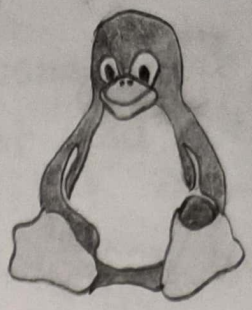
LINUX

Day - 1

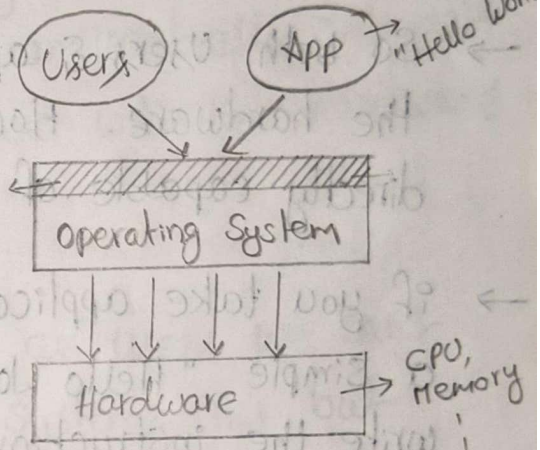
Fundamentals
Linux structure
Linux distributions
Setups
Package Manager



Fundamentals Of Linux

Why OS?

The physical component of any computer is what we call as Hardware.



- What is physical component?

Combination of CPU, memory, storage, motherboard & Network interfaces.

→ There are two primary actors who wants to interact with the hardware or who wants to interact with the computer.

- One is the User that is you & Me.
- Two, S/w applications (APP).

→ Users want to interact with the hardware, maybe to create files on the computer, folders on the computer, or even to manage the resources of the hardware. Maybe to check the status of the CPU.

→ Applications, if for example, if you're watching the video on YouTube, which is nothing but a S/w application.

& for Youtube to run it needs CPU, it needs memory and it also needs some displays. And all of these are available with the hardware.

→ So both Users & application wants to interact with the hardware. However, both of them are not directly capable of interacting with the hardware.

→ if you take applications first, imagine you write a simple "Hello World" program in Python. You do not write the instructions on how to manage the CPU for this program. You do not write the instructions on how to get memory allocated for this program. So obviously your application can't deal with the H/w resources b/c you have not written the code.

→ Similarly, as users, you can't directly interact with the physical component. You atleast need a graphical user interface or command line interface. And both of them are not provided by the hardware.

Because of these actors can't interact with the hardware. But a intermediate layer which is called as Operating System can help them.

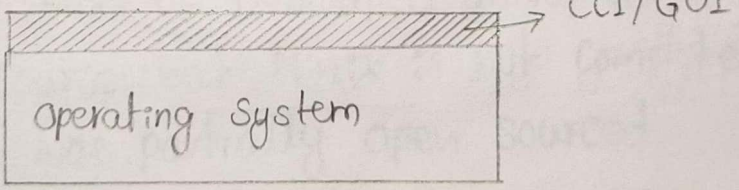
What is operating System?

operating system is a special S/w program that knows how to use the hardware.

So it can basically perform process management, memory management, device management and even network management.

Because application is also a S/w program. it can talk to the operating system which is a S/w layer and this operating system depending on the program it allocates the CPU, it allocates the memory & required hardware resources. This is how your program actually interact with the hardware. (or) S/w interact with the hardware through operating system.

Now you might wonder okay but as users we can't still interact with the operating systems. Yes but what operating systems do specially for the users they provide an additional layer. which is nothing but CLI or GUI.



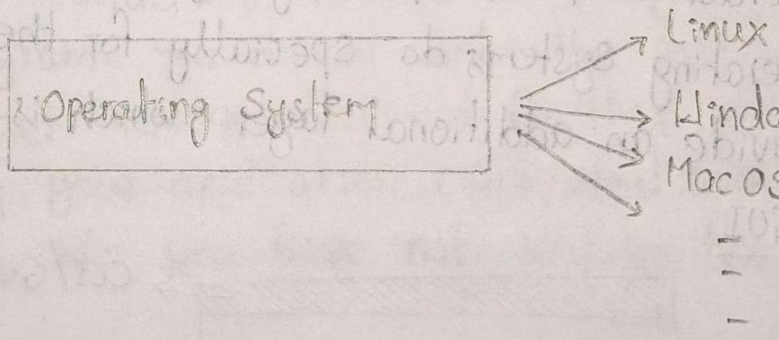
→ if you take Windows operating system, it provides a rich UI. whereas if you take Linux operating system, it provides rich CLI. Anyways, through this layer, users interact with OS & in turn talk to the hardware resources.

Definition of Operating System.

Operating System is an intermediate software layer that acts as bridge between software and hardware or acts as bridge between user & hardware. it performs process management, memory management, device management & N/w management.

→ Linux is not only the Operating System, there are other operating systems like windows, Mac OS & multiple other.

However, these are the three popular operating Systems in the market.



History of Operating Systems.

1960

Unix is the first os that was developed.

→ Before unix it was very difficult for the application developers. Although the application developers were very very less.

→ Still it was difficult for them to interact with the hardware. They used to do some advanced programming or typical hardware level coding to get their application interact with the h/w.

→ Unix solved this problem. So it became very popular in no time.

1970

another os was released which is Minix

→ Minix was kind of developed on top of Unix. Unix was open source but Minix is not completely open source. it was partially open sourced.

1980

There was a revolution in the space of Operating Systems then Microsoft released their OS which is Windows.

→ Microsoft's OS unlike the previous operating systems had a rich graphical user interface.

→ Now this rich graphical user interface made communication to the users with the hardware super simple.

→ They don't even need to know the commands which was required with Unix as well as Minix.

1990

→ Windows was very successful, Minux was also very successful. However, they both were kind of proprietary. They need some license for the develop to use or companies to use very specifically.

** That's where in 1990's a person called Linus worked on the development of Linux kernel & this was the first popular open-source OS.

→ Although Unix was also open-source OS, this was competitive with Windows. or in many of the parameters it also surpassed Windows.

→ So, Linux become the new popular choice for the developers. & also for the companies.

→ Even Today, if you look at Linux vs Windows, 90% of the production workloads today whether it is S/w running on your mobile or the applications that you're using deployed on the servers, most of

them, 90% of them use linux OS.

→ it's not because linux is free.

Many people think b/c linux is free companies go for linux. this is some part true.

However, the major part is that linux is open-source Now b/c linux is open-source, it's not backed by a single company, but it is backed by a community of developers.

so these developers they are huge in number and due to the active contributions they ensur linux is always secure.

