hostitworks.com
24 Free Web Hosting Tools

$ _
LINUX SERVER

Linux Server Admin Cheat Sheet

2026 Edition

Ubuntu · Debian · CentOS · RHEL · AlmaLinux

Essential commands for files, users, services, networking, disk, processes, logs, security, MySQL, Nginx & more.

250+	20	50+	30+
Commands Covered	Topic Sections	Config Snippets	Pages of Reference

Contents

How to Use This Cheat Sheet

Commands use Courier font in the left column. Angle brackets like <> mean replace with your actual value. Square brackets like [port] mean optional parameter. Commands marked # must be run as root or with sudo.

1. Navigation & File Management

Move around, find, copy, move, delete, archive

Directory Navigation

`pwd`	Print current working directory
`ls -la`	List all files including hidden, with permissions and sizes
`ls -lhS`	List sorted by size (largest first), human-readable sizes
`cd /var/www/html`	Change to absolute path
`cd ~`	Go to home directory
`cd -`	Return to previous directory
`tree -L 2`	Show directory tree 2 levels deep (install: apt install tree)

File Operations

`cp -r /src /dst`	Copy directory recursively to destination
`mv file.txt /new/path/`	Move or rename a file or directory
`rm -rf /path/to/dir`	Delete directory and all contents — irreversible, use carefully
`mkdir -p /path/to/new/dir`	Create directory and all parent directories
`touch newfile.txt`	Create empty file or update timestamp
`cat file.txt`	Print file contents to terminal
`less file.txt`	Page through large files (q to quit, /search to find)
`head -n 50 file.txt`	Show first 50 lines of a file
`tail -n 100 file.txt`	Show last 100 lines
`tail -f /var/log/syslog`	Follow a log file in real-time (Ctrl+C to stop)
`wc -l file.txt`	Count lines in a file
`diff file1 file2`	Show differences between two files

Search & Find

`find /var/www -name "*.php"`	Find all PHP files under /var/www
`find / -size +100M`	Find files larger than 100MB anywhere on system
`find . -mtime -7`	Files modified in the last 7 days
`find . -type f -empty`	Find all empty files
`grep -r "search_term" /path`	Recursively search for text string in all files under path
`grep -rn "ERROR" /var/log/`	Find ERROR in logs, showing line numbers
`grep -v "pattern" file`	Print lines NOT matching the pattern
`locate filename`	Fast file search using system database (updatedb to refresh)

Archives & Compression

`tar -czf archive.tar.gz /dir`	Create gzipped tar archive of a directory
`tar -xzf archive.tar.gz`	Extract gzipped tar archive in current directory
`tar -czf - /dir | gzip > f.gz`	Pipe tar directly to gzip for streaming
`zip -r archive.zip /dir`	Create zip archive of directory
`unzip archive.zip -d /output`	Extract zip to specific directory
`du -sh /var/www/*`	Show disk usage of each item under /var/www
`du -sh * | sort -rh | head`	Show top 10 largest items in current directory

2. File Permissions & Ownership

chmod, chown, umask, ACLs

Permissions (chmod)

`chmod 755 /var/www/html`	rwxr-xr-x — standard web directory permissions
`chmod 644 index.php`	rw-r--r-- — standard web file permissions
`chmod 600 ~/.ssh/id_rsa`	rw------- — private key must be this (SSH refuses otherwise)
`chmod -R 755 /var/www/html`	Apply 755 recursively to all directories
`chmod -R 644 /var/www/html`	Apply 644 recursively to all files (use with find for dirs+files)
`find /var/www -type d -exec chmod 755 {} +`	Set 755 only on directories
`find /var/www -type f -exec chmod 644 {} +`	Set 644 only on files

Ownership (chown)

`chown www-data:www-data file`	Set owner and group to www-data (Nginx/Apache user)
`chown -R www-data:www-data .`	Recursively set ownership on current directory
`chown user:group file`	Set specific user and group ownership
`ls -la`	View owner, group, permissions for all files
`id www-data`	Show UID/GID info for the www-data user
`stat filename`	Detailed file info including permissions in octal

