**# File Management**
cd: Change directory
cd ..: to a directory one level up
cd ~: to a home directory
cd -: to a previous working directory
cd <path>: to a specified path

mkdir dir_name: create dir in current loc
mkdir dir1 dir2 dir3: create multiple dir in current loc
mkdir /path/to/directory: create dir in specific path
mkdir -p /a/b/c/: Create a dir with multiple levels of nested dir
mkdir -m <mode> dir_name: Create dir with specific permissions
mkdir -i dir_name: Create dir interactively
mkdir -v dir_name: Create dir and display verbose output

rmdir dir_name: Remove an empty dir
rmdir dir1 dir2 dir3: remove multiple dir in current loc
rmdir /path/to/directory: remove dir from specific path
rmdir -r dir_name: Remove a dir and its contents recursively
rmdir -i dir_name: remove dir interactively
rmdir -v dir_name: remove dir and display verbose output

touch file_name: Create a new empty file
touch file1 file2 file3: Create multiple new empty files simultaneously
touch -t yyyymmddhhmm file_name: Create a new file with a specific timestamp
touch file_name: Update the access and modification timestamps of an existing file

cat file_name: Display the contents of a file
cat file1 file2 file3: Concatenate multiple files and display their contents
cat > file_name: Create or append to a file using user input
cat file1 file2 > new_file: Concatenate files and save the output to a new file
cat -n file_name: Display line numbers with the file contents
cat -v file_name: Display non-printable characters(tab, line breaks) with the file contents

head file_name: Display the first 10 lines of a file
head -n 5 file_name: Display a specific number of lines from the beginning of a file
head -c 100 file_name: Display the first part of a file with a specific number of bytes
head file1 file2: Display the first part of multiple files
command | head -n 20: Display the first few lines of a command output continuously

tail file_name: Display the last 10 lines of a file
tail -n 5 file_name: Display a specific number of lines from the end of a file
tail -c 100 file_name: Display the last part of a file with a specific number of bytes
tail file1 file2: Display the last part of multiple files
command | tail-n 20: Display the last few lines of a command output continuously

rm file_name: Remove a file
rm file1 file2 file3: Remove multiple files simultaneously
rm -r dir_name: Remove a directory and its contents recursively
rm -rf: Remove directories and their contents forcefully
rm -i file_name: Prompt for confirmation before deleting each file
rm -v file_name: Remove files and display verbose output
rm -d dir_name: Remove empty directories

ls: List files and directories in CWD
ls -l: display detailed info in long format
ls -a: list file and dir with hidden files
ls -t: sorts files and dir by modification time
ls -h: shows file sizes in human-readable format
ls -r: lists files and directories in reverse order
ls -R: lists files and dir recursively, including sub dir
ls -S: sorts files by size, with the largest first

cp source dest: copy files and directories
cp -i: ask before overwriting a file
cp -r: recursively copy an entire dir tree
cp -p: preserves permission, ownership, timestamps
cp -a: cp -r + cp -p
cp -v: display the name of files being copied

mv source_file dest_dir: Move a file
mv old_filename new_filename: Rename a file
mv -f source_file dest_dir: Move and overwrite file
mv file1 file2 file3 dest_dir: Move multiple files
mv source_dir dest_dir: Move a directory
mv old_dir new_dir: Rename a directory
mv -i source_file desti_dir: Move with prompt
mv -p source_file dest_dir: Move and Preserve
mv -R source_dir dest_dir: Move a dir recursively

**# Misc**

alias ll='ls -alF'
unalias ll
history
history | grep "search-term"
export MYVAR="Hello World"
export
unset MYVAR

**# Process Management**
ps: Lists the currently running processes
ps aux: Provides detailed information about all processes, including those running in the background.
ps -ef: Displays a full listing of processes with detailed information.
top: Provides real-time monitoring of system processes, resource usage, and system information. Press 'q' to exit.
htop provides an interactive and more user-friendly interface compared to top
kill PID: Sends a termination signal to the process identified by the process ID (PID). killall process_name: Terminates all processes with the specified name.
pkill process_name: Sends a termination signal to processes based on their name. This command is useful for killing multiple processes with similar names.
pgrep process_name: Lists the process IDs (PIDs) of processes matching the specified name.