** so the major reason why companies go for linux over windows is that linux is open-source. it is free & it is super secure compared to other operating systems.

Advantages of Linux over Windows

Cost-Effectiveness

→ free & open source : Linux does not require expensive licensing fees, making it a cost-effective choise for companies.

→ Lower Maintenance Costs : Linux is stable & requires minimal maintenace, reducing operational expenses.

Performance and Efficiency

→ Better Resource Utilization : Linux is lightweight & consumes fewer system resources compared to Windows.

→ High Scalability : Linux efficiently scales from embedded systems to enterprise data centers without performance degradation.

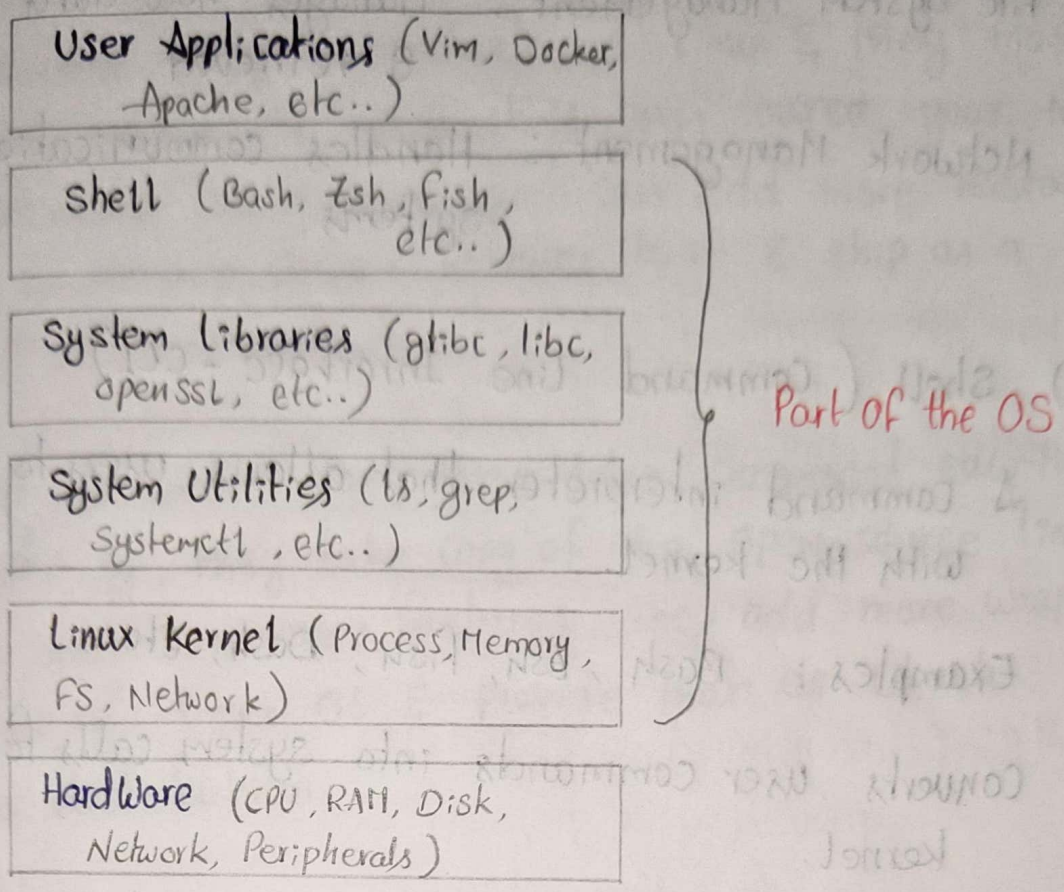
Security and Reliability.

→ Less Vulnerable to Malware : Linux has strong user privilege separation, making it more secure against viruses & malware.

→ frequent and Transparent Updates : Regular security patches ensure system stability without requiring frequent reboots.

→ High Stability : Linux system can run for years without cashes, ensuring better uptime & reliability.

Structure of Linux



(a). Hardware layer

- This is physical components of the computer (CPU, RAM, Disk, N/w interfaces, etc..)
- The OS interacts with hardware using device drivers.

(b). Kernel (core of Linux OS)

→ The Linux Kernel is responsible for directly managing system resources, including:

- Process Management: Schedules process & handles multitasking.
- Memory Management: Allocate & deallocate RAM efficiently.

- Device Drivers : Acts as an interface b/w software & Hardware.
- File System Management : Manages how data is stored & retrieved.
- Network Management : Handles communication b/w systems.

(c). Shell (Command Line Interface - CLI)

- A command interpreter that allows users to interact with the kernel.
- Examples : Bash, Zsh, Fish, Dash, etc..
- Converts user commands into system calls for the kernel.

(d). User Applications.

- End-user programs like web browsers, text editors, DevOps tools, etc.
- Applications interact with OS using system calls via the shell or GUI.

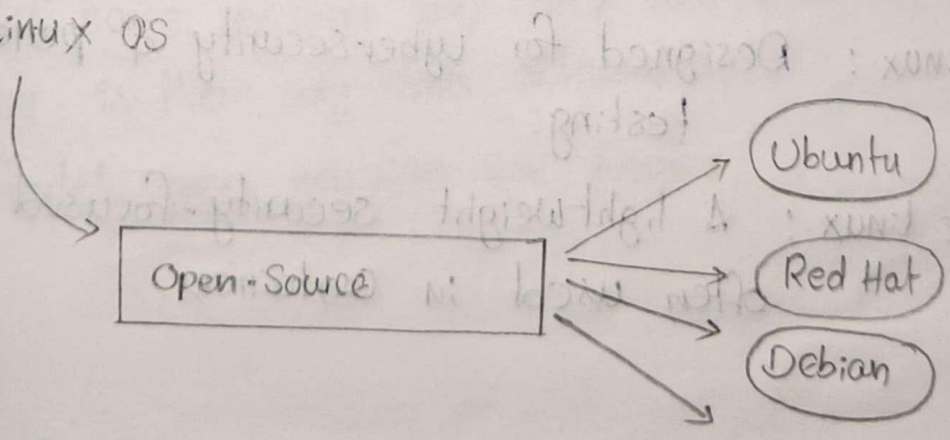
Linux Distributions.

Linux is open-source. So, the people Linux & other contributors, they have developed Linux & they made it open-source. So when it is open source, your free to download the code & you can add more features to it or remove certain features to it & ship as a product.

→ That is where companies like Conanical saw the opportunity, they take copy of the open-source linux they add some more features, they add more wrappers on top of linux OS & provide that as Ubuntu linux distribution.