Web Server Permission Standard

Web files: chmod 644, chown www-data:www-data Web dirs: chmod 755, chown www-data:www-data wp-config.php / .env: chmod 600, chown www-data:www-data Uploaded files: chmod 644, directories 755 (never 777)

3. User & Group Management

adduser, passwd, sudo, groups

Users

`adduser username`	Create new user with home directory and prompt for password
`useradd -m -s /bin/bash user`	Create user with home dir and bash shell
`passwd username`	Set or change user password
`usermod -aG sudo username`	Add user to sudo group (grants sudo access on Ubuntu/Debian)
`usermod -aG www-data user`	Add user to www-data group (web server group)
`userdel -r username`	Delete user and their home directory
`su - username`	Switch to another user account
`whoami`	Show current logged-in username
`id username`	Show user UID, GID, and all group memberships
`last`	Show login history for all users
`lastlog`	Show most recent login for every account
`w`	Show who is currently logged in and what they are doing

Groups & Sudo

`groups username`	List all groups a user belongs to
`groupadd groupname`	Create a new group
`gpasswd -a user group`	Add user to a group
`gpasswd -d user group`	Remove user from a group
`visudo`	Safely edit the sudoers file (use this, not direct edit)
`sudo -l`	List sudo privileges for current user
`sudo -i`	Open interactive root shell via sudo

4. Process Management

ps, top, htop, kill, nice, jobs

Viewing Processes

`ps aux`	List all running processes with CPU and memory usage
`ps aux | grep nginx`	Filter process list for a specific service
`top`	Interactive real-time process monitor (q to quit)
`htop`	Better interactive monitor — color, mouse support (apt install htop)
`pgrep nginx`	Find PID of a process by name
`pstree`	Show process tree showing parent-child relationships
`lsof -i :80`	Show what process is listening on port 80
`lsof -p PID`	Show all files opened by a specific process
`strace -p PID`	Trace system calls of a running process (debug tool)

Managing Processes

`kill PID`	Send SIGTERM (graceful stop) to process by PID
`kill -9 PID`	Send SIGKILL (force stop) — use when normal kill fails
`killall nginx`	Kill all processes with the given name
`pkill -f "php-fpm"`	Kill processes matching pattern in full command line
`nice -n 10 command`	Run command with lower priority (10 = lower, -20 = highest)
`renice -n 5 -p PID`	Change priority of running process
`nohup command &`	Run command immune to hangup, in background
`command &`	Run command in background
`jobs`	List background jobs in current shell
`fg %1`	Bring job 1 to foreground
`Ctrl+Z then bg`	Suspend foreground job then resume it in background

5. Service Management — systemd

start, stop, enable, status, journalctl

Service Control

`systemctl start nginx`	Start the nginx service immediately
`systemctl stop nginx`	Stop the nginx service
`systemctl restart nginx`	Stop and start (brief downtime)
`systemctl reload nginx`	Reload config without downtime (when supported)
`systemctl status nginx`	Show current status, recent log entries, PID
`systemctl enable nginx`	Auto-start service on boot
`systemctl disable nginx`	Remove from autostart
`systemctl is-active nginx`	Check if service is currently running
`systemctl is-enabled nginx`	Check if service is set to start on boot
`systemctl list-units --failed`	List all failed services
`systemctl daemon-reload`	Reload systemd after editing unit files

Journals & Logs (journalctl)

`journalctl -u nginx`	Show all logs for the nginx unit
`journalctl -u nginx -f`	Follow nginx logs in real-time
`journalctl -u nginx --since "1h ago"`	Show nginx logs from the last hour
`journalctl -p err`	Show only error-level and above log entries
`journalctl --disk-usage`	Show how much disk space journals are using
`journalctl --vacuum-size=500M`	Reduce journal storage to 500MB
`journalctl -b`	Show logs from current boot only
`journalctl -b -1`	Show logs from previous boot

