[2nd September 2025 Lab Day](https://www.youtube.com/watch?v=c_a2hBgK2KE)

|  |  |
| --- | --- |
| 60 basic question answers | Page 2 |
| **File Permission and Commands** | Page 5 |
| **Change root password** |  |
| Oracle VM: VM1, VM2 – Connected to Putty |  |
| Oracle VM: VM1, VM2 – Connected to SSH |  |
| aws: control\_node, node1, node 1 – Connected to Putty |  |
| aws: control\_node, node1, node 1 – Connected to SSH |  |
| Change root password by restarting reboot VM. |  |
| **Creating Web Site** |  |
| Creating Ansible: control\_node, node1, node2 |  |
| **Add hard Disk** |  |
| **Hard Link and Soft Link** |  |
| **Hostname Change** |  |

**HOSTNAME CHANGE:**

#hostnamectl set-hostname controlnode

#exec /bin/bash

**Add hard Disk**

>setting --- storage

>controller SATA – icon ‘add hard disk’

#dh -hT

#lsblk

#fdisk /dev/sdb (p-print, w-write, n-new/add, q-quit)

n (press enter)

partition: 1

w (press enter)

>partition has been created

#mkfs.ext4 /dev/sdb1 (create file system)

#mkdir -p /mnt/data

#mount /dev/sdb1 /mnt/data

#df -hT

**Hard Link and Soft Link:**

#ln original.txt hdlink.txt (hard link)

#ln -s original.txt softlink.txt (soft link)

**Inode:** unique identifier in linex file system, every file and directory having unique Inode.

**Hard link**: create the same inode# of both files. If original.txt deleted then other remains.

**Soft link:** it create the different inode# of both files. If orginal.txt deleted then other files contents also deleted.

**Note:** in different file system like xfs and ext the hard link not created.

**Creating Web Site:**

$systemctl status httpd

$dnf install httpd –y

$systemctl enable httpd

$systemctl start httpd

$cd /var/www/tmp

$wget <https://freewebsitetemplates.com/download/files/lawfirm.zip>

$sudo unzip /var/www/html/tem/lawfirm.zip –d /var/www/html/lawfirm

Open browser: http://3.110.207.104/lawfirm/

Tip: ctrl+shift+v copy the link

**Change root password:**

* Reboot Virtual Machine
* Press ‘e’ when on ‘greb’ screen
* Add ‘rd.break’ at the end, where line start with ‘Linux…’
* Ctrl + x
* switch\_root:/# mount –o remount,rw /sysroot
* chroot /sysroot
* passwd
* touch /.autorelabel
* exit
* exit

| **No.** | **Question** | **Answer** |
| --- | --- | --- |
| 1 | What file contains the operating system name and version? | /etc/os-release |
| 2 | Which command displays how long the system has been running? | uptime |
| 3 | What command is used to display the system date and time? | date |
| 4 | What does the command 'hostname -i' display? | The system's IP address |
| 5 | How can you change the hostname permanently in Linux? | hostnamectl set-hostname newname |
| 6 | Which command shows detailed kernel version and system architecture? | uname -a |
| 7 | What is the output of 'uname -r'? | Kernel version |
| 8 | Which command provides CPU architecture and hardware info? | lscpu |
| 9 | Which command shows available and used memory in megabytes? | free -m |
| 10 | Which command shows disk usage in human-readable format? | df -h |
| 11 | Which command shows all network interfaces and IP addresses? | ip a |
| 12 | What is the purpose of the command 'pwd'? | Prints working directory |
| 13 | What does 'ls -l' display? | File permissions and ownership |
| 14 | How does 'ls -ld' differ from 'ls -l'? | Displays directory metadata only |
| 15 | What does 'ls -il' display? | Inode numbers |
| 16 | Which directory contains configuration files? | /etc |
| 17 | Which Linux directory is known to grow because of logs? | /var |
| 18 | Which of the following is typically found in /boot? | initrd, vmlinuz |
| 19 | What is stored in the /home directory? | User files and personal settings |
| 20 | Which of the following is true about /dev? | Contains device files like sda, tty, and sr0 |
| 21 | How do you clear the terminal screen? | clear |
| 22 | Show last 5 lines of a file: | tail -n5 file |
| 23 | Delete a file: | rm file |
| 24 | Move (cut-paste) a file: | mv file /path/ |
| 25 | Which command shows your current working directory? | pwd |
| 26 | Page through a file with search support: | less file |
| 27 | Delete an empty directory: | rmdir dir |
| 28 | Show contents of a text file quickly: | cat file |
| 29 | Copy contents of fileA to fileB: | cat fileA > fileB and cp fileA fileB |
| 30 | View a file page by page (older pager): | more file |
| 31 | Which command prints the current logged-in user? | whoami |
| 32 | Create an empty file or update mtime: | touch file |
| 33 | Change directory one level up: | cd .. |
| 34 | Copy a file to another directory: | cp file /path/ |
| 35 | Create a directory: | mkdir dir |
| 36 | Which command lists files in the current directory? | ls |
| 37 | Which command shows system date/time? | date |
| 38 | Show first 5 lines of a file: | head -5 file and head -n 5 file |
| 39 | Rename a file: | mv file newname |
| 40 | Edit a file (choose a CLI editor): | nano file and vi file |
| 41 | What command helps you change directory? | cd |
| 42 | What is the command and flag for copying in Linux? | cp -r |
| 43 | How do you install httpd package in Linux? | dnf install httpd -y |
| 44 | Before the 'httpd' package is installed, does the service already exist on the system? | No |
| 45 | What does the 'd' stand for in 'httpd'? | daemon/service |
| 46 | Give the full command to check the status of a daemon/service if it is running or not. | systemctl status httpd |
| 47 | How do you start a httpd service? | systemctl start httpd |
| 48 | Home Directory of root is '/' yes or no? | no |
| 49 | What is the Home Directory of User 'natasha' (give the full absolute path)? | /home/natasha |
| 50 | If you want to see all the hidden files under a directory what command will you use? | ls -al |
| 51 | You have a VM provisioned on your VirtualBox. What is the name AWS gives to its VMs? | EC2 |
| 52 | The default user in AWS when you create an EC2 Instance is? | ec2-user |
| 53 | Can you copy files from Windows to Linux using 'WinSCP'? | yes |
| 54 | The Home Directory of root is? | /root |
| 55 | You want to copy 'law-firm.zip' from /home/ec2-user to root's home directory | cp -r /home/ec2-user/law-firm.zip /root |
| 56 | Give the full path of the directories where Webpages are hosted. | /var/www/html |
| 57 | What command is used to make a file? | touch |
| 58 | What command is used for making a directory? | mkdir |
| 59 | How do you delete all files in a folder? | rm -rf \* |
| 60 | Are you learning something new in this class? | yes |