→ There is another company called Red Hat which also does the same thing. & provide that as Red Hat OS.

→ Similarly, Debian, Alpine, Fedora all of them do the same thing.



Linux distributions (distros) are different versions of Linux that package the linux kernel with various s/w, system utilities & package managers. Each distro is designed for different use cases, such as personal computing, server management, or security.

Here are some popular linux distributions:

Ubuntu: One of the most beginner-friendly distros, widely used for personal & server use. it has great community support.

CentOS (discontinued, replaced by AlmaLinux/Rocky Linux) previously a popular choice for servers, based on Red Hat Enterprise

Debian: A very stable and reliable distro, often used as a base for other distros like Ubuntu.

Fedora: A cutting-edge distro that introduces new features before they reach RHEL.

Arch Linux: A lightweight, rolling-release distro for advanced users who like customization.

Kali Linux: Designed for cybersecurity & penetration testing.

Alpine Linux: A lightweight, security-focused distro often used in containers.

Linux Setup

How to install Ubuntu on My machine?

if you are using Windows machine

→ one simple thing that you can do is install WSL (Windows Subsystem for Linux).

PowerShell (or) Command Prompt.

```
wsl --install
```

→ when you run wsl --install, it create a linux like environment on your windows machines. it doesn't create a virtual Machine, but it linux like env on your windows machine.

→ once this is done, just restart your machine (or) reboot your machine. & go to your command prompt (or) Powershell & run WSL again. when you do this, you have Ubuntu running & you can access ubuntu machine on your windows.

NOTE -

What if some reasons you can't run WSL on your machine. is there any other alternative?

Definitely yes, you can use Docker.

Docker Command to Run Ubuntu Linux Container in Windows host (Persistent & Long-Term).

```
mkdir ubuntu-data # create a folder
cd ~
```

```
docker run -dit `
  --name ubuntu-container `
  --hostname ubuntu-dev `
  --restart unless-stopped `
  --cpus="2" `
  --memory="4g" `
  --mount type=bind,source="<source-path>",target=/data `
  -v /var/run/docker.sock:/var/run/docker.sock `
  -p 2222:22 `
  -p 8080:80 `
  --env TZ=Asia/kolkata `
  --env LANG=en_US.UTF-8 `
  ubuntu:latest /bin/bash
```

```
docker exec -it <cont-id> /bin/bash # Enter into Ubunt
```

```
docker ps # List running Docker containers
```

Package Manager

if you're using windows machine, you can go the internet, download Packages.

How to install Packages on Linux ?

Here we are using ubuntu, the package manager is apt.

→ so apt is installed out of the box. so you can just run apt list for example, it shows list of packages that are available on this machine.

Note- first time your installing packages, you need to run `apt update`

```
# installing python package on Ubuntu linux
apt install python3
```

What is a Package Manager ?

A package Manager is a tool automates the process of installing, updating, configuring, and removing software in a linux system. it ensures that s/w & it's dependencies are managed efficiently.

Popular Package Managers in Linux.

Linux Distro	Package Manager	Command Example
Ubuntu, Debian	apt (Advanced Package Tool).	sudo apt install nginx
Fedora, RHEL, CentOS	dnf (or yum for older version)	sudo dnf install nginx
Arch Linux	pacman	sudo pacman -S nginx
openSUSE	zypper	sudo zypper install nginx

Day - 2

Folder Structure

Understanding the folder structure

By default when you set up a linux machine there are a lot of folders that are available on that linux machine.

Let's try to understand what is the significance of that folders.

- sbin → usr/sbin

→ you will see that there are a lot of system binaries in this particular folder.

What are these system binaries ?

basically the commands or the binary files that you can use to manage your system. as an administrator.

Example - useradd

↓

this particular command (or) this particular binary helps you to create the users

→ if you will use this command & when you run this command, this actually goes to this particular folder that is sbin followed by useradd

→ and from here this command gets executed & a user is created

- lib → usr/lib

→ it is library folder

→ you will see there are different kinds of libraries that are available here & these libraries are used by your Linux kernel.

→ so as a user you don't use these libraries these are used by Linux kernel for making system calls with the hardware or basically to execute its action.

- boot

→ boot is basically for booting your linux machine that is starting or restarting your linux machine.

- bin → usr/bin

So there is slight difference b/w sbin & bin

What is difference b/w sbin & bin?

→ sbin stands for system binaries whereas bin stands for user binaries.

→ so if i go to ls /bin so you will find commands here as well but these are not administrative commands or these are not the system commands.

→ Maybe you can grant access to these commands to your regular users as well.