6. Disk & Storage

df, du, lsblk, fdisk, mount, LVM basics

Disk Usage

`df -h`	Show disk space usage for all mounted filesystems
`df -h /var`	Show disk space for specific directory/mount
`du -sh /var/log`	Show total size of /var/log directory
`du -sh /var/www/* | sort -rh`	Sort websites by disk usage, largest first
`du -sh * | sort -rh | head -20`	Top 20 largest items in current directory
`ncdu /`	Interactive disk usage explorer (apt install ncdu)
`find / -size +500M 2>/dev/null`	Find files larger than 500MB on entire filesystem

Block Devices & Partitions

`lsblk`	List all block devices and their mount points
`lsblk -f`	Include filesystem type and UUID information
`blkid`	Show UUIDs and types for all block devices
`fdisk -l`	List all disks and their partitions
`parted -l`	Alternative partition viewer with more detail
`mount | grep /dev/sd`	Show currently mounted physical devices
`umount /mnt/data`	Unmount a filesystem

Filesystem & LVM

`mkfs.ext4 /dev/sdb1`	Format partition as ext4 (destroys all data)
`e2fsck -f /dev/sdb1`	Check and repair ext4 filesystem (unmounted only)
`resize2fs /dev/sdb1`	Resize ext4 filesystem after expanding partition/LV
`lvdisplay`	Show all LVM logical volumes
`vgdisplay`	Show LVM volume group info and free space
`lvextend -L +10G /dev/vg/lv`	Extend LV by 10GB then run resize2fs

7. Networking & Firewall

ip, ss, netstat, UFW, iptables, dig, curl

Network Interfaces & IPs

`ip addr show`	Show all network interfaces and IP addresses
`ip addr show eth0`	Show details for specific interface
`ip route show`	Show routing table
`ip link set eth0 up`	Bring network interface up
`hostnamectl`	Show and set system hostname
`hostname -I`	Show all IP addresses assigned to this machine
`curl ifconfig.me`	Show your public-facing IP address
`curl -s ipinfo.io/json`	Public IP with location info in JSON

Ports & Connections

`ss -tlnp`	Show all TCP listening ports with process names
`ss -tlnp | grep :80`	Check what is listening on port 80
`netstat -tlnp`	Alternative to ss (install net-tools if missing)
`lsof -i :443`	Show process using port 443
`ss -s`	Show summary of all connections
`nmap -sV localhost`	Port scan with service version detection

UFW Firewall (Ubuntu/Debian)

`ufw status verbose`	Show firewall status and all rules
`ufw enable`	Enable the firewall
`ufw allow 22/tcp`	Allow SSH (do this BEFORE enabling ufw)
`ufw allow 80/tcp`	Allow HTTP
`ufw allow 443/tcp`	Allow HTTPS
`ufw allow from 1.2.3.4`	Allow all traffic from a specific IP
`ufw deny from 1.2.3.4`	Block all traffic from a specific IP
`ufw delete allow 8080`	Remove a firewall rule
`ufw logging on`	Enable UFW logging to /var/log/ufw.log
`ufw reset`	Reset all rules to defaults (disables ufw)

DNS & HTTP Testing

`dig example.com A`	Query A record for a domain
`dig example.com MX`	Query MX records
`dig @8.8.8.8 example.com`	Query using Google DNS (bypass local cache)
`nslookup example.com`	Alternative DNS lookup tool
`curl -I https://example.com`	Fetch HTTP headers only (check status code, headers)
`curl -L -o /dev/null -s -w "%{http_code}" https://example.com`	Get HTTP status code only
`ping -c 5 example.com`	Send 5 ICMP packets and show round-trip times
`traceroute example.com`	Trace network path to destination
`mtr example.com`	Real-time traceroute with statistics
`wget -q -O- https://example.com | head`	Fetch URL and show first part of response

8. SSH & Key Management

Connect, keygen, copy-id, tunnels, config

Connecting & Keys

`ssh user@hostname`	Connect to server as user
`ssh -p 2222 user@host`	Connect on non-standard port
`ssh -i ~/.ssh/mykey user@host`	Connect using specific private key
`ssh-keygen -t ed25519 -C "email@example.com"`	Generate modern Ed25519 key pair
`ssh-keygen -t rsa -b 4096`	Generate RSA 4096-bit key pair (legacy)
`ssh-copy-id user@hostname`	Copy public key to server authorized_keys
`cat ~/.ssh/id_ed25519.pub | ssh user@host "cat >> ~/.ssh/authorized_keys"`	Manual key copy
`ssh-add ~/.ssh/mykey`	Add key to SSH agent for session