python3 flaskapp.py & (background process)
fg %1 (foreground process)
Jobs (show background process)
bg %1(foreground to background process)
nohup flask run &

cat nohup.out

**# Ownership and Permission**
chmod 755 file_name: Change permissions using octal notation
chmod -R 755 directory_name: Change permissions recursively (including subdirectories)
chmod u=rw,go=r file_name: Change permissions using symbolic notation
chown new_owner file_name: Change the owner of a file
chown new_owner:new_group file_name: Change the owner and group owner of a file
chown :new_group file_name: Change the group owner of a file or directory

**# User Management**
whoami
groupadd dev-team
cat /etc/group | grep dev-team
useradd -G dev-team pranjal
useradd -G dev-team user1
cat /etc/group | grep dev-team
passwd pranjal
passwd user1
mkdir /home/dev-team
ls -ltr
chown :dev-team /home/dev-team/
chmod g+w /home/dev-team/
chmod o-rx dev-team

su - pranjal
whoami
cd /home/dev-team
touch pranjal-file.txt
ls -lrt
chown :dev-team pranjal-file.txt
ls -lrt
exit

su - user1
cd /home/dev-team
ls -l | grep pranjal-file.txt
echo "This is user1's comment" > pranjal-file.txt
cat pranjal-file.txt
exit

groupadd test-team
useradd -G test-team user2
passwd user2
su - user2
cd /home/dev-team

**# File Transfer and Remote access**

sudo ssh -i "batch5kp.pem" ubuntu@ec2-44-201-158-129.compute-1.amazonaws.com
sudo scp -i "batch5kp.pem" hello.txt ubuntu@ec2-44-201-158-129.compute-1.amazonaws.com:/home/ubuntu

sudo scp -i batch5kp.pem ubuntu@ec2-44-201-158-129.compute-1.amazonaws.com:/home/ubuntu /shared-folder
sudo rsync -av -e "ssh -i batch5kp.pem" omen-ec2
ubuntu@ec2-44-201-158-129.compute-1.amazonaws.com:/home/ubuntu
sudo rsync -av -e "ssh -i batch5kp.pem"
ubuntu@ec2-44-201-158-129.compute-1.amazonaws.com:/home/ubuntu/omen-ec2 omen-ec2
sudo sftp -i batch5kp.pem ubuntu@ec2-44-201-158-129.compute-1.amazonaws.com

**# Monitoring system logs**
journalctl: View all system logs
journalctl -u ssh.service
journalctl -p err: View logs with a error log level
journalctl --since "2023-06-27 00:00:00" --until "2023-06-28 12:00:00": View logs within a specific time range

**# Linux Firewall Configuration**
sudo ufw allow 80: Allow HTTP traffic
sudo ufw default deny incoming: Block all other incoming traffic
ufw reset: reset
ufw delete allow 80/tcp: Delete rule
ufw deny 80/tcp: Deny rule
iptables -L: View Iptable
iptables -A INPUT -p tcp --dport 22 -j ACCEPT: Allow SSH connections
iptables -P INPUT DROP: Drop all other incoming traffic
iptables -F: Flush existing rules

**# Compressing Files**
sudo apt install zip
zip archive.zip file1 file2 file3
zip -u archive.zip file4 file5
zip -r archive.zip
unzip -l archive.zip
unzip archive.zip file1 file2
unzip archive.zip

tar -cvf archive.tar file1 file2 file3: Create a tar archive
tar -cvzf archive.tar.gz file1 file2 file3: Create a compressed tar archive
tar -tvf archive.tar: List the contents of a tar archive
tar -rvf archive.tar file4 file5: Append files to an existing tar archive
tar -xvf archive.tar: Extract files from a tar archive
tar -xzvf archive.tar.gz: Extract files from a compressed tar archive
tar -xvf archive.tar -C /path/to/directory: Extract files from a tar archive to a specific directory