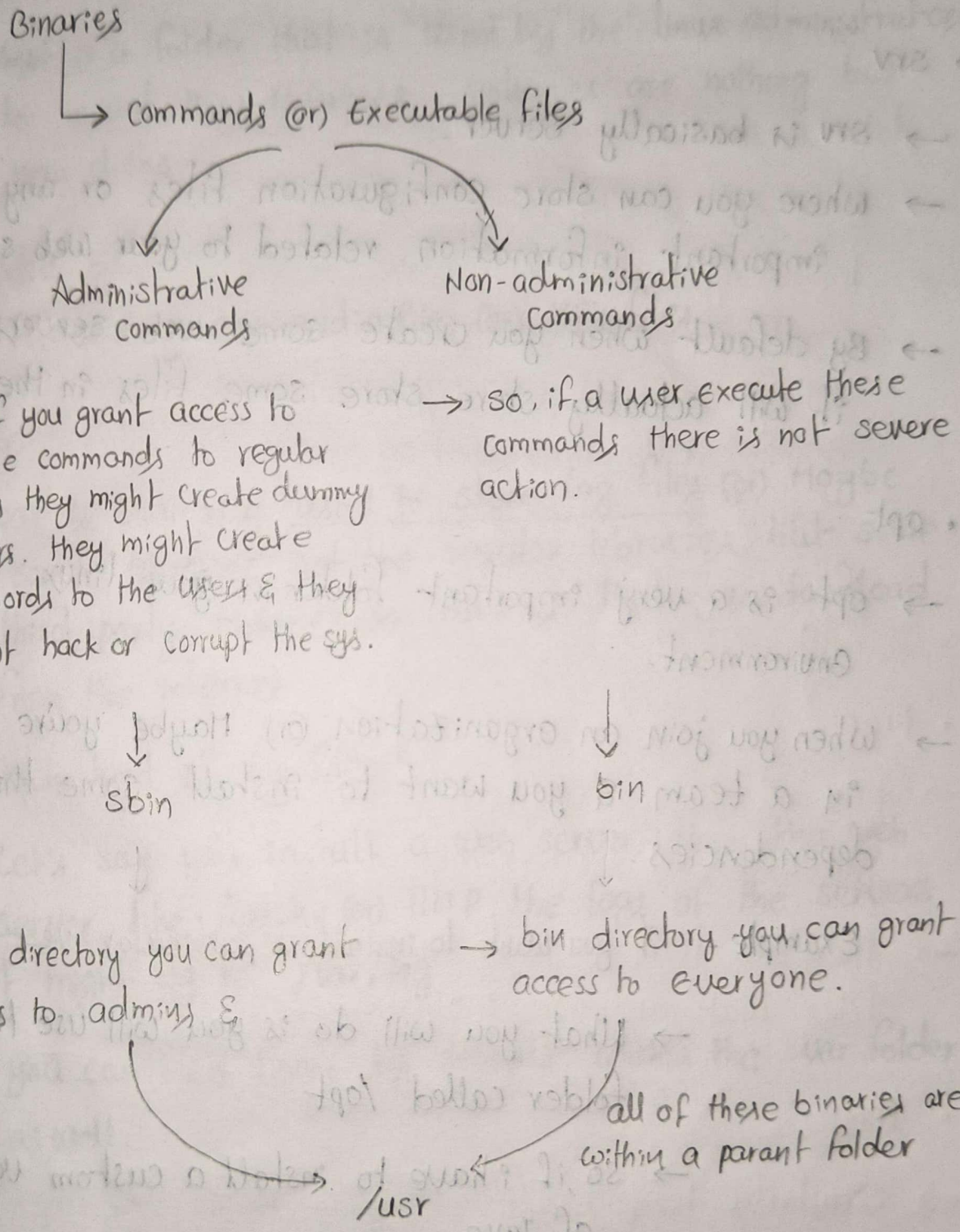
Example - date

↓

it will just print the date & time on this linux machine.

Now, this doesn't have to be administrative action, right?

Anybody can get access to this particular beinaries or this particular commands



- usr

→ the important things within usr (or) user directory is basically bin, lib, sbin & shortcuts are created for all three of them

- srv

→ srv is basically server.

→ where you can store configuration files or any important information related to your web server.

→ By default when you create some web servers it will actually store store some files in the srv.

- opt

→ opt is a very important folder in your linux environment.

→ when you join an organization (or) Maybe you're working in a team & you want to install some third-party dependencies.

Example - if you want to install a particular version of Ja

→ What you will do is you will use this folder called /opt

→ so, if i have to install a custom version of Java,

→ in /opt folder you will create a directory
mkdir custom-tool
& switch to that tool cd custom-tool
& all the related to custom tool are place them /opt/custom-tool

→ so, this is common location for all your third party dependencies,

- mnt

→ mnt means mount.

→ /mnt is a folder that is used by the linux administrators to mount new volumes. Volumes are nothing but new disks.

- media

→ for adding any audio files (or) MP4 files

- var

→ var is basically used to store log files (or) Maybe some libraries not the regular libraries but some third party libraries (or) libraries that you download from the internet

Which kind of log files?

→ Let's say you install a web server. Now this web server like Apache (or) HTTP the logs of the servers it might log to /var/log.

→ you can find them by default within the var folder as well.

- home

→ if you create a user by using adduser

→ within home you will find the users, Now within these folders (users), you can create files that are specific to that user.

- data

→ This is basically for storing any data.

→ let's say i want to share the data with other people (or) i have data related to organizational billing information, organizational cloud cost (or) people in the organization any data that can be shared with other peoples in the organization (or) other Admins in the organization you can put that in /data.

- proc

→ proc is basically virtual file system for your processes & system information.

→ They are temporary folders where you can store files that are not permanent.

- temp

→ Similarly, temp as well, Temp is used the name itself says temporary.

→ imagine you want a have a file that only lives for certain time. instead of deleting it after using, what you can do is you can create that files (or) folders within tmp.

- root

→ root is the home directory of the root.

- run

→ run basically stores the runtime data of the processes. That is while a process is running if it has any runtime data that is stored within the run folder.

- etc

→ it is one of the most important folder of your linux system.

→ within etc you have a lot of system configuration files.

→ it's like c drive (C:/) in windows, within which you have all the system configuration files.

→ Similarly, in linux, you have a folder called etc within which you have all the system configuration files.

→ what does that means? Using these files, you can configure (or) Modify your system. These are just like your settings on your mobile phone (or) settings on your laptop.

Example - passwd

↓

Using this you can modify password of any user of linux machine.

Explanation of System Directories

Symbolic Links (Less Significant)

/sbin → /usr/sbin	system binaries for administrative commands (linked to /usr/sbin)
/bin → /usr/bin	Essential user binaries (linked to /usr/bin)
/lib → /usr/lib	Shared libraries & kernel modules (linked to /usr/lib)

Important System Directories

/boot	Stores files needed for booting the system (not relevant in containers).
/usr	Contains most user-installed applications & libraries.
/var	Stores logs, caches, and temporary files that change frequently.
/etc	Stores system configuration files.

User & Application-Specific Directories

/home	Default location for user home directories
/opt	Used for installing optional third-party s/w.
/srv	Holds data for services like web servers (rarely used in Containers).
/root	Home directory of root user.

Temporary & volatile Directories

/tmp	Temporary files (cleared on reboot)
/run	Holds runtime data for processes.
/proc	Virtual filesystem for process & system information.
/sys	Virtual filesystem for hardware & kernel information.
/dev	Contains device files (e.g., /dev/null, /dev/sda)

Mount Points

/mnt	Temporary mount point for external filesystem.
/media	Mount point for removable media (USB, CDs)
/data	Likely you mounted volume from Windows (C:/ubuntu-data)

Day-3

User Management
File Management
Vi Editor shortcuts

User Management

→ Linux server is not at all safe when there is no user management.

→ linux is a multi-user os, meaning multiple users can operate a system simultaneously, Proper user management ensures security, controlled access, & system integrity.

key files involved in user management

/etc/passwd - stores user account details
/etc/shadow - stores encrypted user passwords.
/etc/group - stores group information.
/etc/gshadow - stores secure group details.

Creating User in Linux

useradd username

Ex- useradd tagore

→ However, a user without password is of no use

create password for user
passwd username

Ex- passwd tagore

→ passwords are stored as encrypted in /etc/shadow

Q. If you have access to linux server, can you decrypt passwords of users on the linux machine.

(or)

Q. As an administrator you shared username & password to a developer after 1 month developer forgot the password. So can you restore the old password of the developer?

Sol- No,

Although you have encrypted passwords here, this is very highly encrypted. this is oneway encryption.

