chgrp`	Change group	`sudo chgrp group file`
`umask`	Default permissions	`umask 022`
`ufw`	Firewall	`sudo ufw enable`
`ufw allow`	Allow port	`sudo ufw allow 22`
`ufw deny`	Deny port	`sudo ufw deny 80`
`ufw status`	Firewall status	`sudo ufw status numbered`
`iptables`	Firewall	`sudo iptables -L -n -v`
`iptables -A`	Add rule	`sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT`
`iptables -D`	Delete rule	`sudo iptables -D INPUT 1`

Command	Description	Example
`firewalld`	Firewall	`sudo firewall-cmd --state`
`fail2ban`	Brute force protection	`sudo systemctl status fail2ban`
`clushw`	CrowdSec	`sudo cscli decisions list`
`lynis`	Security audit	`sudo lynis audit system`
`rkhunter`	Rootkit hunter	`sudo rkhunter --check`
`chkrootkit`	Rootkit check	`sudo chkrootkit`
`openssl`	SSL/TLS	`openssl s_client -connect google.com:443`
`openssl speed`	Test performance	`openssl speed -multi all`
`gpg`	Encryption	`gpg --encrypt file.txt`
`ssh-keygen`	Generate key	`ssh-keygen -t ed25519`

Hardening Commands

```
# Disable root SSH
sudo sed -i 's/PermitRootLogin yes/PermitRootLogin no/' /etc/ssh/sshd_config
sudo systemctl restart sshd

# Set password policy
sudo chage -M 90 -m 7 -W 14 username

# Lock user account
sudo passwd -l username

# Check for open ports
ss -tulnp | grep LISTEN

# Check fail2ban
sudo tail -f /var/log/fail2ban.log
```

🔥 **Daily Use:** `chmod` , `chown` , `ufw` , `sudo` , `passwd`

⏰ Cron & Scheduling

Command	Description	Example
`crontab`	Cron table	`crontab -e`
`crontab -e`	Edit cron	`crontab -e`
`crontab -l`	List cron	`crontab -l`
`crontab -r`	Remove cron	`crontab -r`
`service cron`	Cron service	`sudo service cron status`
`systemctl cron`	Cron systemd	`sudo systemctl status cron`

Cron Format

```
# ┌──────────────── minute (0 - 59)
# │ ┌────────────── hour (0 - 23)
# │ │ ┌──────────── day of month (1 - 31)
# │ │ │ ┌────────── month (1 - 12)
# │ │ │ │ ┌──────── day of week (0 - 6) (Sunday=0 or 7)
# │ │ │ │ │
# │ │ │ │ │
# * * * * * user command to be executed
```

Examples

```
# Run every minute
* * * * * /path/to/script.sh

# Run every hour at minute 0
0 * * * * /path/to/script.sh

# Run daily at 2:30 AM
30 2 * * * /path/to/script.sh

# Run every Sunday at 3 AM
0 3 * * 0 /path/to/script.sh

# Run at startup
@reboot /path/to/script.sh

# Run every 5 minutes
*/5 * * * * /path/to/script.sh

# Run every 30 minutes
*/30 * * * * /path/to/script.sh
```

Special Commands

Command	Description	Example
`at`	Schedule one-time	`at 15:00`
`atq`	List at jobs	`atq`
`atrm`	Remove at job	`atrm 1`
`sleep`	Delay execution	`sleep 10; command`
`timeout`	Timeout command	`timeout 30 long-running-command`
`nohup`	Run after logout	`nohup command &`

🔥 **Daily Use:** `crontab -e`, `crontab -l`, `@reboot`

⚡ Shortcuts & Tricks

Keyboard Shortcuts

Shortcut	Description	Works In
`Ctrl+C`	Kill current command	Terminal
`Ctrl+Z`	Suspend command	Terminal
`Ctrl+D`	Exit / EOF	Terminal
`Ctrl+L`	Clear screen	Terminal
`Ctrl+R`	Search history	Terminal
`Ctrl+A`	Go to start of line	Terminal
`Ctrl+E`	Go to end of line	Terminal
`Ctrl+U`	Cut from start	Terminal
`Ctrl+K`	Cut to end	Terminal
`Ctrl+Y`	Paste (yank)	Terminal
`Ctrl+W`	Cut word	Terminal
`Alt+B`	Back one word	Terminal
`Alt+F`	Forward one word	Terminal
`Alt+D`	Delete word	Terminal
`Tab`	Auto-complete	Terminal
`Tab Tab`	Show completions	Terminal

Shortcut	Description	Works In
`Up Arrow`	Previous command	Terminal
`Down Arrow`	Next command	Terminal

History Commands

Command	Description	Example
`history`	Show history	`history`
`history 100`	Last 100 commands	`history 100`
`history -c`	Clear history	`history -c`
`!!`	Last command	`!!`
`!n`	Command number n	`!42`
`!string`	Last command with string	`!ls`
`!$`	Last argument	`ls -l !$`
`!*`	All arguments	`chmod 755 !*`
`!^`	First argument	`!^`
`!:$`	Last argument	`!:$`
`!:2`	Second argument	`!:2`
`sudo !!`	Run last as root	`sudo !!`

Examples

```
# Repeat last command
!!

# Repeat last command with sudo
sudo !!

# Run last command containing "nginx"
!nginx

# Use last argument from previous command
mkdir /tmp/test
cd !$

# Use all arguments from previous command
ls file1.txt file2.txt
chmod 755 !*
```

