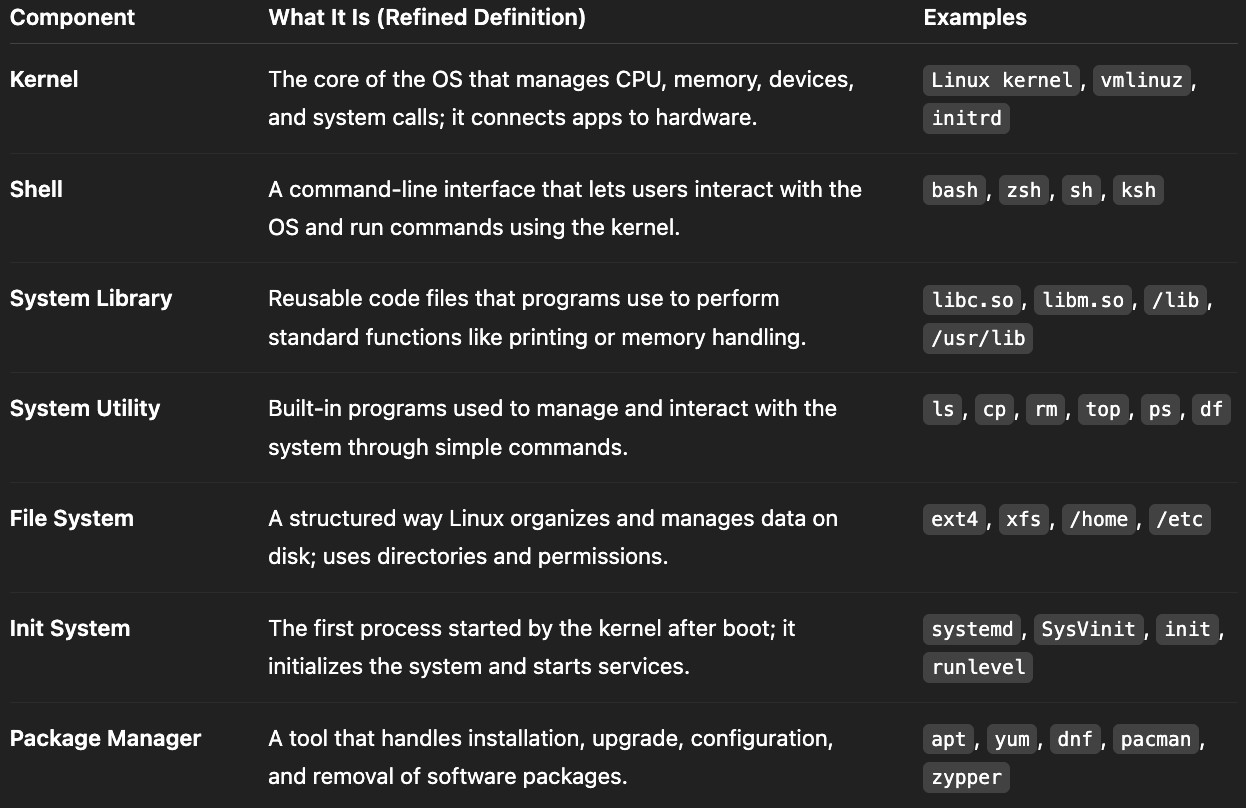
**Components of linux OS**



\* apt is the default package manager of linux. It creates a dependency tree for required package and install all necessary dependencies.

\* The line #!/bin/bash at the **top of a shell script** is called a **shebang. It means run the shell script in bash.**

**Folder structure at root**

**/sbin** (system binaries)- contains command files required by administrator. like adduser, mount a file system, reboot system <system utility>

**/bin**- contains command files required by all users like cp,ls,r,cat,mkdir,apt <system utility>

**/lib**- used by kernel <system library>  
  
**/boot**- contains files needed to restart the system.

**/usr**- contains most of the installed softwares and it’s files documents. It’s like ProgramFiles in windows

**/srv**- contains data for system services like websites, FTP, and databases like /srv/http, /src/ftp etc.

**/opt**- used to store third-party software or manually installed applications. Every user should manually install 3rd party software here only. Not managed by apt or yum.