Delete User

userdel username

Ex- userdel tagore

→ we have created user with useradd but details of the user was not saved.?

sometimes we want to take the details of the user for example full-name of the user to which group user belongs to & also want to create a home directory for the user.

adduser username #it will create in home directory.

(a) What is the difference b/w useradd & adduser?

→ Both of these commands does the same thing. However adduser creates /home directory for user. it also take a lot of user information.

→ whereas useradd is a quick way of creating the users where it does not prompt for details. it does not ask user information (or) it does not even create the home directory.

(a) What is the use of useradd? b/c adduser looks to be more powerful.

→ useradd is very useful when you are writing scripts. b/c of you don't want tool or the command to prompt inputs or to ask inputs from you.

Groups

Why do we need groups?

if you have five users in the organization, you can manage it through the users.

→ if all users are related to one env, instead of doing giving (or) retrive permissions user level, you will just go to the group & you will modify the permissions of group.

creating a group

groupadd groupname

Ex - groupadd devops

adding user to group

usermod -aG groupname username

Ex - usermod -aG devops tagore

File Management

File & Directory Management

1. ls - Lists files & directories in the current location.
2. cd /path/to/directory - changes the working directory
3. cd .. - move up one directory level (or) Back button
4. pwd - prints the current working directory.
5. mkdir new_folder - Creates a new directory
6. rmdir empty_folder - Remove an empty directory.
7. touch filename - Creates a file
8. rm filename - Deletes a file
9. rm -r folder_name
   rm -rf folder_name - Delete a folder & it's contents

   -r : Recursively deletes the folder & its contents.
   -f : forces deletion without asking for confirmation.
10. cp file1.txt file2.txt - Copies a file
11. cp -r source_directory destination_directory - Copies a directory recursively.
12. mv old_name new_name - Moves (or) Renames a files (or) directory.

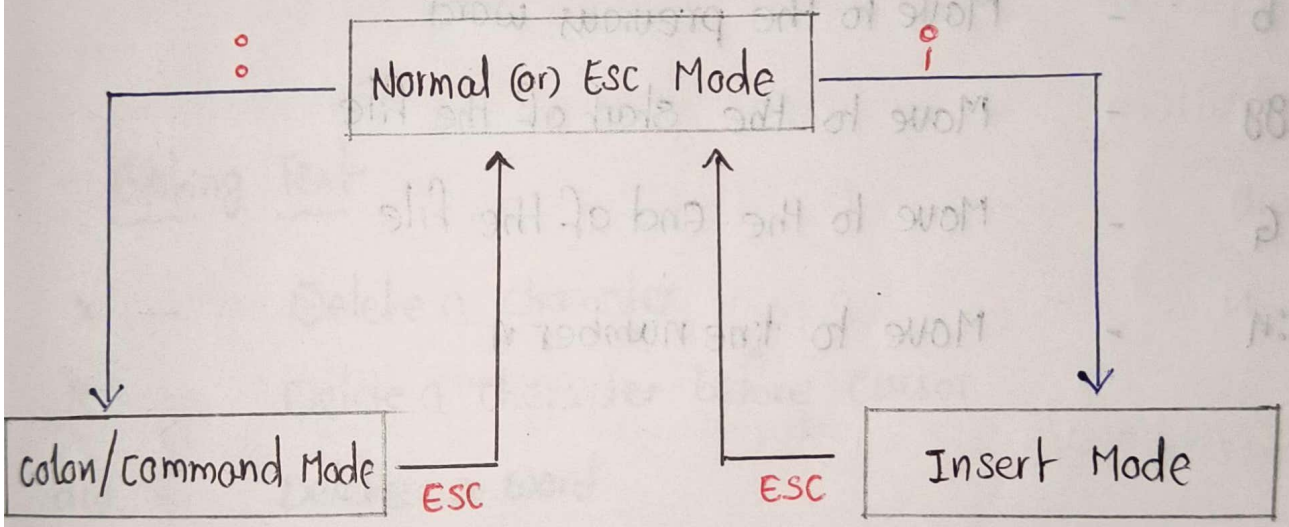
File Viewing & Editing

1. cat file.txt - Displays file content
2. tac file.txt - Displays file content in reverse orde
3. less file.txt - opens a file for viewing with scrolling support.
4. more file.txt - Similar to less, but moves forwa
5. head -n 10 file.txt - Display the first 10 lines of a fi
6. tail -n 10 file.txt - Display the last 10 lines of a file
7. nano file.txt - opens a simple text editor
8. vi file.txt - opens a powerful tex editor
9. echo 'Hello' > file.txt - Writes text to a file, overwriting exiting content
10. echo 'Hello' >> file.txt - Appends text to a file without overwriting.

VI Editor Shortcuts

Modes in VI Editor

- Normal Mode (or) Esc Mode (default) - Used for navigation & command execution.
- Insert Mode - Used for text editing (Press i to enter, Esc to exit).
- Command Mode - Used for saving, quitting & searching (press : in Normal Mode).



Basic Navigation

h	-	Move left
l	-	Move right
j	-	Move down
k	-	Move up
0	-	Move to the beginning of the line
^	-	Move to the first non-blank char of the line
$	-	Move to the end of the line
w	-	Move to the next word
b	-	Move to the previous word
gg	-	Move to the start of the file
G	-	Move to the end of the file
:n	-	Move to line number n

Insert Mode Shortcuts

i - insert before cursor

I - insert at the beginning of the line

a - Append after cursor

A - Append at the end of the line

o - opens a new line below

O - open a new line above

Esc - Exist insert mode

Editing Text

x - Delete a character

X - Delete a character before cursor

dw - Delete a word

dd - Delete a line

d$ - Dete from cursor to end of line

d0 - Dete from cursor to ~~begging~~ beginning of line

D - Delete from cursor to end of line

u - Undo last action

Ctrl+r - Redo an undone change

yy - Copy (yank) a line

yw - Copy (yank) a word

P - Paste after the cursor

p - Paste before the cursor

Search and Replace

/pattern - Search forward for a pattern

?pattern - Search backward for a pattern

n - Repeat last search forward

N - Repeat last search backward

:%s/old/new/g - Replace all occurrences of "old" with "new".

:s/old/new/g - Replace all occurrences in the current line

Working with Multiple files

:e filename	-	Open a new file
:w	-	Save file
:wq	-	Save & exit
:q!	-	Quit without saving
:split filename	-	Split screen horizontally & open another file
:vsplit filename	-	Split screen vertically.
Ctrl + w + w	-	Switch b/w split screens.

Day - 4

File Permissions

File Permission Management

Why file permissions is required on a linux server?