SSH Config & Hardening

`/etc/ssh/sshd_config`	Main SSH server configuration file
`PermitRootLogin no`	Disable direct root login — always set this
`PasswordAuthentication no`	Disable password auth — enforce key-only login
`Port 2222`	Change SSH port (security by obscurity — use with firewall)
`AllowUsers username`	Whitelist specific users who can SSH in
`MaxAuthTries 3`	Limit auth attempts before disconnect
`systemctl restart sshd`	Apply sshd_config changes
`ssh -N -L 3306:localhost:3306 user@host`	Tunnel local port 3306 to remote MySQL
`ssh -N -R 8080:localhost:80 user@host`	Reverse tunnel — expose local port remotely

~/.ssh/config Example — Multiple Server Shortcuts

```
Host prod HostName 203.0.113.10 User deploy IdentityFile ~/.ssh/prod_key Port 22 Host staging HostName 203.0.113.11 User deploy IdentityFile ~/.ssh/staging_key # Usage: ssh prod (instead of ssh -i ~/.ssh/prod_key deploy@203.0.113.10)
```

9. Package Management

apt (Ubuntu/Debian) and dnf/yum (RHEL/CentOS/AlmaLinux)

apt — Ubuntu & Debian

`apt update`	Refresh package list from repositories (run before install)
`apt upgrade -y`	Upgrade all installed packages to latest versions
`apt full-upgrade -y`	Upgrade including handling dependency changes
`apt install nginx`	Install a package
`apt install -y nginx php-fpm`	Install multiple packages, auto-confirm
`apt remove nginx`	Remove package but keep config files
`apt purge nginx`	Remove package and all its config files
`apt autoremove`	Remove unused dependency packages
`apt search nginx`	Search for packages by name
`apt show nginx`	Show package details, version, dependencies
`dpkg -l | grep nginx`	List installed packages matching a name
`dpkg -L nginx`	List all files installed by a package
`apt-cache policy nginx`	Show available versions and current installation

dnf / yum — RHEL, CentOS, AlmaLinux, Rocky

`dnf update -y`	Update all packages
`dnf install nginx -y`	Install a package
`dnf remove nginx`	Remove a package
`dnf search nginx`	Search available packages
`dnf info nginx`	Show package information
`rpm -qa | grep nginx`	List installed RPM packages matching name
`dnf list installed`	List all installed packages
`dnf history`	Show package transaction history

10. Nginx Web Server

Config, virtual hosts, SSL, logs, testing

Nginx Commands

`nginx -t`	Test nginx config for syntax errors — run before reload
`nginx -T`	Test and print complete final config
`systemctl reload nginx`	Gracefully reload config (zero downtime)
`systemctl restart nginx`	Full restart (brief downtime)
`nginx -s reopen`	Reopen log files (after logrotate)
`tail -f /var/log/nginx/access.log`	Follow access log in real-time
`tail -f /var/log/nginx/error.log`	Follow error log in real-time
`cat /var/log/nginx/access.log | awk '{print $1}' | sort | uniq -c | sort -rn | head`	Top IPs by request count

Nginx Server Block — WordPress with PHP-FPM

```
server { listen 80; server_name example.com www.example.com; return 301 https://$host$request_uri; } server { listen 443 ssl http2; server_name example.com www.example.com; root /var/www/example.com/public; index index.php; ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem; ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem; location / { try_files $uri $uri/ /index.php?$args; } location ~ \.php$ { fastcgi_pass unix:/run/php/php8.2-fpm.sock; include fastcgi_params; fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name; } location ~* \.(css|js|jpg|png|webp|woff2)$ { expires 1y; } }
```

