TRAINING DAY 11 REPORT:

- ## Linux Networking Commands

  Some important **Linux commands for networking**, which help check IP addresses, connections, and network devices.

1. IP & Network Info

Command	Description
ip a	Shows IP address & network interfaces
ifconfig	(Old) IP address & interface info
ip r	View routing table (like `route`)
hostname -I	Show only the system's IP

2. Network Connectivity

Command	Description
ping	Test if a host is reachable
traceroute	Shows the path packets take
curl or wget	Download web pages or files
netstat -tuln	Show listening ports
ss -tuln	Faster replacement for netstat

3. DNS and Name Resolution

Command	Description
nslookup	Query DNS info for a domain
dig	Detailed DNS query tool
host	Simple domain lookup

4. Devices and Interfaces

Command	Description
ip link	List all network interfaces
ethtool eth0	Show details of an Ethernet device
iwconfig	Wireless network config info
nmcli	Manage network connections (GUI alternative)

5. ARP, MAC & Packet Tools

Command	Description
arp -a	View ARP table (IP ↔ MAC mapping)
tcpdump	Capture live network packets
nmap	Scan network for devices/ports

- ***`grep` Command in Linux***

`grep` command, which is used to **search text** in files or output. It's super useful when I want to **find specific words, lines, or patterns** in huge files.

What is `grep`?

`grep` stands for:
Global **R**egular **E**xpression **P**rint

It is used to **search for specific words, phrases, or patterns** in:

- Files

- Command output

- Logs

- **Basic Syntax:**

  ```
  grep [options] pattern filename
  ```

Examples :

Command	Meaning
`grep error log.txt`	Find lines with "error" in log.txt
`grep -i error log.txt`	Case-insensitive search for "error"
`grep -r "admin" /etc`	Recursively search "admin" in all files in /etc
`grep -n "root" /etc/passwd`	Show line numbers with matches
`grep -v "test" file.txt`	Show all lines **not** containing "test"
`ps aux	grep firefox`	

- ## Working with Linux User Administration

  How Linux handles **users, groups, and file permissions**, and how I can **create, modify, or delete users** using simple terminal commands.

  ### Basic Linux User Types:

  1. **Root user (`UID=0`)** → Full control (superuser)
  2. **Regular users** → Created by admin or during install
  3. **System users** → For running services (e.g., `mysql`, `www-data`)

1. Managing Users

Task	Command Example
Add a new user	`sudo adduser username`

Task	Command Example
Delete a user	`sudo deluser username`
Add user to a group	`sudo usermod -aG groupname username`
View all users	`cat /etc/passwd`
View all groups	`cat /etc/group`

After adding a user, you can **set password** with:

```
sudo passwd username
```

2. File Ownership:

Every file/folder has 3 owners:

- **User (u)** → file creator
- **Group (g)** → assigned group
- **Others (o)** → everyone else

Check with:

```
ls -l filename
```

3. File Permissions

Symbol	Meaning
r	Read
w	Write
x	Execute
-	No access

Example:

```
-rwxr-xr-- 1 user group file.sh
```

Meaning:

- **User:** read, write, execute

- **Group:** read, execute
- **Others:** read

4. Change Ownership / Permissions

Task	Command Example
Change owner	sudo chown user file
Change group	sudo chgrp group file
Change permissions (symbolic)	chmod u+x file.sh
Change permissions (numeric)	chmod 755 file.sh

Symbolic Example:

- `u+x` → add execute to user
- `g-w` → remove write from group

Numeric Example:

- `chmod 755 file.sh` → rwx (user), rx (group), rx (others)

User Home Directories:

- All user data is in `/home/username`

- Root user has `/root` directory