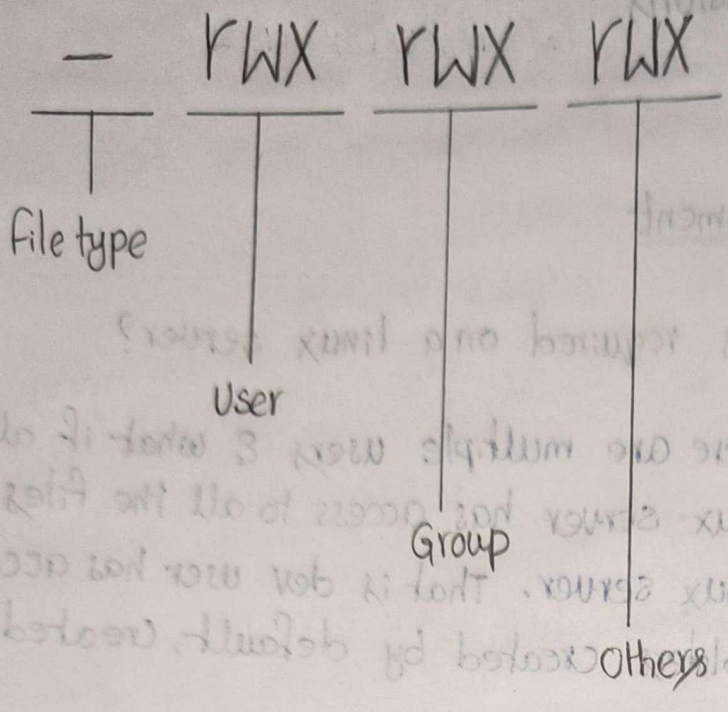
On a linux server there are multiple users & what if all the users on the linux server has access to all the files & folders on the linux server. That is dev user has access to all the files & folders created by default, created by other users.

Similarly, QE1 user has access to all the files & folders by default & created by dev user

This is going to be a huge problem. Why?
What if a user deletes important folders like /etc or /sbin
This is going to corrupt the entire file system, not only important folders.

→ so, file permission is a feature that complements user management of Linux.

→ By default, linux provides permissions on each every file (or) folder, if you want to modify it a/c to your requirements, you can use commands like chown, chmod



Read (r) → Allowable to read, Numeric value = 4

Write (w) → Allowable to write, Numeric value = 2

Executable (x) → Allowable to executable, Numeric value = 1

\- → No Permission Set

File type

\- → regular file

d → directory

l → Symbolic link

Symbolic Mode

Sr.No.	chmod operator & Description
1.	+ Adds the designated permission(s) to a file (or) directory
2	- Removes the designated permission(s) from a file (or) directory
3.	= Sets the designated permissions

Numbering format for

→ the second way to modify permissions with the chmod command is to use number to specify each set of permissions for the file.

Number	Octal Permission Representation	Ref
0	No permission	---
1	Execute permission	--x
2	Write permission	-w-
3	Execute & Write permission: 1 (execute) + 2 (write) = 3	-wx
4	Read permission	r--
5	Read & execute permission: 4 (read) + 1 (execute) = 5	r-x
6	Read & write permission: 4 (read) + 2 (write) = 6	rw-
7	All permissions: 4 (read) + 2 (write) + 1 (execute) = 7	rwx

Changing Permission with chmod

Changes files or directory permissions. It controls who can read, write, or execute a file

Using Symbolic Mode

Modify permissions using symbols

→ Add (+), remove (-), set (=) permissions

Examples -

```
chmod u+x filename      # Add execute for user
chmod g-w filename      # Remove write for group
chmod o=r filename      # Set read-only for others
chmod u=rwx, g=rx, o= filename  # set full access for user,
                                read/execute for group, &
                                no access for others
```

Using Numeric (octal) Mode

Each permissions have a value.

→ Read (+), Write (2), Execute (1).

Examples -

```
chmod 755 filename  # User (rwx), Group (r-x), others (r-x)
chmod 644 filename  # User (rw-), Group (r--), others (r--)
chmod 700 filename  # User (rwx), No access for group & oth
```

Changing ownership with chown

Changes the owner of file or directory.

Modify file owner & group.

```
chown newuser filename  # change owner
chown newuser:newgroup filename  # change owner & group
chown :newgroup filename  # changes only group.
```

Recursively changes ownership of a directory & its contents

```
chown -R newuser:newgroup directory/
```

Changing Group Ownership with chgrp

changes the group ownership of a file or directory.

```
chgrp newgroup filename  # change group
chgrp -R newgroup directory/  # change group recursively
```

Special Permissions

SetUID (s on user execute bit)

Allows users to run a file with the owner's permissions.

```
chmod u+s filename
```

SetGID (s on group execute bit)

Files: Users run the file with the group's permissions.

Directories: files created inside inherit the group.

```
chmod g+s filename      # set on file
chmod g+s directory/    # set on directory
```

Skicky Bit (t on others execute bit)

Used on directories to allow only the owner to delete their files.

```
chmod +t directory/
```

Default Permissions : umask

umask defines permissions for new files & directories.

Check current umask.

```
umask
```

Set a new umask:

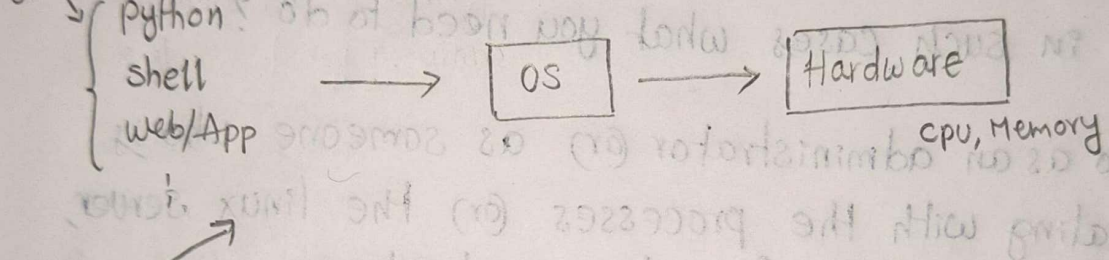
```
umask 022   # Default: 755 for directories, 644 for files
```

Day - 5

Process - Management
Monitoring
Networking
Disk - Management

Process Management

A process is a running instance of any program
for example we are running python, shell application on a linux server.



→ S/w applications (or) programs can't directly interact with hardware, that is they can't directly consume underlying CPU, Memory (or) File system. there is an intermediate layer which is called as OS.