**File Permission and Commands:**

two ways to change permission:

**Symbolic:** Read/write/execute (separate change the permission)

**Absolute**: change permission through one command

**Authorization:** two type of authorization allowed

1. Ownership
2. Permission

**OWNER:** the user who create the file is the owner of that file.

**GROUP:** if user is group member and create file then other user of that same group are owner of that file.

**ACL:** It allow more specific set of permission to a file or directory without changing owndership and permission.

Command user: setfacl / getfacl

#getfacl abc.txt

#setfacl -m u:user:rwx abc.txt

**folder with directories and files and permission**

#useradd noman

#passwd noman

#mkdir /work

#touch /work/abc.txt

#chmod o+w /work/abc.txt (given file permission)

#chmod o+w /work (given directory permission)

#su - noman

$echo “this is my commants – noman” >> /work/abc.txt

$cat /work/abc.txt

**File Delete:**

#rm abc.txt **Prompts for confirmation** if the file is **write-protected** (depending on settings).

#rm –r abc.txt **Recursively deletes directories and their contents**.

#rm –f abc.txt (**Force delete,** no confirmation)

**Directory Delete:**

#rmdir /work

#rm –r /work (directory with contents)

#rm –rf /work (directory – force delete)

**User Delete:**

#userdel noman

#userdel –r noman (remove user with directories and files)

#kill -9 process#

**Check if the user still exists in system files**

#grep noman /etc/passwd

**Group Delete:**

#groupdel (Delete a group)

#gpasswd –d u1 workgroup (remove user u1 from workgroup)

**Option 1: Give write permission to 'noman' using ACL (best for single user access)**

#getfacl /work/abc.txt (check the permission)

#setfacl -m u:noman:w /work/abc.txt (given write permission to user noman)

#setfacl -m u:noman:wr /work/abc.txt (given write permission to user noman)

#setfacl -m u:noman:rw -m g:devs:rw -m o::rw /work/abc.txt (given to user, group, other for read/write)

**Option 2: Change group ownership (for shared group access)**

**#groupadd workgroup**

**#usermod –aG workgourp noman**

**#chown root:workgroup /work/abc.txt**

**#chmod 664 /work/abc.txt**

**#getent group /**

**#getent group | grep '^w'**

**#groups u1 (to check user ‘u1’ is the member of group ‘abc’)**

**#gpasswd –d u1 abc (removing user u1 from group abc)**

**Option 3: Just make the file world-writable (not recommended for production)**

#chmod o+w /work

#chmod ug+rw /work/abc.txt (set **read and write** for both user and group)

#chmod ugo+rwe /work/abc.txt (set **read, write** and **execute** for user, group and others)

**Group: used to manage permissions (read, write, execute) for multiple users at once.**

**Command Description Example**

#getent group list all groups

#groups u1 specific user groups

#id u1 detailed info i.e. 10

#getent group 10 wheel:x:10:sysadmin,noman,user1

#cat /etc/group view group file directly

#cut -d: -f1 /etc/group list group names only

#getent group wheel search specific group

#getent group | grep '^w' starting with w (group and attached users)

#group '^w' /etc/group local groups only

#newgrp wheel

#groups

#newgrp developers temporarily changes active group

**LISTING:**

#ls –R /work List inside directory

**Copy/Move:**

#cp /home/u1/abc.txt /home/u1/shared

#mv /home/u1/abc.txt /home/u1/shared

SSH:

C:\> ssh [noman@10.0.2.15](mailto:noman@10.0.2.15) open windows terminal and execute

C:\>ping 10.0.2.15

sudo cp -r /home/user1/\* /home/user2 to copy dir and files to other folder

or

udo mv /home/user1/\* /home/user3 (to move dir and files to other folder)

|  |  |
| --- | --- |
| Ping 127.0.0.1  LocalHost | ping localhost is a **basic test** to ensure that **your system's internal networking** is functioning properly. |
| **What ping localhost Does**   * localhost resolves to **127.0.0.1** (IPv4 loopback address). * ping sends **ICMP echo requests** to that address. * It waits for an **echo reply**.  Example Output PING localhost (127.0.0.1) 56(84) bytes of data.  64 bytes from localhost (127.0.0.1): icmp\_seq=1 ttl=64 time=0.025 ms  Means: your **loopback interface and networking stack are working fine.**  **If It Fails**  If ping localhost fails:   * /etc/hosts might be misconfigured * Loopback interface (lo) might be down * ICMP packets might be blocked by firewall * System’s networking stack may be damaged |
| **ICMP (**Internet Control Message Protocol) | An **ICMP Echo Request** is a type of network message used by the ping command to check if another device (host) is reachable. |
|  |  |

9999999999

1. Download Putty and install

2. Putty is used to login to other servers

3. it is open source that connect to remote system.

5. run Oracle Virtual BOx Machine

6. Run VM1 server

4. Search 'Putty' and execute it.

7. Host Name (IP Address): Enter VM1 Server IP i.e. 192.168.100.14

8. Connection type: SSH

9. Saved Sessions: VM1...Save...OPen

10. change font size: r-click on bar...Change Setting...Apperarance...Change

11. Save Logs: change Setting...Logging...Browse...set location

12. open file 'practice quiz'. which given on what's up. and copy into memory

13. open putty screen and work below

14. touch quiz01.sh (sh is batch script - create new file)

15. ls

16. vi quit01.sh ...press i (open in editor)

17. copy and peast into editor. r-Click and peast

18. Esc :wq! and press Enter (:wq! save and quit) (:qa! quit without save)

19. cat quiz01.sh

20. ls -l quiz01.sh (it has no permission x means execute)

21. chmod +x quiz01.sh (permission has been changed now it is executible)

if we put this file into temp folder than everyone can execute it.