**/mnt**- Used by system admin to temporarily access external devices or partitions by mounting them into the file system

**/var**- /var/log stores log files of system, services, applications. (basically all logs)

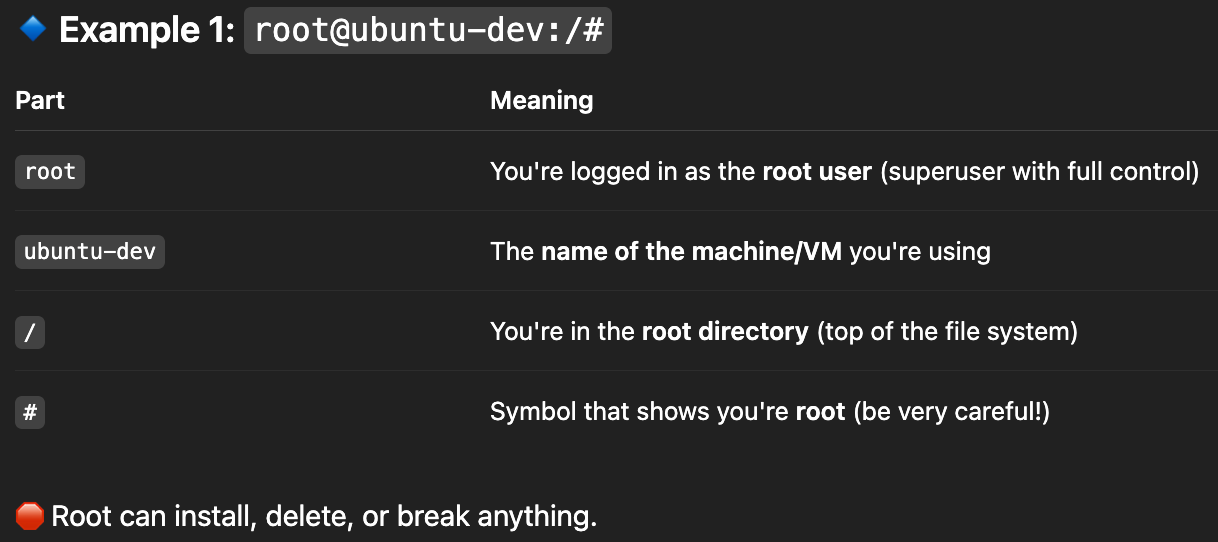
**/home**- Stores personal files for each normal user, which are user specific like /home/Ishan/scripts , /home/Ishan/Documents

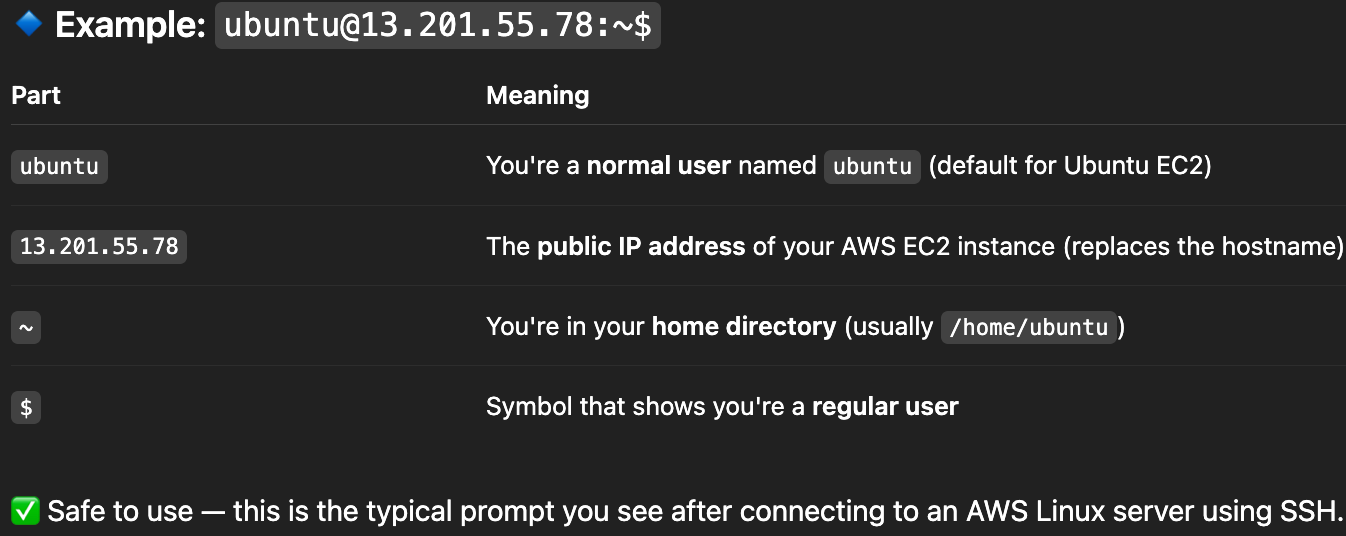
**/etc**- stores all configuration files for system (hostname) and major applications. If you install a tool like Apache, Docker, or MySQL, their configs typically live in /etc