→ However, if one of your process, let's say is CPU intensive (or) one of your process is memory intensive, then this process going to utilize most of your hardware resource.

Example -

if you are running three processes like python, shell scripting, web/App, in this one of process python consume 90% of CPU (or) Memory remaining 10% consumed by shell scripting. then there is no CPU (or) Memory resource for web/App

→ web server might return out of Memory error in the logs.

→ Now in such cases what you need to do ?

So as an administrator (or) as someone who is dealing with the processes (or) the linux server, you need to perform three activities.

1. View / check
2. Kill, Stop, Resume
3. Prioritize, deprioritize

} Process Management

① View / check

ps - list running processes

ps aux - lists all running processes

ps aux | nl - no.of running processes

ps aux | wc -l

② What is the difference b/w ps aux & ps -ef

ps aux shows you the memory utilization but ps -ef does not show you the memory.

② kill, Stop, Resume

How do you kill a process ? Why do you kill a process?

kill PID - kill / terminate a process by PID.

if you want to kill particular process for that first we fetch the PID.

```
ps aux | grep process_name.      # copy the PID
kill PID.
```

kill -9 PID - force kill a process

② What is the difference b/w kill PID & kill -9 PID ?

kill -9 PID is forcefully deleting the process.

kill -3 PID - to get the thread dump especially for java applications.

kill -STOP PID - Stop a running process (or) process is temporarily stopped

kill -CONT PID - Resume a stopped process

③. Prioritize & deprioritize

renice -n 10 -p PID - Lower priority of a process

-n

→ The nice value control a process's priority for CPU time.

The range if from -20 (highest priority) to 19 (lowest priori

renice -n -5 -p PID - increase priority of process (requires root).

What is Process Management ?

A process ~~Management~~ is a instance of a running program. Linux provides multiple utilities to monitor, manage & control processes effectively. Each process has a unique Process ID (PID) & belong to a parent process.

Index of commmands Covered

Viewing Processes.

ps aux - View all running processes

ps -u username - View processes for a specific user

ps -C processname - show a process by name

pgrep processname - find a process by name & return its PID

pidof processname - find the PID of a running program

Managing Processes.

kill PID - Terminate a process by PID

pkill processname - Terminate a process by name

kill -9 PID - force kill a process.

pkill -9 processname - Kill all instances of a process

kill -STOP PID - Stop a running process

kill -CONT PID - Resume a stopped process

renice -n 10 -p PID - Lower priority of a process

renice -n -5 -p PID - Increase priority of process (required root)

Background & Foreground Processes.

command & - Run a command in the background
jobs - list background jobs.
fg %jobnumber - Bring a job to the foreground
ctrl + z - Suspend a running process
bg %jobnumber - Resume a suspended process in the background.

Monitoring System Processes

top - interactive process viewer
htop - User-friendly process viewer (requires installation)
nice -n 10 command - Run a command with a specific priority.
renice -n -5 -p PID - Change priority of an existing process.

Daemon Process Management

systemctl list-units --type=service - List all sys daemon
systemctl start service-name - Start a daemon/servic
systemctl stop service-name - Stop a daemon/servic
systemctl enable service-name - Enable a service at startup.

Monitoring

in Monitoring is we try to check which processes are consuming more CPU, which processes consuming more memory & which processes consuming more disk. (or) which files are consuming more disk.

top	-	Real-time system monitoring.
htop	-	interactive process viewer (requires installation)
vmstat	-	Report system performance statistics
free -m	-	Show memory usage
free -h	-	Show memory usage in human readable format
nproc	-	Amount of CPU that is available on this system

Disk Monitoring

df -h	-	Check disk space usage
du -sh /path	-	show disk usage of a specific directory
du -sh *	-	Check how much disk space each item in your current directory is using.
iostat	-	Display CPU & disk I/O statistics

Network Monitoring.

ifconfig - show network interfaces (deprecated, use ip a)

ip a - show network interface details.

netstat -tulnp - show active connections & listining ports.

ss -tulnp - Alternative to "netstat" for socket statistics

ping hostname - Test network connectivity.

traceroute hostname - show network path to a host

nslookup domain - Get DNS resolution details.

Log Monitoring

tail -f /var/log/syslog - Live monitoring of system logs

journalctl -f - Live system logs for systemd-based distros.

dmesg | tail - View kernel logs.

Networking.

ping google.com - Checks connectivity to a remote server

ifconfig - Displays n/w interfaces (deprecated, use ip).

ip a - shows IP address of network interface

netstat -tulnp - Display open network connections.

curl http://example.com - fetches a webpage's content.

wget http://example.com/file.zip - Download a file from the internet.

Disk Management

Disk & Storage Management - Managing disks & storage efficiently is crucial for system performance & stability. Linux provides various commands to monitor, partition, format, mount & manage disk storage

Viewing Disk information

lsblk	-	Display block devices
fdisk -l	-	List disk partition
blkid	-	show UUIDs of devices
df -h	-	Check disk space usage
du -sh /path	-	Show size of a directory.

Partition Management

fdisk /dev/sdx	-	Create & manage partitions
parted /dev/sdx	-	Alternative to fdisk for GPT disks
mkfs.ext4 /dev/sdx1	-	format a partition as ext4
mkfs.xfs /dev/sdx1	-	format a partition as xfs

Mounting and Unmounting

sudo mount <device> <mount_point>

mount /dev/sdx1 /mnt - Mount a partition

unmount /mnt - Unmount a partition

mount -o remount, rw /mnt - Remount a partition as read-write.

Logical Volume Management (LVM)

pvcreate /dev/sdX - Create a physical volume

vgcreate vg_name /dev/sdX - Create a volume group

lvcreate -L 10G -n lv_name vg_name - Create a logical volume

mkfs.ext4 /dev/vg_name/lv_name - format an LVM partition

mount /dev/vg_name/lv_name /mnt - Mount an LVM partition

Swap Management

mkswap /dev/sdX - Create a swap partition

swapon /dev/sdX - Enable swap space

swapoff /dev/sdX - Disable swap space

Adding EBS volume to the EC2 instance.