22. ./quit01.sh (execute the file)

99999999999999

AWS Free Tier Account - create free account

https://aws.amazon.com/console/

1. Search bar 'EC2' then click on EC2 which is blue color

2. click on 'Launch INstance' in yellow

3. Name: myserver

4. Summary...Number of INstances: 2

5. Key pair (logic)...create new key pair...

6. key pair name: nitkeys... .ppk (for use with Putty)...create key pair

7. create security group

Allow SSH traffic from

Allow HTTPS traffic from the internet

Allow HTTP triffic from the internet

8. Press 'Launch Instances'

9. view all instances button

10. now see two machines...refresh button press

11. click 1st machine check box...connect...connect

12. now this is your machine (ec2 machine)...close it

13. open your putty, on menu bar r-click...new session...

14. now open aws page --- connect...

here is public ip and user name is ec2, copy the ip and return Putty

15. paest IP in IP addrss like this username@IP i.e. ec2-user@216.216.54.55 (its aws ip)

16. Name: aws...save...IN left column SSH...(check video 1:35) link https://www.youtube.com/watch?v=vAIw4Nbv5\_Y

17. sudo su -

cd /var

ll

there is no HTML file

dnf install httpd -y (install the package

ll

systemctl enable --now httpd

systemctl status httpd now apachi http server is installed, all http, web side on web server.

we install the web server, the appachi HTTP server is now installed in machine

cd /www

ll

pwd

cd html

ll

touch index.html

vi ... i... type here 'My name is MNS and my first webside'

ecs...shift:wQ!

open browser...copy ip and pease in browser

18. goback to machine putty

systemstl stop httpd

19. execute web again... so its not work because

httpd service is down.

20. start it again systemstl restart httpd

99999999999999

00000000000000000

**What is a Hypervisor?**

A **hypervisor** is a specialized software layer that allows multiple **virtual machines (VMs)** to run on a single **physical machine (host)**. It manages the hardware resources (CPU, memory, storage, etc.) and allocates them to each virtual machine as needed.

### Why is the Hypervisor Important?

Without a hypervisor, each operating system would need its own physical hardware. The hypervisor **abstracts the hardware**, so many operating systems can share the same physical system safely and efficiently.

### Types of Hypervisors

There are two main types:

1. **Type 1 (Bare Metal Hypervisor):**
   * Runs directly on the physical hardware.
   * More efficient and secure.
   * Examples: VMware ESXi, Microsoft Hyper-V, Xen.
2. **Type 2 (Hosted Hypervisor):**
   * Runs on top of a regular operating system (like Windows or Linux).
   * Easier to set up, but less performant.
   * Examples: VMware Workstation, Oracle VirtualBox.

### What Does the Hypervisor Do?

* **Creates and manages VMs**
* **Allocates resources** (CPU, memory, disk) to each VM
* **Isolates VMs** from one another
* Allows VMs to **run different operating systems** on the same physical host

|  |  |  |
| --- | --- | --- |
| The uname command in Linux is used to display system information. |  |  |
| uname | Print system name |  |
| uname -a | All system information | Linux abc 5.14.0-570.17.1.e19\_6.x86\_64 #1 SMP PREEMT\_DYNAMIC Fri May 23 22:4747:81 UTC 2025 x86\_64 x86\_64 x86\_64 GNU/Linux |
| uname -r | Kernal release/version | 5.14.0-570.17.1.e19\_6.x86\_64 |
| uname -s | Kernel name | Linux |
| uname -n | Network domain/node hostname | abc |
| uname -v | Kernel version | #1 SMP PREEMPT\_DYNAMIC Fri May 23 22:4747:81 UTC 2025 |
| uname -m | Machine hardware name | X86\_64 |
| uname -p | Processor type (may show unknown) | X86\_64 |
| uname -i | Hardware plateform (may show unknown | X86\_64 |
| uname -o | Operating system | GM/Linux |
|  |  |  |
| Cat /etc/os\_release | To view the file in ‘etc’ directory |  |
|  |  |  |
| clear | Clear the screen |  |
| Hostname | Display hostname | abc |
| hostnamectl set-hostname myserver | Change hostname to myserver |  |
| /bin/bash | Re initializing |  |
| Bash | Re initializing |  |
|  |  |  |
| cat /etc/shells | will display the list of valid login shells available on your Unix/Linux system. | /bin/sh  /bin/bash  /usr/bin/bash  /usr/bin/sh |
| echo $SHELL | To see current shell | /bin/bash |
| Lscpu | displays **detailed CPU architecture information** about your system. | Architecture: x86\_64  CPU op-mode(s): 32-bit, 64-bit  Byte Order: Little Endian  CPU(s): 8  On-line CPU(s) list: 0-7  Thread(s) per core: 2  Core(s) per socket: 4  Socket(s): 1  Vendor ID: GenuineIntel  Model name: Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz  CPU MHz: 2112.004  Virtualization: VT-x  L1d cache: 32K  L1i cache: 32K  L2 cache: 256K  L3 cache: 8192K |
| pwd | stands for **Print Working Directory**. |  |
|  |  |  |
| total used free shared buff/cache available  Mem: 7947 1523 4321 198 2103 5930  Swap: 2047 0 2047  Swap is disk-based memory used when RAM runs out.  If used under Swap is high, you may be low on RAM.  "Swap ڈسک پر مبنی میموری ہوتی ہے جو اُس وقت استعمال ہوتی ہے جب RAM ختم ہو جائے۔  اگر Swap کے تحت 'used' کی مقدار زیادہ ہو تو اس کا مطلب ہے کہ آپ کے سسٹم میں RAM کم پڑ رہی ہے۔" | | |
|  |  |  |
|  |  |  |
|  |  |  |
| Ip a: 192.168.100.14 |  |  |
| Ssh [root@192.168.100.14](mailto:root@192.168.100.14) | To check last login |  |
| Pwd | Print working directory |  |
| Cd bin | Change directory |  |
| Cd  Cd / or cd ..  Ls (list of  Ll | Out side the door |  |
| Ctrl c | Kill process |  |
| Cd /etc |  |  |
|  |  |  |
| Login shell:  Login shell is first process  NonLogin Shell: |  |  |
| When login then bash invoked |  |  |
| pS $$  file /bin/bash | Process in kernel | What is standard input |
| Echo $PATH  Echo $USER  Useradd Ali  Passwd ali |  |  |
| System files.  .filename (hidden file) |  |  |
| Init 0 | logout |  |