**/tmp**- stores cache and temporary data. emptied on reboot.

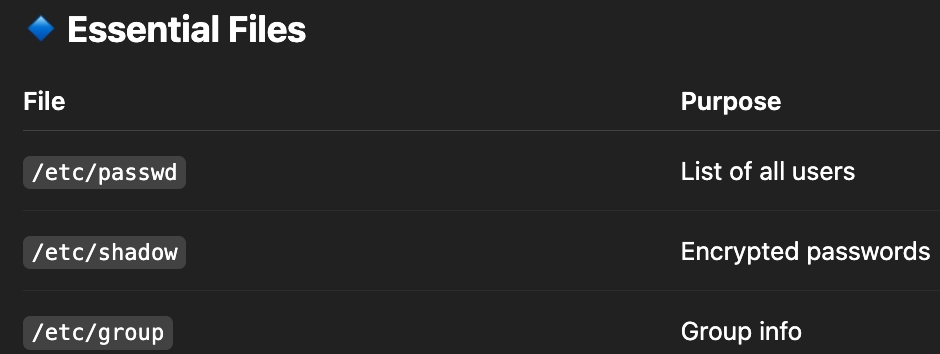
**/run**- stores runtime data of processes like processID, sockets docker.sock

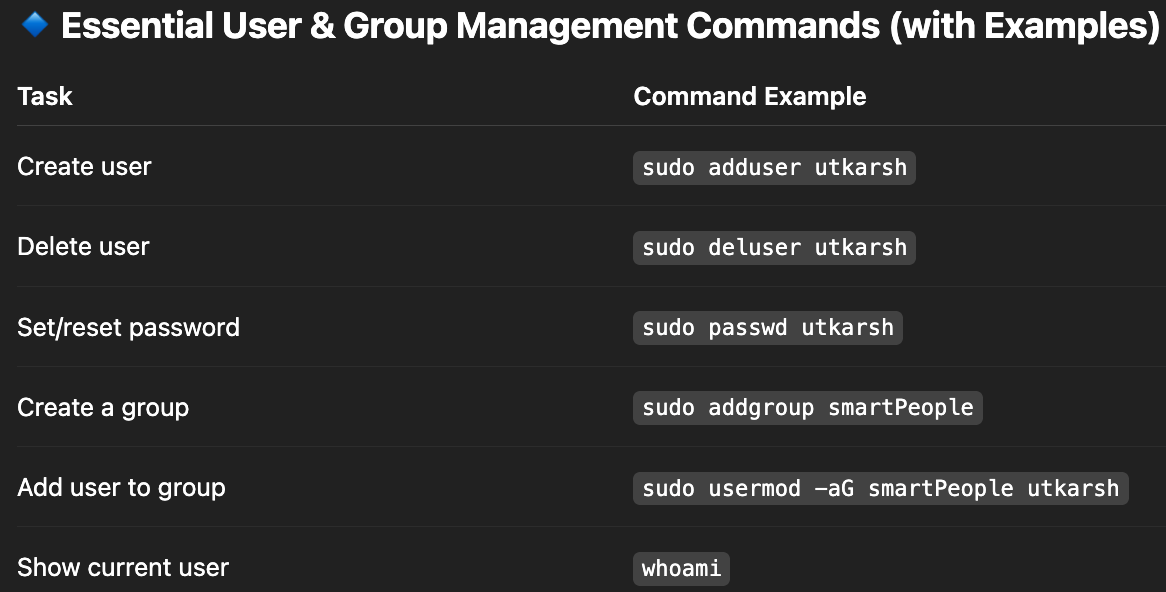
**Linux prompt**  
  
user@hostname:directory\_symbol





**User & Permission Management**





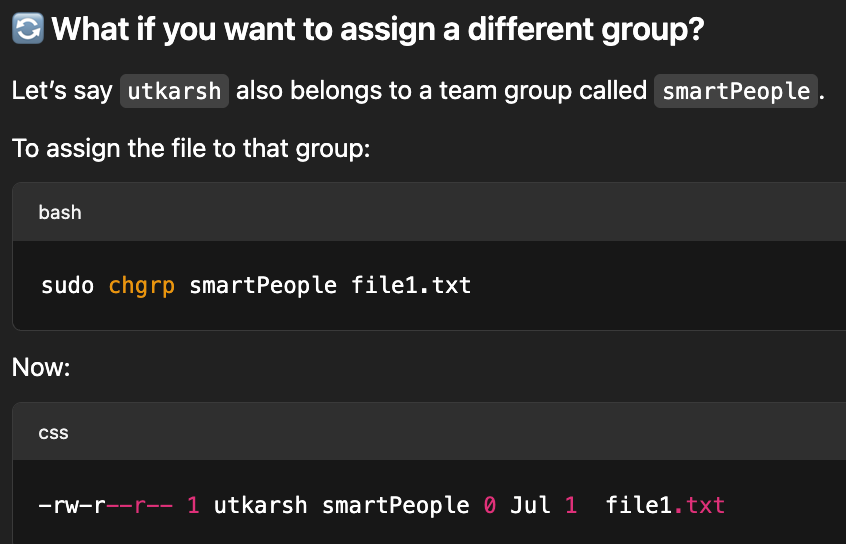
- “useradd” command doesn’t create a user directory under /home, while “adduser” does.

-We can’t restore a user’s password, we can only view encrypted password.

Switch user : su usernameX

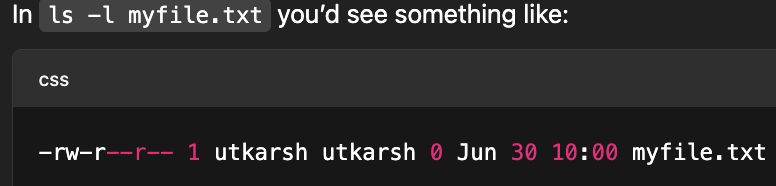
Change owner: sudo chown new-owner file1.txt

Change group: sudo chgrp new-grp file1.txt



When your account was created (e.g., “utkarsh”), the system also created a group with the same name (“utkarsh”) and made that your default group.

When a user creates a file, Linux automatically assigns two things: user (owner) & user’s default group.

