Ubuntu Linux Essentials Cheatsheet

A quick reference guide to essential commands and concepts for navigating and managing Ubuntu Linux systems.

Basics & File System

Basic Commands

`pwd`	Print working directory (current location).
`ls`	List directory contents. **Options:** `-l` (long listing) `-a` (all files, including hidden) `-h` (human-readable sizes)
`cd [directory]`	Change directory. **Examples:** `cd ..` (go up one level) `cd ~` (go to home directory) `cd /` (go to root directory)
`mkdir [directory]`	Create a new directory.
`rm [file]`	Remove/delete a file. **Options:** `-r` (recursive, for directories) `-f` (force, no prompt)
`cp [source] [destination]`	Copy files or directories. **Example:** `cp file.txt /tmp/`
`mv [source] [destination]`	Move or rename files/directories.
`touch [file]`	Create an empty file or update timestamp.
`man [command]`	Display the manual page for a command.

File Viewing & Editing

`cat [file]`	Concatenate and display file content.
`less [file]`	View file content page by page (scrollable).
`head [file]`	Display the first 10 lines of a file. **Option:** `-n X` (display X lines)
`tail [file]`	Display the last 10 lines of a file. **Option:** `-f` (follow file as it grows)
`grep [pattern] [file]`	Search for lines matching a pattern in a file. **Example:** `grep "error" /var/log/syslog`
`nano [file]`	Open file in the Nano text editor (simple).
`vim [file]`	Open file in the Vim text editor (powerful, complex).
`echo [text]`	Display text. **Example:** `echo "Hello World"` `echo "Hello" > file.txt` (redirect output to file, overwrites) `echo "World" >> file.txt` (append output to file)

Permissions (chmod)

`chmod [permissions] [file]`	Change file permissions.
Permissions Structure	Represented as `owner | group | others` Each part has `read (r) | write (w) | execute (x)`
Symbolic Notation	`u` : user (owner) `g` : group `o` : others `a` : all `+` : add permission `-` : remove permission `=` : set permission
Symbolic Examples	`chmod u+x script.sh` (Add execute for owner) `chmod g-w file.txt` (Remove write for group) `chmod o=r data.csv` (Set read for others, remove others' other permissions) `chmod a+rw index.html` (Add read/write for all)

Octal Notation	Each permission type has a value:
	`r = 4`
	`w = 2`
	`x = 1`
	`- = 0`
	Sum values for `owner`, `group`, `others`.

Octal Values per Entity	`7 = rwx` (4+2+1) `6 = rw-` (4+2+0) `5 = r-x` (4+0+1) `4 = r--` (4+0+0) `3 = -wx` (0+2+1) `2 = -w-` (0+2+0) `1 = --x` (0+0+1) `0 = ---` (0+0+0)
Octal Examples	`chmod 755 script.sh` (Owner: rwx, Group: r-x, Others: r-x) `chmod 644 file.txt` (Owner: rw-, Group: r–, Others: r–)
Recursive Change	Use `-R` for recursive changes on directories and their contents. **Example:** `chmod -R 755 mydir/`

Package Management (APT)

Updating & Upgrading

`sudo apt update`	Fetches the list of available packages from the repositories and updates the package information database. Run this before installing or upgrading.
`sudo apt upgrade`	Installs the newest versions of all packages currently installed on the system from the sources enumerated in `/etc/apt/sources.list` and its subsections.
`sudo apt full-upgrade`	Performs the same function as `upgrade` but will remove currently installed packages if it is necessary to upgrade the system as a whole. Can sometimes resolve dependency issues.
`sudo apt autoremove`	Removes packages that were automatically installed to satisfy dependencies for other packages and are no longer needed.
`sudo apt autoclean`	Removes old downloaded package files from the `/var/cache/apt/archives/` directory that can no longer be downloaded and are essentially useless.
Best Practice	Always run `sudo apt update` followed by `sudo apt upgrade` regularly to keep your system secure and up-to-date.

Installing & Removing

`sudo apt install [package_name]`	Install a new package and its dependencies. **Example:** `sudo apt install firefox`
`sudo apt install [pkg1] [pkg2]`	Install multiple packages at once.
`sudo apt remove [package_name]`	Remove a package (leaves configuration files).
`sudo apt purge [package_name]`	Remove a package and its configuration files.
`sudo dpkg -i [package.deb]`	Install a package from a .deb file (doesn't handle dependencies automatically).
`sudo apt --fix-broken install`	Attempt to correct a system with broken dependencies.

Searching & Information

`apt search [keyword]`	Search for packages containing the keyword.
`apt show [package_name]`	Display detailed information about a package (version, dependencies, size, description, etc.).
`apt list --installed`	List all installed packages.
`apt list --upgradable`	List all packages that have available updates.
`dpkg -L [package_name]`	List all files installed by a specific package.
`dpkg -S [file_path]`	Find which package a file belongs to.