111111111111

- df -h file system-connected to disk

-df -sh total size of contents

-ls -l metadata size of listed dir, type , owner, permission, user

-ls -ld

-pwd print working directory

-cat abc.txt to view abc.txt file

-su - sysadmin to change user

-history bash commands, stored in ~/.bash\_history

touch abc.txt xyz.txt

su root

sudo cp -r /home/user1/. /home/user2 copy all to other user

sudo mv /home/user1/\* /home/user3 (move all to other user

cut -d: -f1 /etc/passwd list of users

cut -d: -f1 /etc/passwd | ws -l number of users

-----

---------------------------------------------------------

-uname -a display system information

Linux rocky-linux 5.14.0-362.24.1.el9\_3.x86\_64 #1 SMP PREEMPT\_DYNAMIC Thu Jun 20 11:25:30 UTC 2024 x86\_64 x86\_64 x86\_64 GNU/Linux

name -a is useful for:

Checking kernel version

Confirming system architecture

Troubleshooting hardware/kernel issues

Providing system info for support

| Part | Description |

| ------------------------------ | ------------------------------------------------------------------------------------------------------------------------------- |

| `Linux` | \*\*Kernel name\*\* – Always "Linux" for Linux systems. |

| `rocky-linux` | \*\*Hostname\*\* – The network name of your machine (set in `/etc/hostname`). |

| `5.14.0-362.24.1.el9\_3.x86\_64` | \*\*Kernel version\*\* – Includes major, minor, patch numbers, and distro-specific identifiers (`el9\_3` means RHEL 9.3-compatible). |

| `#1` | \*\*Build number\*\* – Indicates this is the first build of the kernel. |

| `SMP` | \*\*Symmetric Multi-Processing\*\* – Means the kernel supports multi-core CPUs. |

| `PREEMPT\_DYNAMIC` | \*\*Preemption model\*\* – Refers to how the kernel handles task preemption (dynamic configuration). |

| `Thu Jun 20 11:25:30 UTC 2024` | \*\*Build time\*\* – When the kernel was compiled. |

| `x86\_64` | \*\*Machine architecture\*\* – 64-bit architecture (AMD64/Intel64). |

| `x86\_64` | \*\*Processor type\*\* – Usually the same as machine architecture. |

| `x86\_64` | \*\*Hardware platform\*\* – Again usually the same on modern systems. |

| `GNU/Linux` | \*\*Operating system\*\* – Refers to the OS as a whole (Linux kernel + GNU tools). |

-uname -r Displays the kernel release version of your Rocky Linux system.

5.14.0-362.24.1.el9\_3.x86\_64

🔹 Breakdown:

5.14.0 – Kernel version (major.minor.patch)

362.24.1 – Red Hat's internal build version

el9\_3 – Built for Enterprise Linux 9.3 (compatible with Rocky Linux 9.3)

x86\_64 – 64-bit architecture

cut -d: -f1 /etc/passwd

--------------------------------------------

-ls -ld : detailed info about directory itself. not its contents.

drwxr-xr-x 123 root root 4096 Aug 3 12:34 /etc

d=directory, -=file, l=symbolic link

rwxr-xr-x permissions (owner/group/others)

123 number of hard links to the directory

root owner of the directory

root group that owns the directory

4096 size in bytes

Aug 3 12:34 last date modified and time

/etc Directory name

drwxr-xr-x these 10 string gives imp info about file type and permission

d file type, d=directory

rwx owner (user) permissions : owner can read/write/execute

r-x group permission : group can read/write

r-x others (everyone else) permission :everyone can read/write

- permission not granted

--------------------------------------------------------

-ls -l it will list the contents of the /etc MetaData: file size, type, permission, ownere, timestamps

-----------------

man useradd help

useradd noman1

passwd noman1

su - noman1

-------------------23/8/2025

cloud archetecture

work flow:

code---github---jenkins---build---Dev---uat

ansable: automation tool

dewox

# check services(httpt)

systemctl status httpd

port 80 for http

port 443 for https

which port is listing

#netstat -tunap | grep -i listen

command not found, so download it

#dnf install net-tools

sshd: d demaion (provide file which required)

----24/8/2025

systemctl -t help

404 error service not available

#systemctl start httpd:

start, stop, restart, reload, enable, disable

Boot Time: enable and disable

Run time: start, stop, restart, reload

systemctl status httpd

systemctl status sshd

q; you are unable to ssh into a machine tell me what could be the problem.

a: ssh service--> sshd --> daemon

#firewall -cmd --list-all

#nmcli con show

service = ssh daemon = sshd

firewall allow ssh

if port 80 down then server is not working

SRE jobs: front end and back end

core business of pharmacy

client--cvs.com---DNS---get IP resolve---client

---cvs server (Data center) - IP

DHCP provide the IP to data center

dynamic host protocol DHCP

server send back bunch of files to client

backend:

frontend:

q:

#nmtui

#nmdi

prob: first think: DNS could be problem

after patching ip could be change.

check in /etc/resolve.conf on client machine

root@client]# cat /etc/resolv.conf

cpu, memory, HD

systemctl status sshd | grep -i listen

2:

SSH---putty---having IP address -- server

--unable to loggin machine

--worknode: typically login to dev environment

aws

--client machine ssh server

if not access

then server is down

31/8/2025

sda 20 g

partition:

swap partition

boot partition

fdisk utility

vbox = 5g

logical volume manager (LVM) if boot partition fill then can be add more

SYSTEM USER

NORMATL USER

NETWORK USER

SYSTEM USER

SUDO USER

22222222222222222

Root/test12#$

Sysadmin/abc

-Dotnet –info

-whereis dotnet

=>List all installed packages on Ubuntu

dpkg --get-selections

dpkg --get-selections | grep -i dotnet

=>Install .NET SDK (includes ASP.NET Core)

-wget https://packages.microsoft.com/config/ubuntu/22.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb

-sudo dpkg -i packages-microsoft-prod.deb

-sudo apt update

-sudo apt install -y dotnet-sdk-8.0

=>Verify Installation

-Dotnet –version

lsb\_release –a                                      #Ver

cat /etc/os-release                                #Ver

hostnamectl                                          #host

dotnet --info

**🖥️ System Information**

CopyEdit

uname -a         # Kernel and system info

hostname         # Show hostname

uptime           # How long the system has been running

whoami           # Show current user

top              # Real-time process viewer

htop             # Advanced version of top (install it first)

neofetch         # Stylish system info (install with `sudo apt install neofetch`)