**# Text Processing**

grep "pattern" filename: Search for a pattern in a file
grep "pattern" file1 file2 file3: Search for a pattern in multiple files
grep "pattern" -r directory: Search for a pattern in all files within a directory (recursively)
grep -i "pattern" filename: Search for a pattern, ignoring case sensitivity
grep -n "pattern" filename: Display line numbers along with matching lines
grep -v "pattern" filename: Search for lines that do not match a pattern
grep -E "regex_pattern" filename: Use regular expressions for pattern matching
grep -w "pattern" filename: Search for whole word matches only
grep -c "pattern" filename: Display the number of matching lines

grep -A 2 -B 3 "pattern" filename: Search for a pattern, displaying a specific number of lines before and after each matching line

sed 's/search_string/replacement_string/' filename: Replace text in a file
sed 's/search_string/replacement_string/g' filename: Replace all occurrences of a pattern in a file
sed -i.bak 's/search_string/replacement_string/' filename: Replace text in a file, and create a backup of the original file
sed '/pattern/d' filename: Delete lines containing a specific pattern in a file
sed -e 's/pattern1/replacement1/' -e 's/pattern2/replacement2/' filename: Perform multiple operations with the -e option
sed -n '5,10p' filename: Print specific lines from a file

awk '{print $1, $3}' filename: Print specific columns from a file
awk -F',' '{print $1, $3}' filename: Specify a custom delimiter (e.g., comma) for input and output
awk '$3 > 50 {print $1, $3}' filename: Use a condition to filter records
awk '{print $1, $3}' filename > output.txt: Redirect output to a file
awk -F',' -v OFS='|' '{print $1, $3}' filename: Specify output field separator
awk '{print NR, $0}' filename: Print line number and content
awk '{total += $2} END {print "Total:", total}' filename: Perform calculations on selected fields
awk '/pattern/ {print}' filename: Print lines where a specific column matches a pattern

sort filename: Sort lines of a file in ascending order
sort -r filename: Sort lines of a file in descending order
Sort lines ignoring case sensitivity: sort -f filename
sort -R filename: Sort lines in a file in a random order
sort -u filename: Sort lines and remove duplicate lines
sort filename -o outputfile: Sort lines in a file and write the result to a new file

wc -l filename: Count the number of lines in a file
wc -w filename: Count the number of words in a file
wc -c filename: Count the number of characters in a file
wc filename: Count the number of lines, words, and characters in a file
command | wc: Count the number of lines, words, and characters from the output of a command
wc -l -w -c file1 file2 file3: Count the number of lines, words, and characters for multiple files and display a total at the end

find . -type f: Find files in the current directory and its subdirectories
find . -type d: Find directories in the current directory and its subdirectories
find . -name "filename": Find files or directories with a specific name
find . -iname "filename": Find files or directories ignoring case sensitivity in the name
find . -name "*.extension": Find files or directories with a specific extension
find . -mtime -n: ind files or directories modified within a certain time frame (n days)
find . -size +1M: Find files or directories based on their size
find . -user username: Find files or directories owned by a specific user
find . -perm 644: Find files or directories with specific permissions
find . -type f -exec command {} \;: Execute a command on each found file or directory
find . -empty: Find empty files or directories
find . -atime +n: Find files based on their access time (n days)
find . -group groupname: Find files or directories based on their group ownership
find . -regex "pattern": Find files or directories with specific names using regular expressions

**#System Information**
uptime | awk '{print $1,$2,$3}'
uptime | awk '{print $3,$4}' | cut -f1 -d,
uptime | awk '{print $6,$7,$8,$9,$10}'
top -n 1 -b | grep "load average:" | awk '{print $10 $11 $12}'
free -h: Check RAM and SWAP Usages
df -h: Check Disk Usages
du directory_path: Display the disk usage of a directory
du -h directory_path: Display the disk usage in a human-readable format
uname -m: Architecture
uname -r: Kernel Release
uname -o: Os type
who: Logged In users
---
**#Package Management:**
Updating Package Lists:

- apt update: Updates the local package lists from the repositories.
- apt-get update: Updates the local package lists from the repositories.

Installing Packages:

- apt install package_name: Installs the specified package and its dependencies.
- apt-get install package_name: Installs the specified package and its dependencies.