Step-1

→ First login into instance & list the blocks by using lsblk. (or) sudo fdisk -l

```
NAME        MAJ:MIN  RM  SIZE  RO  TYPE  MOUNTPOINTS
xvda        202:0    0   8G    0   disk
├─xvda1     202:1    0   7G    0   part  /
├─xvda14    202:14   0   4M    0   part
├─xvda15    202:15   0   106M  0   part  /boot/efi
└─xvda16    259:0    0   913M  0   part  /boot
```

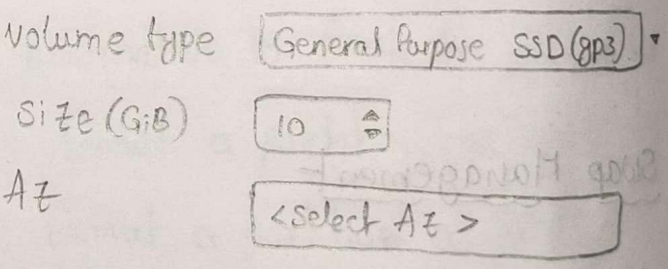
What are partitions ?

partitions are nothing but you created a volume. Volume is a type of block storage & that block storage instead of 8 GB on a pole is divided into partitions.

& each partition can be formatted & mounted to a particular location.

Step-2 Create volume.

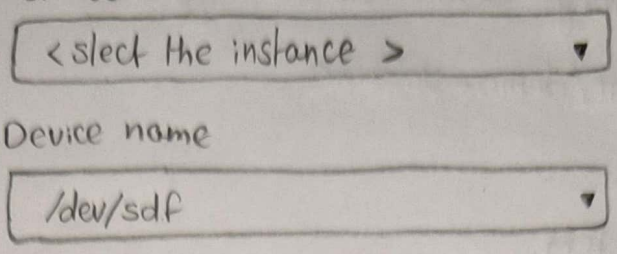
EBS → Volumes → create volume



Note - The created volume should be same Az as where instance is.

Step-3 Attach the volume to Instance.

→ Move to created volume → Actions → Attach volume



→ click on Attach volume.

Step-4 Relogin into instance & Mount the volume to instance.

→ Now re-check the list of the blocks by using lsblk.

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
xvda	202:0	0	8G	0	disk	
├─xvda1	202:1	0	7G	0	part	/
├─xvda14	202:14	0	4M	0	part	/boot/efi
├─xvda15	202:15	0	106M	0	part	/boot
└─xvda16	259:0	0	913M	0	part	
Xvdf	202:80	0	10G	0	disk	

NOTE: You can't directly use this 10GB at this point, b/c of two reasons.

- it is off block storage
- you haven't mounted this yet.

→ Block storage you can't use it. Only you have to split this into partitions. format it to particular file sys

only then you can use it.

→ even if you don't divide this to partitions, it's okay. But you need to format it

creating directory before formating

```
mkdir -p /mnt/demo-volume
ls -ltr /mnt
```

make a file system

```
mkfs -t ext4 /dev/xvdf
```

Mount

```
mount /dev/xvdf /mnt/demo-volume/
```

→ Now recheck `lsblk`

```
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda    202:0    0    8G  0 disk
├─xvda1 202:1    0    7G  0 part /
├─xvda14 202:14  0    4M  0 part
├─xvda15 202:15  0  106M  0 part /boot/efi
└─xvda16 259:0   0  913M  0 part /boot
xvdf    202:80   0   10G  0 disk /mnt/demo-volume
```

10 Linux command every s/w engineer should know

1. TOP [system performance]

what does top do ?

it helps you monitor system performance in real time.

Ex - Monitor CPU utilization,
Memory utilization,
Active processes checking,

top command outputs are -

PID USER %CPU %MEM TIME+ COMMAND

htop → it similar just like top b/c it's just a wrapper on top of top.

- it gives you a better readability.

2. PS [Process related info]

→ it is used to list running processes, process ID and also you can visualize the process hierarchy.

PS aux → it provides you a very detailed info, but it is not real time.

pstree -p → we can see hierarchy of all the processes.

3. netstat [N/w connection]

→ it is used to inspect the n/w connections. Especially if you want to know if a port is available or not.

netstat -tuln → we will see the ports that are already in use.

4. tcpdump [latency issues checking] [N/w troubleshooting]

→ it will capture and analyze n/w package for diagnizing the connectivity issues.

sudo tcpdump -i enX0 port 80

(or) eth0

ifconfig

tcpdump -i enX0

→ we will receive the complete n/w packet information of your primary n/w interface.

5. Ping, traceroute [instance is not able to connect to the internet, this is the first place that you would start with]

6. Disk utilization

df -h → to see how much this space is left
→ show disk space usage of all mounted filesystems.

check utilization of a particular folder, what is the size of this particular folder (ex- opt)

du -sh opt
→ show total disk usage of a folder (-s for summary, -h for human-readable)

7. Memory utilization

free -h → check the memory utilization

8. journalctl [systemd related logs]
→ we will get the complete logs of the services.

check for particular service

journalctl -u nginx

see the complete logs from the last boot or from the last restart of your linux machine

journalctl -b

9. List of open files.

how to check the process that is using Particular port. we can use during conflict.

```
lsof -i :8085
```

→ Using this command you can check who has used this port (or) who is using this particular port.

10. Log Analysis

default log files /var/log

read logs

```
sudo cat /var/log/auth.log
```

read last 10 lines of this log file

```
sudo tail -n 10 /var/log/auth.log
```

read first 10 lines of this log file

```
sudo head -n 10 /var/log/auth.log
```

Shortcuts

history → lists past commands with line numbers

CTRL + R → Reverse search on your history (type to find past commands)

export

Ex- `export PS1="Linux Commands $:"`

`export PS1="$PWD $:"`

!n → Run command number n from history

!! → Re-run the last command (very useful after adding sudo !)

!keyword → Runs the most recent command starting with keyword.

Ctrl + A	Move cursor to the beginning of the line
ctrl + E	Move cursor to the end of the line
ctrl + U	Delete from cursor to beginning of the line
ctrl + K	Delete from cursor to end of the line
ctrl + W	Delete the word before the cursor
Ctrl + L	Clear the terminal screen (like clear command)
ctrl + C	Terminate the current command/process
ctrl + D	Log out or exit the terminal
ctrl + Z	Suspend the current process (can resume with fg
ctrl + R	Reverse search through command history
Tab	Auto-complete commands, files, or directories
Up/Down Arrow (Ctrl+P, Ctrl+N)	Navigate through command history
ctrl + Shift + C	Copy selected text in terminal
Ctrl + Shift + V	Paste copied text in terminal
ctrl + Shift + N	Open a new terminal window.
Ctrl + Shift + T	Open a new terminal tab.
ctrl + Shift + Q	Completely close the terminal window.
Ctrl + Y	This will paste the erased text that you saw with Ctrl+W, Ctrl+U, Ctrl+K shortcuts.
Alt + .	Use the last argument of the previous comman
Alt + B	Move the cursor one word backward
Alt + F	Move the cursor one word forward
Alt + D	Delete the word after the cursor
Alt + Delete	Delete the word before the cursor