**📁 File & Directory Commands**

CopyEdit

ls               # List files in directory

ls -l            # Detailed listing permission

cd [dir]         # Change directory

pwd              # Print current directory

mkdir [dir]      # Create new directory

rm [file]        # Remove file

rm -r [dir]      # Remove directory and contents

cp [src] [dest]  # Copy files

mv [src] [dest]  # Move or rename files

touch [file]     # Create empty file

find . -name "\*.txt"  # Find files

find . -name "\*.txt" | less    # Find files one by one wise

find . -name "\*.txt" | more    # Find files down arrow to page wise

**🧰 Package Management (APT)**

CopyEdit

sudo apt update                # Refresh package index

sudo apt upgrade               # Upgrade all packages

sudo apt install [package]    # Install a package

sudo apt remove [package]     # Remove a package

sudo apt purge [package]      # Remove including config

sudo apt autoremove           # Remove unused packages

dpkg -i [package.deb]         # Install a .deb file

**🧑‍💻 User Management**

CopyEdit

adduser [name]        # Add new user

sudo passwd [user]    # Set or change password

usermod -aG sudo [user]   # Give user sudo rights

who                   # Show who is logged in

**🔐 Permissions**

CopyEdit

chmod +x [file]       # Make file executable

chmod 755 [file]      # Set permission

chown user:user [file] # Change ownership

**🔎 Searching**

CopyEdit

grep "text" [file]    # Search in file

grep -r "text" .      # Recursive search in directory

find / -name [file]   # Find file by name

**🌐 Networking**

CopyEdit

ip a                 # Show IP addresses

ping [host]          # Check network connection

ifconfig             # Network interfaces (install with `net-tools`)

curl [url]           # Fetch URL

wget [url]           # Download file

netstat -tulpn       # Show open ports (install `net-tools`)

**📦 System Monitoring & Processes**

CopyEdit

ps aux               # Show all running processes

kill [PID]           # Kill a process

kill -9 [PID]        # Force kill

df -h                # Disk usage

du -sh \*             # Directory size

free -h              # RAM usage

**🔁 Process Management**

CopyEdit

jobs                 # Show background jobs

fg                   # Bring job to foreground

bg                   # Send job to background

**🔧 Disk & USB Tools**

CopyEdit

mount                # Mount a device

umount               # Unmount a device

lsblk                # List block devices

fdisk -l             # Show disk partitions

df -h                # Disk free space

**🔄 System Management**

CopyEdit

reboot               # Reboot system

shutdown now         # Shutdown immediately

shutdown -h +10      # Shutdown in 10 minutes

sudo systemctl restart [service]  # Restart a service

sudo systemctl status [service]   # Service status

**📦 Snap & Flatpak (Alternative Package Managers)**

CopyEdit

sudo snap install [package]         # Install via snap

sudo snap remove [package]

flatpak install flathub [package]   # Install via flatpak

flatpak run [package]

**🧪 Development Tools**

CopyEdit

gcc file.c -o file    # Compile C program

python3 script.py     # Run Python script

node file.js          # Run Node.js script

git clone [repo]      # Clone git repo

**📜 Other Useful Commands**

CopyEdit

history               # Show command history

alias ll='ls -la'     # Create alias

clear                 # Clear terminal screen

man [command]         # Manual for command

Would you like a downloadab

**Bash**, **KSH**, and **CSH** are all **types of Unix/Linux shells** — basically command-line interpreters that allow users to interact with the operating system by typing commands.

Virtualisation allows one physical machine (like a server or computer) to run **multiple virtual machines (VMs)**, each behaving like an independent system with its own operating system and applications.

Shell: Is a Command-Line Interpreter that connects a user to Operating System and allows to execute the commands.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| date | Check today date | | |  | |
| cat /etc/shells | Let us check how types of shells are available in our syste | | |  | |
| Ps $$ | Let us check our shell process ID & kill the shell process from another session           ps: Stands for **process status** – it's used to view information about active processes.           $$: This is a special variable in **bash** and many other shells. It holds the **process ID (PID)** of the currently running shell. | | | PID TTY          TIME CMD  1234 pts/0    00:00:00 bash    PID: 1226  TTY: tty1  STAT: Ss  Time: 0:00  Command: -bash | |
| echo $SHELL | Let us check what shell we have? | | | /bin/bash | |
| whereis os-release | Os-release:   /etc/os-release | | |  | |
| uptime -help | get information on “uptime: | | |  | |
| man date | s get more information on “date | | |  | |
| pwd | Where do we stand in the system as a user | | |  | |
| Cal | Calendar | | |  | |
| Ssh [root@10.0.0.4](mailto:root@10.0.0.4) |  | | |  | |
| Sudo poweroff | Power off the linux machine | | |  | |
| exit |  | | |  | |
| uname –a | All System Information: | | | Linux myserver 5.14.0-503.14.1.el9\_5.x86\_64 #1 SMP PREEMPT\_DYNAMIC Fri Nov 15 12:04:32 UTC 2024 x86\_64 x86\_64 x86\_64 GNU/Linux | |
| uname -r | KERNEL RELEASE VERSION | | | 5.14.0-503.14.1.el9\_5.x86\_64 | |
| lscpu | **display detailed information about the CPU architecture** of your system.     To check if your CPU supports 64-bit.    To see how many physical/logical CPUs are available.    To verify virtualization support.    To troubleshoot performance or compatibility issues. | | | Architecture:            x86\_64  CPU op-mode(s):          32-bit, 64-bit  Byte Order:              Little Endian  CPU(s):                  8  On-line CPU(s) list:     0-7  Thread(s) per core:      2  Core(s) per socket:      4  Socket(s):               1  Vendor ID:               GenuineIntel  Model name:              Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz  CPU MHz:                 1992.000  L1d cache:               32K  L1i cache:               32K  L2 cache:                256K  L3 cache:                8192K | |
| cat /proc/cpuinfo | **detailed information about each processor core** on your system. It reads data from the **/proc/cpuinfo** virtual file, which contains info provided by the Linux kernel. | | | processor   : 0  vendor\_id   : GenuineIntel  cpu family  : 6  model       : 158  model name  : Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz  stepping    : 10  cpu MHz     : 1992.000  cache size  : 8192 KB  physical id : 0  siblings    : 8  core id     : 0  cpu cores   : 4  apicid      : 0  initial apicid  : 0  fpu         : yes  flags       : fpu vme de pse tsc ... | |
| pwd |  | | |  | |
| Ls |  | | |  | |
| cd |  | | |  | |
| cat |  | | |  | |
| su - username | **Switch user** | | |  | |
| Echo $PATH |  | | |  | |
| Echo $SHELL |  | | |  | |
| echo $HOST |  | | |  | |
| echo $USER |  | | |  | |
|  |  | | |  | |
| Commands to Set Up Apache on Rocky Linux  Once you're logged in to your EC2 instance using PuTTY: | | | | | |
| sudo dnf update -y | | | Update the system | |  |
| sudo dnf install httpd -y | | | **Install Apache (httpd)** | |  |
| sudo systemctl start httpd | | | Start Apache | |  |
| sudo systemctl enable httpd | | | Enable Apache to start on boot | |  |
| sudo firewall-cmd --permanent --add-service=http  sudo firewall-cmd –reload | | | Allow HTTP through the firewall (if firewalld is running)  ***Note:****This step may not be needed if your AWS Security Group already allows port 80 (HTTP). But it’s safe to run anyway.* | |  |
| http://<your-ec2-public-ip> | | | Test Apache  pen your browser and go to | |  |
|  | |  | |  | |
|  | |  | |  | |
|  | |  | |  | |
|  |  |  |  |  |  |