11. MySQL / MariaDB

Connect, databases, users, backup, restore, optimization

MySQL CLI

`mysql -u root -p`	Connect to MySQL as root
`mysql -u user -p dbname`	Connect to specific database as user
`mysql -u root -p -e "SHOW DATABASES;"`	Run single query without interactive mode
`SHOW DATABASES;`	List all databases
`USE dbname;`	Switch to a database
`SHOW TABLES;`	List tables in current database
`DESCRIBE tablename;`	Show table structure and column types
`SHOW PROCESSLIST;`	Show active queries — find slow/stuck queries
`SHOW STATUS LIKE "Innodb_buffer_pool%";`	Check InnoDB buffer pool usage
`SELECT VERSION();`	Show MySQL/MariaDB version

Database & User Management

`CREATE DATABASE mydb CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;`	Create database with UTF-8
`DROP DATABASE mydb;`	Delete database permanently — no recovery
`CREATE USER 'user'@'localhost' IDENTIFIED BY 'password';`	Create MySQL user
`GRANT ALL PRIVILEGES ON mydb.* TO 'user'@'localhost';`	Grant full access to database
`GRANT SELECT,INSERT ON db.* TO 'ro'@'%';`	Grant read+write from any host
`FLUSH PRIVILEGES;`	Apply privilege changes immediately
`SHOW GRANTS FOR 'user'@'localhost';`	Show all grants for a user
`DROP USER 'user'@'localhost';`	Delete a MySQL user

Backup & Restore

`mysqldump -u root -p dbname > backup.sql`	Dump single database to SQL file
`mysqldump -u root -p --all-databases > all.sql`	Dump all databases
`mysqldump -u root -p --single-transaction dbname > safe.sql`	Live backup without table locks (InnoDB)
`mysql -u root -p dbname < backup.sql`	Restore a database from SQL dump
`mysqldump -u root -p dbname | gzip > backup.sql.gz`	Compress backup on the fly
`zcat backup.sql.gz | mysql -u root -p dbname`	Restore from gzipped backup

12. SSL Certificates — Certbot / Let's Encrypt

Issue, renew, auto-renew, troubleshoot

Certbot Commands

`apt install certbot python3-certbot-nginx`	Install Certbot with Nginx plugin (Ubuntu/Debian)
`certbot --nginx -d example.com -d www.example.com`	Issue cert and auto-configure Nginx
`certbot certonly --nginx -d example.com`	Issue cert only — do not modify Nginx config
`certbot certonly --standalone -d example.com`	Issue using built-in server (stop Nginx first)
`certbot certificates`	List all installed certificates and expiry dates
`certbot renew --dry-run`	Test renewal process without actually renewing
`certbot renew`	Renew all certificates that expire within 30 days
`certbot renew --force-renewal`	Force renew even if not near expiry
`certbot delete --cert-name example.com`	Remove a certificate

Auto-Renewal via Systemd Timer

```
# Certbot installs a systemd timer automatically — verify it: systemctl status certbot.timer systemctl list-timers certbot* # If timer is missing, add cron manually: 0 3 * * * certbot renew --quiet --post-hook "systemctl reload nginx"
```

13. Log Management

Key log locations, analysis, logrotate, grep tricks

Key Log Locations

`/var/log/syslog`	General system messages (Ubuntu/Debian)
`/var/log/auth.log`	Authentication attempts, SSH logins, sudo usage
`/var/log/nginx/access.log`	All HTTP requests received by Nginx
`/var/log/nginx/error.log`	Nginx errors, PHP errors (if logged here)
`/var/log/php8.2-fpm.log`	PHP-FPM pool errors and crashes
`/var/log/mysql/error.log`	MySQL startup errors, crashes, replication issues
`/var/log/mail.log`	Email sending/receiving, SMTP authentication
`/var/log/fail2ban.log`	IPs blocked by fail2ban for brute force attempts
`/var/log/ufw.log`	Firewall blocked connection attempts

Log Analysis One-Liners

`grep "FAILED PASSWORD" /var/log/auth.log | awk '{print $11}' | sort | uniq -c | sort -rn | head`	Top IPs with failed SSH logins
`awk '{print $1}' /var/log/nginx/access.log | sort | uniq -c | sort -rn | head -20`	Top 20 IPs by request count
`awk '$9 == 500' /var/log/nginx/access.log | head -20`	Show 500 error requests in Nginx log
`grep "wp-login.php" /var/log/nginx/access.log | wc -l`	Count WordPress login attempts
`awk '{print $7}' access.log | sort | uniq -c | sort -rn | head -20`	Top 20 requested URLs
`tail -f /var/log/nginx/error.log | grep -v "favicon"`	Follow errors, ignoring favicon 404s