Users & Processes

User and Group Management

`whoami`	Display the current effective username.
`id [username]`	Display user and group information for a given user.
`sudo adduser [username]`	Create a new user (interactive process). Creates home directory, sets shell, adds to a group with the same name.
`sudo deluser [username]`	Delete a user (leaves home directory and mail spool).
`sudo deluser --remove-home [username]`	Delete a user and their home directory/mail spool.
`sudo usermod -aG [groupname] [username]`	Add an existing user to an existing group. **Example:** `sudo usermod -aG sudo myuser` (Add user to sudo group)
`sudo addgroup [groupname]`	Create a new group.
`chown [user]:[group] [file]`	Change file owner and group. **Example:** `chown myuser:mygroup myfile.txt` `chown -R myuser: mydir/` (Recursively change owner only)
`chgrp [group] [file]`	Change file group.

Process Management

`ps aux`	Display information about running processes. **Options:** `a`: show processes for all users `u`: display process owner `x`: show processes not attached to a terminal
`top`	Display dynamic real-time view of running processes (press `q` to exit).
`htop`	Interactive process viewer (more user-friendly than `top`, may need `sudo apt install htop`).
`kill [PID]`	Send a signal (default is TERM, 15) to a process to terminate it gracefully.
`kill -9 [PID]`	Send a KILL signal (9) to forcefully terminate a process (use as a last resort).
`pkill [name]`	Kill processes by name. **Example:** `pkill firefox`
`pgrep [name]`	Find process IDs by name. **Example:** `pgrep sshd`
`jobs`	List jobs running in the background or stopped in the current shell session.
`fg [%jobspec]`	Bring a background job to the foreground.

Running Commands as Another User (sudo)

`sudo [command]`
 	
Execute a command with root privileges.
 	
Example:
`sudo apt update`	

`sudo -i`	Start an interactive root shell.
`sudo -s`	Start a shell as the superuser, but with the current user's environment variables.
`sudo su -`	Switch to the root user's environment (similar to `sudo -i`).
`sudo visudo`	Edit the `/etc/sudoers` file safely (determines which users can run which commands as root). Use this instead of a regular text editor.
Adding User to sudo Group	Users in the `sudo` group (or sometimes `adm`) can use `sudo`. **Command:** `sudo usermod -aG sudo [username]` (Requires logging out and back in or starting a new shell session to take effect).

Networking & System Info

Network Commands

`ip addr show`	Display network interface information (IP addresses, MAC addresses). **Alternative (older):** `ifconfig` (may require `sudo apt install net-tools`)
`ip route show`	Display the IP routing table. **Alternative (older):** `route -n`
`ping [hostname/IP]`	Send ICMP ECHO_REQUEST to network hosts (test connectivity).
`traceroute [hostname/IP]`	Print the route packets take to a network host.
`netstat -tulnp`	Display active network connections, listening ports, and associated processes. **Options:** `t` : tcp `u` : udp `l` : listening sockets `n` : numeric addresses `p` : show PID/program name (requires root)
`ss -tulnp`	Faster alternative to `netstat`.
`ssh [user]@[hostname/IP]`	Secure Shell: Connect to a remote server. **Example:** `ssh myuser@192.168.1.100`
`scp [source] [destination]`	Secure Copy: Copy files securely over SSH. **Examples:** `scp myfile.txt user@remote:/path/` (local to remote) `scp user@remote:/path/myfile.txt .` (remote to local)

System Information

`uname -a`	Print all system information (kernel name, hostname, kernel release, kernel version, hardware, OS).
`lsb_release -a`	Display LSB (Linux Standard Base) and distribution-specific information (Ubuntu version). **Example:** `No LSB modules are available. Distributor ID: Ubuntu Description: Ubuntu 22.04.3 LTS Release: 22.04 Codename: jammy`
`df -h`	Report file system disk space usage in human-readable format.
`du -sh [directory]`	Estimate file space usage (summary, human-readable). **Example:** `du -sh /var/log/`
`free -h`	Display amount of free and used memory in the system in human-readable format.
`uptime`	Show how long the system has been running, number of users, and load averages.
`date`	Print or set the system date and time.
`hostname`	Display the system's hostname.

System Control (systemd)

`sudo systemctl status [service]`	Show the status of a systemd service (running, stopped, etc.). **Example:** `sudo systemctl status apache2.service`
`sudo systemctl start [service]`	Start a service.
`sudo systemctl stop [service]`	Stop a service.
`sudo systemctl restart [service]`	Restart a service.
`sudo systemctl enable [service]`	Enable a service to start automatically at boot.
`sudo systemctl disable [service]`	Disable a service from starting automatically at boot.
`sudo systemctl reload [service]`	Reload configuration files for a service (if supported).
`sudo systemctl list-units --type=service`	List all loaded service units.