Ps $$:

|  |  |
| --- | --- |
|          **PID** |          **Process ID** – a unique number assigned to each running process by the operating system. In this example, 1234 is the ID of the bash shell. |

|  |  |
| --- | --- |
|          **TTY** |          **Terminal Type** – the terminal associated with the process. pts/0 refers to a "pseudo-terminal" (like when you're using a terminal emulator or SSH session). |

|  |  |
| --- | --- |
|          **TIME** |          **CPU Time** – the total amount of CPU time the process has used since it started. 00:00:00 means it hasn't used any noticeable CPU time yet. |

|  |  |
| --- | --- |
|          **CMD** |          **Command** – the command that started the process. In this case, it's bash, which is your shell. |

myserver Hostname of machine

5.14.0-503.14.1.el9\_5.x86\_64 Kernel version #1 Kernel built version SMP Symmetric multiprocessing support PREEMPT\_DYNAMIC Preemption model used by the kernel Fri Nov 15 12:04:32 UTC 2024 Build time and date of the kernel X86\_64 (3 times) Machine hardware name, Processor type, Hardware platform GNU/Linux Operating System

|  |
| --- |
| #ls   :   #ls –a    :    #ls –al   :    #al –al /abc.txt  #ls –l a.txt   :   ll a.txt       same results shows  #ls -il                                      files and directories with permission                                                  permissions, ownership, timestamps, and disk block locations  #ls /dev sda\*     list of files starting from sda  pwd    print working directory  cat a.txt       view file  #cat /dev os-release  #ls /etc/os-release          to view linux OS detail  #uptime                               How long the System has been running with date, time user info.  #uptime –p                         up 43 minutes  #runlevel            what services and processes are running.  #top                      **to**monitor system processes in real time — like Task Manager for the terminal.  ps -e                   command lists **all running processes** on the system.                                  ps = process status                                  -e = show every process (same as –a)  hostname –I       just print IP Address i.e. 192.168.1.100  hostnamectl       to check the host name  hostname           to check the host name  sudo hostnamectl set-hostname webserver1       to change hostname permanently -  not change  hostnamectl set-hostname webserver1                 to change hostname  hostname newhostname                                             to change hostname  uname –r                             prints **only the kernel version** of your Linux system.  uname –a                            **all detail system information** related to the kernel and machine.  uname –m                          shows the **machine hardware name**  lsb\_release –a                   command not found  kernelctl stuatus               command not found  cpuinfo                                 command not found  cat /proc/cpu  lscpu                                      **summary of the CPU architecture and hardware info.**, including number of cores, threads,                                                  model, and more.  slblk       information about **block devices** such as hard drives, SSDs, USB drives, and partitions.                  lsblk shows the tree **from disks down to partitions**  Slblk –s reverses that — it shows a **device (like a partition or LVM)** and **all the block devices it**  **depends on** (its parents).  #ls –l /dev/input              shows input devices (keyboard, mouse, touchpads) as **character devices**    free                                       shows **memory usage** (RAM and swap) on your system.  free –m                                shows output in Megabytes  free –g                                  shows output in Gigabytes    #du        Show size of current directory and its subdirectories:  **"disk usage"** /size and is used to check how much space files and directories are using.    #du –h                  Show sizes in **human-readable** format:  #du –sh                Show **only the total size** of a directory:  #du –h /var         Show**the total size** of a var folder  #du -h abc.txt    Show**the total size** of a abc.txt file.  #du -h --max-depth=1 /var           Show size of all subdirectories, human-readable:    #df         to **display the amount of disk space** available and used on **mounted filesystems**.  #df –h   **human-readable** (displays sizes in KB, MB, GB)  df –hT   Show all file systems with type    #ifconfig               ifconfig: command not found                                  ifconfig is part of the old net-tools package, which is **deprecated** on most                                  modern Linux distributions (Ubuntu, Debian, CentOS, etc.) and is **not installed by**  **default anymore**.              ifconfig is part of the legacy net-tools package.  #sudo dnf install net-tools  #ifconfig    #mkdir test  #rmdir test  #echo ‘This is line 1’ > abc.txt  #echo ‘add lines >> abc.txt  #cat abc.txt  #rm abc.txt  rm file1.txt file2.log file3.jpg  #ls –l a\*      (list of file start with a)  #chmod ugo+rwx abc.txt      (add permission to User,group,other to read,write,execute)  #chmod ugo-rwx abc.txt      (remove permission to User,group,other to read,write,execute)      #touch abc.txt (create new file)  #vim abc.txt       command not found  # sudo dnf install vim  #vim a.txt  #i (press i) for edit mode  / type your text / Press ESC to return mode  :w (save),    :q (quit), :wq (save and quit), :ql (quit without saving)    #cd                                         change directory  #cp abc.txt /test              copy file into test dir  #cp –r abc.txt /test         copy file into test dir  #cp a\* /test                       copy files start with ‘a’ into test dir  #rm -rf \*                              remove all files  #rm -rf /home/noman   remove dir          httpd     **HTTP Daemon** — it's a **web server software** (computer act like a web server)                  that accept **HTTP requests** (website URL) in browsers) and serves **web pages, files, or data** in                  response.    # dnf install httpd –y                       (it install httpd package – for root user)  #sudo dnf install httpd                   (it install httpd package – for others users)  #do dnf install httpd –y                 (it install httpd package – for others users without asking)  #sudo systemctl start httpd         (Start the httpd service)  #sudo systemctl enable httpd     (Enable **httpd** to start on boot)  #sudo systemctl status httpd      (Check status (optional))  #sudo firewall-cmd --permanent               (add-service=http)  #sudo firewall-cmd --permanent               (add-service=https)  #sudo firewall-cmd                                         (reload) |