14. Cron Jobs & Scheduling

crontab syntax, systemd timers, at command

Crontab

`crontab -e`	Edit crontab for current user
`crontab -l`	List all cron jobs for current user
`crontab -r`	Remove all cron jobs for current user (careful!)
`crontab -u www-data -e`	Edit crontab for www-data user
`sudo crontab -e`	Edit root's crontab
`/etc/cron.d/`	System-wide cron jobs directory (one file per app)
`/etc/cron.daily/`	Scripts run daily by cron daemon automatically

Cron Expression Reference

```
# .---------------- minute (0-59) # | .------------- hour (0-23) # | | .---------- day of month (1-31) # | | | .------- month (1-12) # | | | | .---- day of week (0-7, 0 and 7 = Sunday) # | | | | | * * * * * command 0 3 * * * Daily at 3:00 AM */15 * * * * Every 15 minutes 0 0 * * 0 Every Sunday at midnight 0 0 1 * * First day of every month at midnight 30 23 * * 1-5 Mon-Fri at 11:30 PM
```

15. System Performance & Monitoring

CPU, RAM, load, I/O — diagnosis commands

CPU & Load

`top`	Interactive CPU and process monitor
`htop`	Better interactive monitor with color display
`uptime`	Show load averages for 1, 5, 15 minutes
`nproc`	Show number of CPU cores/threads available
`lscpu`	Detailed CPU information (cores, threads, cache)
`vmstat 1 5`	System stats every 1 second, 5 times (CPU, mem, I/O)
`mpstat -P ALL 1`	Per-CPU statistics every 1 second
`sar -u 1 5`	CPU utilization history (install sysstat)

Memory & Swap

`free -h`	Show RAM and swap usage in human-readable format
`cat /proc/meminfo`	Detailed memory statistics
`vmstat -s`	Summary of memory, swap, and I/O statistics
`swapon --show`	Show active swap devices and usage
`swapoff -a && swapon -a`	Clear swap — moves pages back to RAM (use carefully)

Disk I/O

`iostat -x 1`	Extended I/O stats every 1 second (install sysstat)
`iotop`	Interactive I/O monitor by process (apt install iotop)
`dstat`	All stats: CPU, memory, disk, network combined (apt install dstat)
`hdparm -t /dev/sda`	Test raw read speed of disk device

16. Security Hardening

fail2ban, unattended-upgrades, auditd, essential hardening

fail2ban

`apt install fail2ban`	Install fail2ban brute-force protection
`systemctl enable --now fail2ban`	Enable and start fail2ban
`fail2ban-client status`	Show all active jails
`fail2ban-client status sshd`	Show SSH jail — banned IPs, current counts
`fail2ban-client set sshd banip 1.2.3.4`	Manually ban an IP
`fail2ban-client set sshd unbanip 1.2.3.4`	Unban an IP
`/etc/fail2ban/jail.local`	Local config overrides — create this, never edit jail.conf

Automatic Security Updates

`apt install unattended-upgrades`	Install automatic security update daemon
`dpkg-reconfigure --priority=low unattended-upgrades`	Configure via wizard
`/etc/apt/apt.conf.d/50unattended-upgrades`	Main config file — control what gets updated
`unattended-upgrades --dry-run -d`	Test what would be upgraded without applying

System Audit

`lynis audit system`	Comprehensive security audit (apt install lynis)
`rkhunter --check`	Rootkit scanner (apt install rkhunter)
`chkrootkit`	Alternative rootkit checker (apt install chkrootkit)
`netstat -tlnp`	Check all open listening ports — audit regularly
`ss -tlnp`	Modern alternative to netstat
`last -n 20`	Review last 20 logins — check for unexpected access
`ausearch -m USER_LOGIN -ts today`	Audit log login events today (install auditd)