Aliases

Add to `~/.bashrc` or `~/.zshrc`:

```
# Navigation
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
alias ls='ls --color=auto'

# Git
alias gs='git status'
alias ga='git add'
alias gcm='git commit -m'
alias gco='git checkout'
alias gb='git branch'
alias gpush='git push'
alias gpull='git pull'
```

```
# Docker
alias dc='docker compose'
alias dps='docker ps -a'
alias dlogs='docker logs -f'
alias dexec='docker exec -it'

# System
alias update='sudo apt update && sudo apt upgrade -y'
alias clean='sudo apt clean && sudo apt autoclean'
alias mem='free -h'
alias disk='df -h'
alias ports='ss -tulnp'
alias processes='ps aux | grep'

# Reload shell
source ~/.bashrc
```

Common One-Liners

```
# Create directory and cd into it
mkdir -p /path/to/dir && cd $_

# Create a file with content
cat > file.txt <<EOF
Content here
EOF

# Append to file
cat >> file.txt <<EOF
More content
EOF

# Check CPU usage
top -bn1 | grep "Cpu(s)" | sed "s/.*, *\([0-9.]*\)%* id.*/\1/" | awk '{print 100 - $1"% CPU usage"}'

# Check memory usage
free -t -m | awk 'NR==2{printf "%.2f%", $3*100/$2 }'

# Check disk usage
df -h | awk '$NF=="/"{printf "%s", $5}'

# Check public IP
curl -s ifconfig.me

# Check open ports
ss -tulnp | grep LISTEN

# Find and kill process on port
sudo lsof -i :8080
sudo kill -9 $(sudo lsof -i :8080 -t)

# Count files in directory
find /path -type f | wc -l

# Create backup
tar -czvf backup-$(date +%Y-%m-%d).tar.gz /path/to/backup
```

Environment Variables

Command	Description	Example
`echo $VAR`	Show variable	`echo $PATH`
`export VAR=value`	Set variable	`export MY_VAR=hello`
`env`	Show all variables	`env`
`env	grep VAR`	Find variable	
`set`	Show all variables	`set`
`unset VAR`	Remove variable	`unset MY_VAR`
`source file`	Load variables	`source .env`

File Descriptors

Descriptor	Description	Use
0	STDIN	Input
1	STDOUT	Output
2	STDERR	Error

Examples:

```
# Redirect stdout to file
command > file.txt

# Redirect stderr to file
command 2> error.log

# Redirect both to file
command > output.log 2>&1

# Redirect both to different files
command > output.log 2> error.log

# Discard output
command > /dev/null 2>&1

# Pipe both stdout and stderr
command 2>&1 | grep "error"

# Send to both file and stdout (tee)
command | tee output.txt
```

📚 Final Tips

1. **RTFM**: `man command` or `command --help`
2. **Tab completion**: Use `Tab` to auto-complete
3. **History**: Use `Ctrl+R` to search command history
4. **Pipes**: Combine commands with `|`
5. **Redirect**: Save output with `>` and `>>`
6. **Background**: Run in background with `&`
7. **Superuser**: Use `sudo` when needed
8. **Check first**: Use `whatis`, `whereis`, `which`
9. **Practice**: The more you use these, the more natural they become
10. **Aliases**: Create shortcuts for commands you use often

🎯 Quick Reference Chart

Most Used Commands

Category	Command	Use Case
Navigation	`cd`, `ls`, `pwd`	Change directory, list files
File Ops	`touch`, `cp`, `mv`, `rm`	File operations
Text	`cat`, `less`, `grep`, `awk`, `sed`	Text processing
Process	`ps`, `top`, `htop`, `kill`	Process management
System	`uname`, `uptime`, `free`, `df`, `lscpu`	System info
Network	`ip`, `ping`, `curl`, `dig`, `ss`	Network diagnostics
Package	`apt`, `dnf`, `snap`, `flatpak`	Package management
Permissions	`chmod`, `chown`, `chgrp`	File permissions
Search	`find`, `grep`, `locate`	File searching
SSH	`ssh`, `scp`, `ssh-keygen`	Remote access

📖 Next Steps

1. **Print this cheat sheet** and keep it at your desk
2. **Practice daily**: Use these commands in your work
3. **Create aliases** for commands you use often
4. **Learn more**: Linux Command Line Book (Free)
5. **Test yourself**: LPIC-1 Free Practice Questions →

> **Remember**: The command line is powerful. With great power comes great responsibility. Always double-check before running destructive commands (`rm -rf` , `dd` , etc.).

Resources & Related Platforms

- **tobias-weiss.org** - More cheat sheets, articles and store products
- **graphwiz.ai** - AI, knowledge graphs and agentic development
- **courses.graphwiz.ai** - Interactive practice exams and study guides
- **ki-kompetenz-training.org** - EU AI Act compliant AI literacy training

tobias-weiss.org

DevOps Engineering & Linux Certification

More Resources

- → tobias-weiss.org — Articles, guides, and free resources
- → courses.graphwiz.ai — LPI practice exams & study tools
- → graphwiz.ai — AI infrastructure & knowledge graphs
- → ki-kompetenz-training.org — EU AI Act compliant AI literacy training

Generated: https://tobias-weiss.org