Updating Packages:

- apt upgrade: Upgrades all installed packages to their latest versions.
- apt-get upgrade: Upgrades all installed packages to their latest versions.
- apt full-upgrade: Performs a full upgrade, including the installation or removal of packages if necessary.
- apt-get dist-upgrade: Performs a distribution upgrade, handling changes in dependencies and package removals.

Removing Packages:

- apt remove package_name: Removes the specified package but retains its configuration files.
- apt-get remove package_name: Removes the specified package but retains its configuration files.
- apt purge package_name: Removes the specified package along with its configuration files.
- apt-get purge package_name: Removes the specified package along with its configuration files.

Searching for Packages:

- apt search keyword: Searches for packages containing the specified keyword in their names or descriptions.
- apt-cache search keyword: Searches for packages containing the specified keyword in their names or descriptions.

Additional Commands:

- apt show package_name: Displays detailed information about a package.
- apt-cache show package_name: Displays detailed information about a package.
- apt list: Lists all installed packages.
- apt autoremove: Removes automatically installed packages that are no longer needed.

### crontab -e
*** * * * * /path/to/command**

### Starting a Service:
$ sudo systemctl start service_name
Example: $ sudo systemctl start apache2 starts the Apache web server.
### Stopping a Service:
$ sudo systemctl stop service_name
Example: $ sudo systemctl stop apache2 stops the Apache web server.
### Restarting a Service:

$ sudo systemctl restart service_name
Example: $ sudo systemctl restart apache2 restarts the Apache web server.
**Reloading Configuration of a Service**:
$ sudo systemctl reload service_name
Example: $ sudo systemctl reload nginx reloads the configuration of the Nginx web server.
**Enabling a Service to Start on Boot:**
$ sudo systemctl enable service_name
Example: $ sudo systemctl enable ssh enables the SSH service to start automatically on boot.
**Disabling a Service from Starting on Boot:**
$ sudo systemctl disable service_name
Example: $ sudo systemctl disable apache2 disables the Apache web server from starting automatically on boot.
**Checking the Status of a Service:**
$ sudo systemctl status service_name
Example: $ sudo systemctl status nginx displays the status of the Nginx web server.
**Viewing the Logs of a Service:**
$ sudo journalctl -u service_name
Example: $ sudo journalctl -u mysql shows the logs of the MySQL service.
**Checking if a Service is Active or Enabled:**
$ sudo systemctl is-active service_name
$ sudo systemctl is-enabled service_name
Example: $ sudo systemctl is-active apache2 checks if the Apache web server is currently active. $ sudo systemctl is-enabled nginx checks if the Nginx web server is enabled to start on boot.

#### # Networking

ifconfig: Displays or configures network interfaces and their associated parameters.
Example: $ ifconfig displays information about all active network interfaces.

ip: Provides extensive control and information about network interfaces, routing tables, and more.
Example: $ ip addr show displays IP addresses assigned to network interfaces.

ping: Sends ICMP Echo Request packets to a specified host or IP address to check network connectivity.
Example: $ ping google.com sends ICMP Echo Request packets to Google's servers to check network connectivity.

traceroute: Traces the route packets take to reach a destination host, showing each hop along the way.
Example: $ traceroute google.com traces the route to Google's servers.

netstat: Displays network connections, routing tables, and network interface statistics.
Example: $ netstat -tuln displays a list of all listening TCP and UDP ports on the system.

nslookup: Queries DNS servers to retrieve DNS-related information about a domain or IP address.
Example: $ nslookup google.com performs a DNS lookup for the domain name "google.com".

dig: Similar to nslookup, dig is a versatile DNS lookup utility that provides detailed information about DNS records.
Example: $ dig google.com retrieves DNS information for the domain name "google.com".

wget: Downloads files from the web using various protocols, including HTTP, HTTPS, and FTP.
Example: $ wget http://example.com/file.zip downloads the file "file.zip" from the specified URL.

curl: Performs data transfers using various protocols, allowing you to send and receive data to/from remote servers.
Example: $ curl http://example.com retrieves the content of the specified URL.

ssh: Initiates a secure shell connection to a remote server for secure remote access and command execution.
Example: $ ssh username@hostname establishes an SSH connection to the specified hostname.