17. Backup & File Transfer

rsync, scp, automated backup scripts

rsync — Efficient File Sync

`rsync -avz /local/ user@host:/remote/`	Sync local dir to remote (trailing / matters)
`rsync -avz --delete /local/ user@host:/remote/`	Mirror — delete files removed from source
`rsync -avz --progress --exclude="*.log" /src/ /dst/`	Show progress, exclude log files
`rsync -avz -e "ssh -p 2222" /src/ user@host:/dst/`	Use custom SSH port
`rsync -avz --dry-run /src/ /dst/`	Preview what would be synced without doing it
`rsync -avz user@host:/remote/ /local/`	Pull files from remote to local

scp — Secure Copy

`scp file.txt user@host:/path/`	Copy single file to remote server
`scp -r /local/dir user@host:/path/`	Copy directory recursively
`scp user@host:/remote/file .`	Copy file from remote to current directory
`scp -P 2222 file user@host:/`	Use custom SSH port

Automated MySQL Backup Script

```
#!/bin/bash DATE=$(date +%Y%m%d_%H%M) BACKUP_DIR="/var/backups/mysql" mkdir -p $BACKUP_DIR mysqldump -u root -p'yourpassword' --single-transaction --all-databases \ | gzip > $BACKUP_DIR/all_dbs_$DATE.sql.gz # Keep last 7 days only find $BACKUP_DIR -name "*.sql.gz" -mtime +7 -delete # Add to crontab: 0 2 * * * /usr/local/bin/mysql_backup.sh
```

18. Useful One-Liners & Recipes

Copy-paste ready commands for common tasks

System Info at a Glance

`uname -a`	Show kernel version, architecture, hostname
`cat /etc/os-release`	Show Linux distribution and version
`lsb_release -a`	Ubuntu/Debian distro info
`uptime && free -h && df -h`	Quick system health snapshot in one line
`ss -tlnp && ufw status`	Open ports + firewall status
`systemctl list-units --failed`	All failed services at once

Web Server Recipes

`nginx -t && systemctl reload nginx`	Safe: test config then reload only if valid
`certbot renew --dry-run && echo OK`	Test SSL renewal works
`curl -sk https://localhost -o /dev/null -w "%{http_code}\n"`	Check HTTPS returns 200
`grep " 500 " /var/log/nginx/access.log | tail -20`	Last 20 500 errors
`tail -f /var/log/nginx/error.log | grep -v " 404 "`	Watch errors excluding 404s
`apache2ctl -t`	Test Apache config syntax

MySQL One-Liners

`mysqlcheck -u root -p --all-databases --auto-repair`	Check and repair all tables
`mysql -u root -p -e "SHOW STATUS LIKE 'Slow_queries';"`	Count slow queries since startup
`mysql -u root -p -e "SELECT table_schema, SUM(data_length+index_length)/1024/1024 AS MB FROM information_schema.tables GROUP BY 1 ORDER BY 2 DESC;"`	Database sizes in MB
`mysqldump -u root -p db | gzip | ssh user@backup "cat > /backups/db.sql.gz"`	Stream backup directly to remote server

Disk & Process Emergencies

`df -h && du -sh /var/log/* | sort -rh | head -10`	Find what is filling the disk
`lsof | grep deleted | grep -v DEL`	Find deleted files still held open (disk not freed)
`kill $(lsof -t -i:80)`	Kill whatever is using port 80
`pkill -9 -u baduser`	Kill all processes owned by a user
`echo 3 > /proc/sys/vm/drop_caches`	Clear file system caches (frees RAM temporarily)

hostitworks.com

24 Free Web Hosting & Developer Tools

Bookmark these free tools — they complement every command in this cheat sheet.

■ DNS & Network

- DNS Lookup Tool
- DNS Propagation Checker
- IP Lookup & WHOIS
- Port Checker

■ Security

- SSL Certificate Checker
- HTTP Header Checker
- Password Generator
- PHP/MySQL Version Checker

■ Performance

- Website Speed Tester
- TTFB Speed Tester
- Server Uptime Checker
- Website Ping Tool

→ Visit hostitworks.com for All 24 Free Tools ←

Browser-based tools — no account, no signup, completely free.