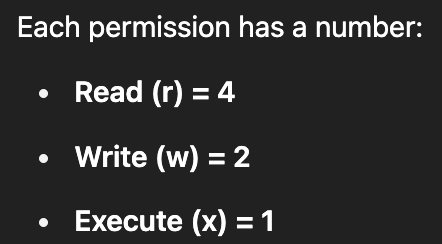


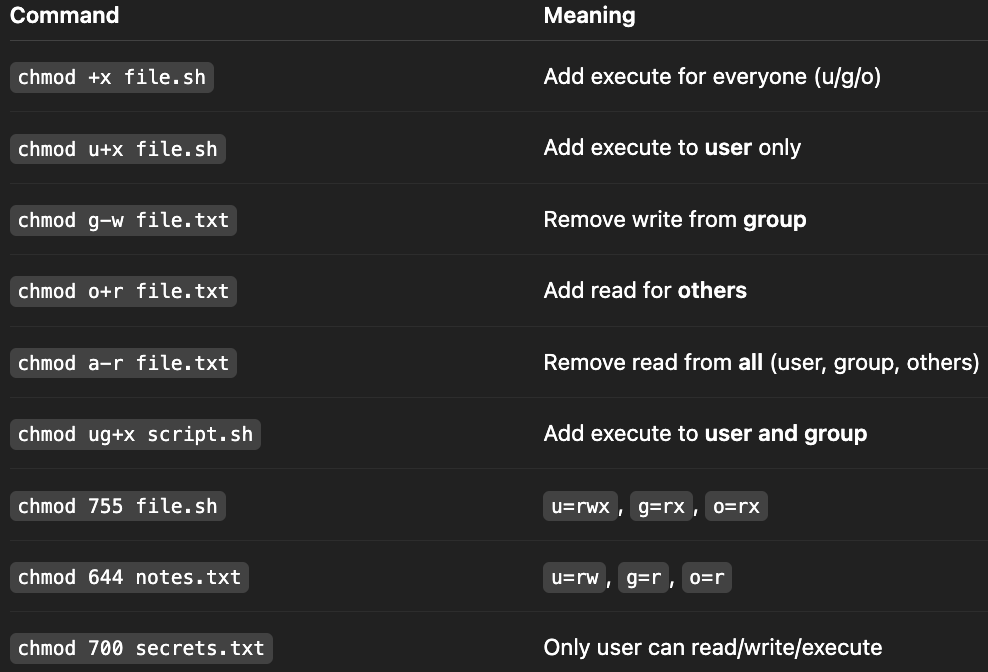
**chmod command (change mode)**

**- 3 type of user we set permission for - user (u), group (g), others (o).**

**User means owner of file. Group includes users in the file’s group**

**- 3 types of access- read, write, execute**





-rwxr-xr-- 1 utkarsh smartPeople 5467 Jun 28 test.sh

| **Section** | **Example** | **What It Means** |
| --- | --- | --- |
| -rwxr-xr-- | Permissions | File type, here regular (for directory: d, link: l) + who can read/write/execute |
| 1 | Link count | Number of hard links to this file |
| utkarsh | Owner | The user who owns the file |
| smartPeople | Group | The group that owns the file |
| 5467 | File size | Size in **bytes** |
| Jun 28 | Modified date | Last modified **month and day** |
| test.sh | File name | Name of the file |

**Basic Commands**

touch <file> - create empty file

vi <file> - opens the file in text editor, if not created then create and opens.

cat <file> - to show content of the file

man <command> - tells what that command does and available flags like “man ls”

pwd - print working directory

mkdir - make directory

tail -n 5 utkarsh.txt - last 5 lines of utkarsh file. Used to see the recent logs

head -n 10 utkarsh.txt - first 10 lines of utkarsh file

mv - to rename a directory or file. Eg. mv projects final\_projects

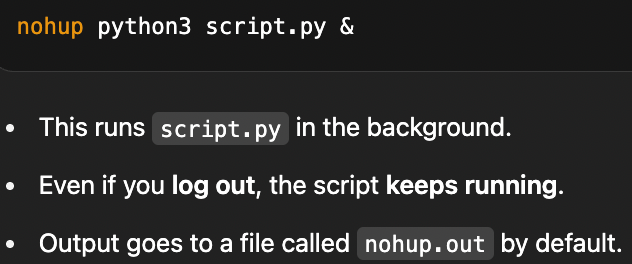
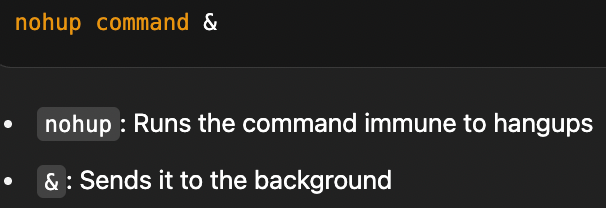
df -h - shows free/used disk space. used to check when disk partition filling.

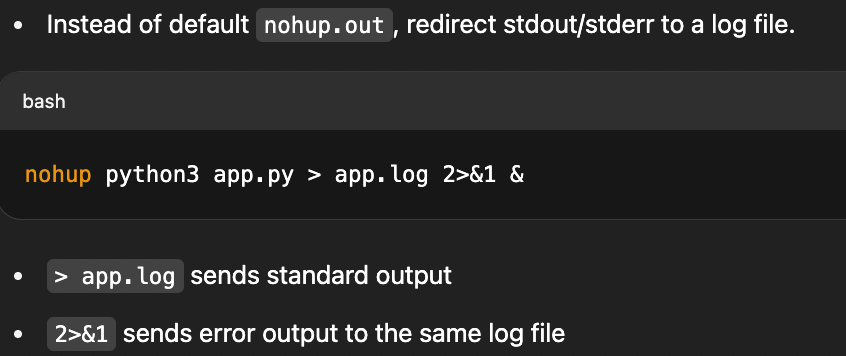
free -h - shows free/used RAM.