| #ls          **list the contents of a directory**.   | **Command** | **Description** | | --- | --- | | ls -l | Long listing format (shows permissions, owner, size, etc.) | | ls -a | Shows **all** files, including hidden ones (those starting with .) | | ls -lh | Human-readable sizes (K, M, G) | | ls -R | Recursively lists subdirectories | | ls -lt | Sorts by modification time, newest first | |
| **#runlevel**   defines what services and processes are running.                      In short determines the system's mode of operation, like single-user mode, multi-user                     mode, or graphical mode.   | **Runlevel** | **Mode** | | --- | --- | | 0 | Halt / Shutdown | | 1 | Single-user mode (Rescue) | | 3 | Multi-user mode (Text-based) | | 5 | Multi-user mode (Graphical) | | 6 | Reboot | |
| #uname –r          prints **only the kernel version** of your Linux system.  5.14.0-327.el9.x86\_64  5.14.0 → Kernel version number  -327.el9 → Distribution-specific build or patch version (here, for Rocky Linux 9)  x86\_64 → Kernel architecture (64-bit) |
| #uname –a         **all system information** related to the CUP (machine) and kernel.  Linux rocky-linux 5.14.0-327.el9.x86\_64 #1 SMP Thu Jul 27 13:15:32 UTC 2025 x86\_64 x86\_64 x86\_64 GNU/Linux   | **Part** | **Meaning** | | --- | --- | | Linux | Kernel name | | rocky-linux | Hostname | | 5.14.0-327.el9.x86\_64 | Kernel version | | #1 SMP Thu Jul 27 13:15:32 UTC 2025 | Kernel build info (build number, SMP support, build date) | | x86\_64 (three times) | Machine hardware, processor, platform | | GNU/Linux | Operating system | |
| #uname –m  x86\_64   | **Output** | **Architecture** | **Description** | | --- | --- | --- | | x86\_64 | 64-bit | Most common for modern systems | | i686 | 32-bit | Older 32-bit Intel/AMD CPUs | | aarch64 | 64-bit ARM | Used on newer ARM devices, servers | | armv7l | 32-bit ARM | Used on Raspberry Pi and similar | |  |  |  | |  |  |  | |
| #lsblk    information about **block devices** such as hard drives, SSDs, USB drives, and partitions.  NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS  sda      8:0    0   100G  0 disk  ├─sda1   8:1    0    50G  0 part /  ├─sda2   8:2    0    45G  0 part /home  └─sda3   8:3    0     5G  0 part [SWAP]  sr0     11:0    1  1024M  0 rom    #lsblk –s               reverses that — it shows a **device (like a partition or LVM)** and **all the block devices it**  **depends on** (its parents).  sda  ├─sda1  ├─sda2  └─sda3 |
| #df –h   **human-readable** (displays sizes in KB, MB, GB)    Filesystem      Size  Used Avail Use% Mounted on  /dev/sda1        50G   25G   23G  53% /  tmpfs           1.9G  2.0M  1.9G   1% /run  /dev/sdb1       100G   60G   37G  62% /mnt/data     | **Column** | **Meaning** | | --- | --- | | **Filesystem** | The device or virtual FS (e.g., /dev/sda1) | | **Size** | Total size of the filesystem | | **Used** | Space used | | **Avail** | Space available | | **Use%** | Usage percentage | | **Mounted on** | Where it is mounted in the directory tree | |
|  |
| When to use sudo:   * sudo is needed when you run commands that require administrative (root) privileges. Installing software (dnf install), modifying system files, starting/stopping system services, changing firewall settings — these all usually need sudo. * Regular users (non-root) cannot perform these actions without sudo.   Do *all* users use sudo?   * Only users who have been granted permission to use sudo (usually members of the wheel or sudo group) can run commands with sudo. * If a user is *not* in those groups, trying sudo will either:   + Ask for a password and then deny access, or   + Simply deny access outright. |
|  |
|  |

#ls –l      permission, ownership, time, files/dir names

#ls –il     total size, permission, ownership, time, files/dir names

|  |  |
| --- | --- |
| Static IP address | **that does not change** over time. It is manually assigned to a device (like a server, PC, router, or printer) and remains constant |
| dynamic IP address | which is automatically assigned by a DHCP server and can change over time. |
|  |  |

**HOW DO YOU RECOVER ROOT PASSWORD**

1. Start (Power on) the System.

2. Wait for GRUB menu - press e to edit.

3. Find the line starting with linux and type the rd.break at the end.

4. Press Ctrl+x

5. mount –o remount,rw /sysroot

6. chroot /sysroot                                        (change root to sysroot)

7. touch /.autorelabel

8. exit

9. exit

**DELETING USER**

The userdel command can’t delete a user if they are **currently logged in** or **have processes running**. You must first stop those processes.

#adduser noman

#userdel noman               Delete the User

#Ps –u noman                    to see **all processes by the user**:

sudo kill -9 4472                Kill Specific Process (e.g. 4472)

sudo pkill -u noman        Kill All User Processes

sudo userdel noman       Delete the User

sudo userdel -r noman   With deleting home directory

sudo rm -rf /home/abc  **forcefully and recursively** remove everything under /home/abc.

getent passwd noman   verify user has deleted

sudo userdel abc              Deletes the user account, but **leaves the home directory**

sudo userdel -r abc          Deletes the user **and** their home directory.

sudo pkill -u abc                Kills all running processes of user abc, but doesn’t remove files.

Note: If just ran userdel without -r, their files remain in /home.

