**📆 Week 1 – Linux Basics (Detailed Explanation)**

**🔹 Day 1 – Linux Introduction & Navigation**

**🔸 Why this matters in DevOps?**

* All servers (AWS EC2, Kubernetes nodes, Docker containers) mostly run **Linux**.
* You must know how to **navigate**, **check files**, and **move inside directories** to work on systems.

**🔸 File System Hierarchy**

Think of Linux like a **tree**:

* / → root (starting point).
* /home → each user’s personal space.
* /etc → system configuration files.
* /var/log → log files (super important for debugging).
* /bin → essential commands.

**🔸 Commands**

pwd # shows your current location in the system

ls # lists files in the current directory

ls -l # shows files with details (permissions, owner, size)

ls -a # shows hidden files (files starting with .)

cd / # go to root directory

cd ~ # go to home directory

cd /var/log # move to logs directory

👉 **Practice:** Move to /var/log, list files, then go back to home (cd ~).

**🔹 Day 2 – File & Directory Management**

**🔸 Why this matters in DevOps?**

* You’ll constantly create/edit config files (nginx.conf, docker-compose.yml, kubeconfig).
* You must be able to move/copy files between directories.

**🔸 Commands**

touch file1.txt # create a new empty file

echo "Hello DevOps" > file1.txt # add text to a file

cat file1.txt # view content of a file

mkdir devops # make new directory

mv file1.txt devops/ # move file into folder

cp devops/file1.txt devops/file2.txt # copy file

rm devops/file2.txt # delete file

👉 **Practice:**

* Create a folder week1/
* Inside it, create notes.txt with "Linux is fun".
* Copy it to backup.txt, then delete notes.txt.

**🔹 Day 3 – Viewing & Editing Files**

**🔸 Why this matters in DevOps?**

* Logs, configuration files, and YAML manifests are often **large**.
* You need ways to quickly **view or edit them** without crashing your terminal.

**🔸 Commands**

cat file.txt # show file content

less file.txt # open file, scroll with arrows, press q to quit

head -n 10 file.txt # show first 10 lines

tail -n 10 file.txt # show last 10 lines

tail -f logfile.log # continuously watch a log file

👉 **Practice:** Open /var/log/messages, scroll, and find an “error” or “warning”.

**🔹 Day 4 – Users & Permissions**

**🔸 Why this matters in DevOps?**

* DevOps engineers must **secure files**.
* For example, database passwords must be **hidden from normal users**.

**🔸 Permissions explained**

* Format: rwxr-xr--
  + r = read, w = write, x = execute.
  + First 3 = owner, next 3 = group, last 3 = others.

**🔸 Commands**

whoami # show current user

id # show user ID and groups

ls -l # check file permissions

chmod 600 file # only owner can read/write

chmod +x script.sh # add execute permission

chown root file # change owner of file (requires sudo)

👉 **Practice:** Create secure.txt, give yourself read/write only (chmod 600 secure.txt).

**🔹 Day 5 – Processes & Monitoring**

**🔸 Why this matters in DevOps?**

* You need to monitor **running applications** (like Nginx, MySQL, Docker).
* When servers crash, you must **kill or restart** stuck processes.

**🔸 Commands**

ps aux # show all running processes

top # live process monitor (press q to quit)

kill -9 PID # force kill a process

free -m # check memory usage

df -h # check disk usage

uptime # how long system has been running

👉 **Practice:** Run top, find the process with highest CPU usage, and note its PID.

**🔹 Day 6 – Package Management & Services**

**🔸 Why this matters in DevOps?**

* You often need to **install software** (Nginx, Docker, Kubernetes tools).
* Services (like sshd, nginx) must be **started, stopped, restarted**.

**🔸 Commands**

yum list installed # list installed packages

yum install nano -y # install nano editor

systemctl status sshd # check status of SSH service

systemctl stop sshd # stop service

systemctl start sshd # start service

systemctl enable sshd # enable service on boot

👉 **Practice:** Check if sshd is running, then restart it.

**🔹 Day 7 – Review & Mini Project**

**🔸 Mini Project**

1. Create a folder project/.
2. Inside, create app.log.
3. Add dummy logs:
4. echo "ERROR: Something failed" >> app.log
5. echo "INFO: Service started" >> app.log
6. Change file to read-only (chmod 400 app.log).
7. Watch it live with:
8. tail -f app.log

This simulates how DevOps engineers monitor logs of real applications.

**✅ By the end of Week 1, you will:**

* Comfortably **navigate Linux**.
* **Create, edit, move, delete files**.
* Understand **permissions & ownership**.
* Monitor **processes & logs**.
* Manage **services & packages**.