scp command - Used to copy files between local <-> remote. Uses ssh

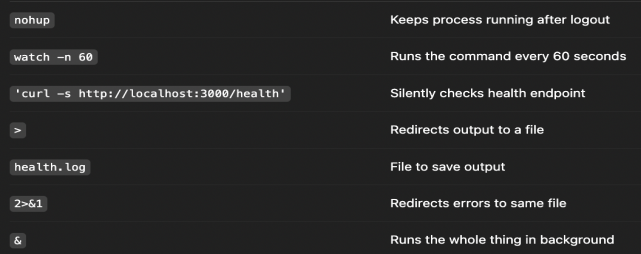
scp -i my-key.pem utk.txt ubuntu@13.234.22.111:/home/ubuntu/

nohup - It stands for "no hang up", it prevents a process from being stopped when the terminal session is closed.

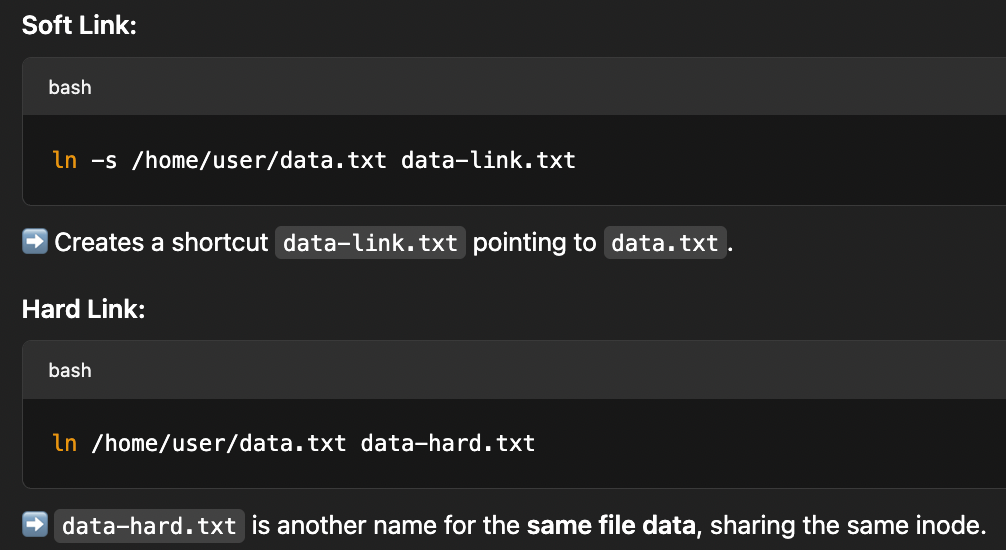




nohup watch -n 60 'curl -s http://localhost:3000/health' > health.log 2>&1 &



softlink vs hardlink - Soft links are like shortcuts to a file, while hard links are backup of file. Soft links break if the original file is deleted, but hard links do not.



**Processes commands**

top - shows all details about process and thread on system like Task Managr. Used to monitor node health.

kill -9 <processID> - force kill a process

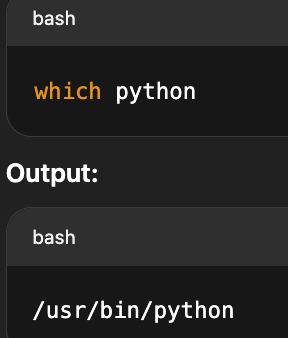
**System commands**

sudo - It allows a regular (non-root) user to run commands as the root user. It is a group. We can add users to sudo group using:

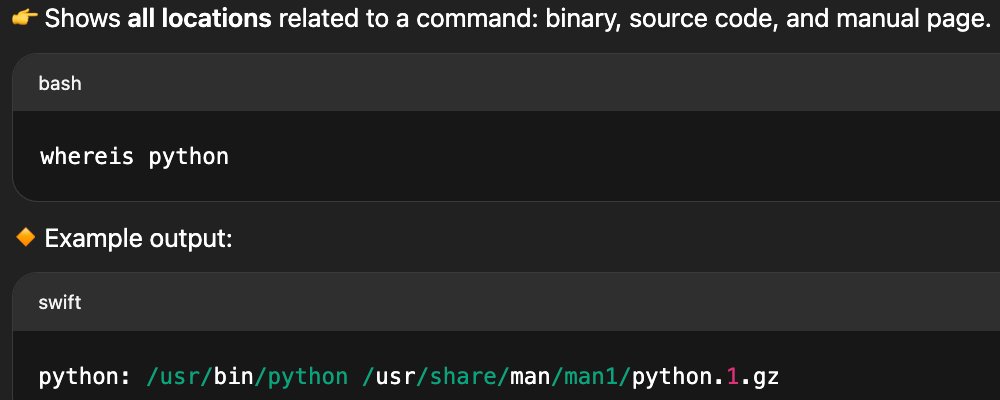
“sudo usermod -aG sudo username”

uptime - system running time, number of logged-in users, and load averages (how many process waiting for CPU)

which - find the full path of executable files.

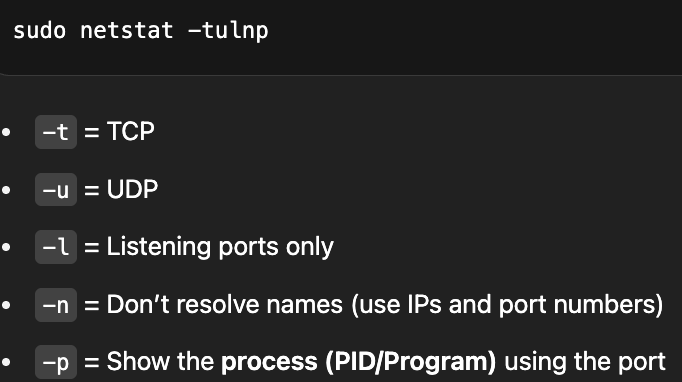


whereis



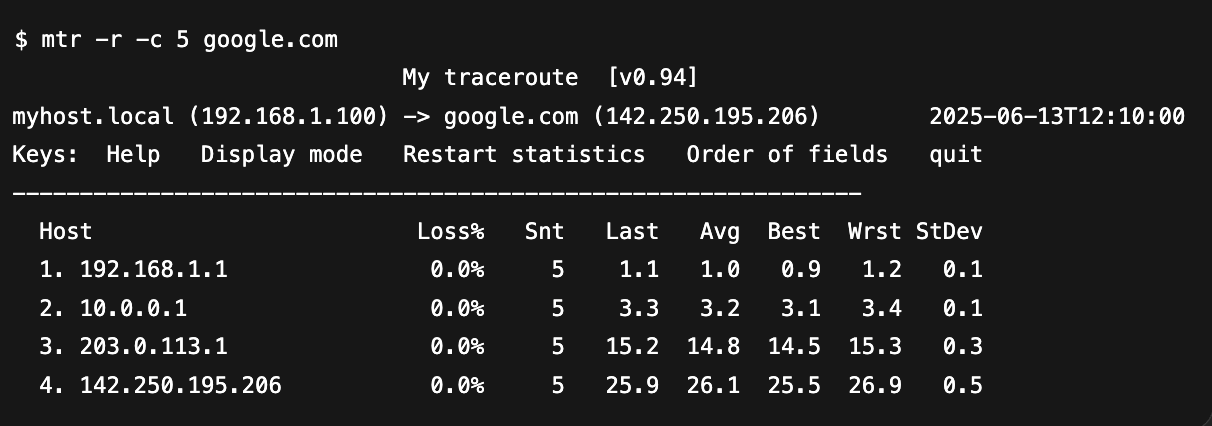
**Networking commands**

netstat- check which ports are open,tcp udp connections, whether a service is listening, and which process is running on what port. It’s useful for debugging connectivity or firewall issues.



ip a : ip a shows everything about your computer’s network connections, including: IP addresses (IPv4 and IPv6), Whether the connection is active, Hardware (MAC) address Network interface details (like Wi-Fi or Ethernet or loopback)

mtr : (My Trace route) .Traces the route to a host with hops(like traceroute). Continuously pings each hop along the path (like ping). Helps to detect where delays or packet loss are happening.



telnet : tMainly used to test connectivity to specific ports on a remote host. Helps check if a service (like a web server on port 80, or an SMTP mail server on port 25) is reachable.