There are 5 Type of Users in Linux:

1. System User (This is Root User and the Admin)

2. Normal User (This is a user created by Root User)

3. Network User (this is a remote user that is logs-in Windows Active Directory or LDAP in the case of Linux Users}

4. System User based on an installed application like Oracle etc.

5. Sudo User (normal users who have admin or root privileges.

#w                   (show currently logged in user)

#who               (show currently logged in user with detail i.e. user   tty   login date/time)

#whoami         (show currently logged in user, only user name)

# cut -d: -f1 /etc/passwd          (all users that exist, only name)

#getent passwd                       (all users that exist, with detail)

Note:   the best way to switch user Ctrl+Alt+F2’

                #su -      (root)

                #su – noman      (other user)

|  |
| --- |
| Prob: I login to root user then I logged in as 'su - noman' . when execute 'w' then only root user show.  Even though you're now running commands as noman (via su - noman), you're **still using the same terminal session that belongs to root**.   * The w command shows users **logged in through a TTY or PTY** (e.g., SSH sessions, console logins, terminal windows). * su - noman switches the user identity **within the same session** — it doesn't create a new login session.   So w continues to show **root**, because the controlling terminal still belongs to root.  Even though you're now running commands as noman (via su - noman), you're **still using the same terminal session that belongs to root**.   * The w command shows users **logged in through a TTY or PTY** (e.g., SSH sessions, console logins, terminal windows). * su - noman switches the user identity **within the same session** — it doesn't create a new login session.   So w continues to show **root**, because the controlling terminal still belongs to root.  Solution:  How to Actually Log In as noman (and see in w)  To see noman listed as a logged-in user in w, you need to **log in as noman from a new terminal or SSH session**, not via su.  ✔ Option 1: Switch TTY  1.       Press **Ctrl+Alt+F2** (or F3, F4...) to switch to another virtual terminal.  2.       Log in as noman directly. |

To close user noman (it will not display through #w command)

#pkill -u noman

#exit

$w

$whoami

$lastlog

#lastb               (bad logins)

#last                 (List of Users Logged-in and Logged out / shutdown)

A **user group** is a collection of users in Linux that share **the same permissions** for accessing files, directories, or resources.

Why Groups Matter:

         Makes **permission management easier**.

         Lets you assign access to multiple users **at once**.

$groups                       Check Groups for the Current User

$groups noman           Check Groups for a Specific User

$id noman                          Detailed Group Info (Using id)

$cat /etc/group            List All Groups on the System

$getent group              List All Groups on the System

Solution: Add noman to the wheel Group

$su –                                                     log in as root

#usermod -aG wheel noman       Add noman to the wheel group - This **grants sudo access** to the user.

#su – noman

$groups noman                                Have noman Log Out and Back In and confirm wheel group access

$ whoami                                            it show ‘noman’

$sudo whoami                                  it show ‘root’                     That confirms noman can now use sudo.

$sudo grep wheel /etc/sudoers

sudo groupadd permitgroup                       Create the New Group

sudo usermod –aG permitgroup noman add user ‘noman’ to group

groups noman                                                   Make sure permitgroup is listed.

getent group permitgroup                           verify it created or not

sudo groupdel permitgroup                         delete group

**Create user when logging the msg display: Authentication failure**

getent passwd noman                   (Make sure the user was created successfully)

sudo useradd -m noman               (If not, recreate the user)

sudo passwd noman                                       (Set the password)  or

echo "noman:abc" | sudo chpasswd        (Set the password)

sudo passwd -S noman                                  (Check Account Status)

sudo passwd -u noman                                  (If status is L (locked), unlock it)

udo journalctl –xe                                           (Check logs for exact error messages)

## Evaluation 2 Practical Guide (Step-by-Step)

### ✅ 1. ****Add New Disk to VM (VirtualBox)****

**From VirtualBox:**

1. Shut down the Rocky Linux VM
2. Go to **Settings** → **Storage**
3. Click on the **Controller: SATA** → Click the **+ Disk** icon
4. Choose **Create new disk** → Select size (e.g., 2 GB) → Finish
5. Start the VM

Now, the new disk (e.g., /dev/sdb) will be available inside the VM.

### ✅ 2. ****Create a Partition on the New Disk****

Inside your VM:

lsblk # Confirm the new disk, e.g., /dev/sdb

sudo fdisk /dev/sdb

**Inside fdisk:**

* Press n → new partition
* Accept defaults
* Press w → write and exit

Verify:

lsblk

Now you should see /dev/sdb1.

### ✅ 3. ****Create Physical Volume and LVM****

sudo pvcreate /dev/sdb1 # Create physical volume

sudo vgcreate myvg /dev/sdb1 # Create volume group

sudo lvcreate -L 1G -n mylv myvg # Create logical volume (1GB)

sudo mkfs.xfs /dev/myvg/mylv # Format it

Create a mount point and mount it:

sudo mkdir /mnt/mydata

sudo mount /dev/myvg/mylv /mnt/mydata

Optional: add to /etc/fstab for permanent mount.

### ✅ 4. ****Set Up Swap Space****

You can use a file or a partition. Here's how to do it from the same disk (if space allows):

sudo lvcreate -L 512M -n swaplv myvg

sudo mkswap /dev/myvg/swaplv

sudo swapon /dev/myvg/swaplv

Make it permanent:

echo '/dev/myvg/swaplv none swap sw 0 0' | sudo tee -a /etc/fstab

Check:

swapon --show

free -h

### ✅ 5. ****(Optional) Set Up RAID (mdadm)****

Only if **RAID** is required and you have **at least 2 new disks**.

Example for RAID 1 (mirror):

sudo mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc

watch cat /proc/mdstat # to monitor sync

Create filesystem and mount:

sudo mkfs.ext4 /dev/md0

sudo mount /dev/md0 /mnt/raid1

## ✅ Final Checklist

| **Task** | **Status** |
| --- | --- |
| Add new virtual disk in VirtualBox | ✅ |
| Partition the new disk | ✅ |
| Create LVM (PV → VG → LV) | ✅ |
| Format and mount LV | ✅ |
| Create and enable swap | ✅ |
| Setup RAID (optional) | ✅ (if needed) |

Let me know which part you'd like to try first — I can walk you through that one in **real time**.

**BASH** stands for **Bourne Again SHell**. It is:

### 🔹 A Command-Line Shell:

* **Shell** = A program that takes commands from the user and gives them to the operating system to execute.
* BASH is one of the most widely used **Unix/Linux shells**.

How is BASH different from the DOS command prompt

* Case sensitivity
* Dos (slash \), Linux (backslash /)