🧠 Topic 1: Introduction to Linux OS & File System

What is Linux?

Linux is an open-source, Unix-like operating system used extensively in servers, cloud computing, DevOps pipelines, and containers. It’s preferred in DevOps for its stability, scripting power, and community support.

Key Concepts

- **Distributions (Distros):** Ubuntu, CentOS, RHEL, Debian, Alpine

Linux Boot Process:

```
BIOS → GRUB → Kernel → init/systemd → Shell → User
```

-

Linux File System Structure

- `/` – Root directory (everything starts here)
- `/bin` – Essential binary executables (e.g., `ls`, `cp`)
- `/etc` – Configuration files
- `/home` – User directories
- `/var` – Logs, mail, spool
- `/dev` – Device files
- `/proc` – Kernel and process info (virtual filesystem)

File Permissions

- View with: `ls -l`

Modify with:

```
chmod 755 file.sh
chown user:group file.sh
```

-

File Links

- **Hard Link:** `ln file1 file2`
- **Soft Link (Symlink):** `ln -s file1 file2`

🧰 Topic 2: Essential Linux Commands for DevOps

Navigation & File Handling

```
cd /var/log     # Change directory
pwd             # Print working directory
ls -la          # List files with permissions
touch file.txt  # Create a file
cat file.txt    # Display contents
cp file1 file2  # Copy
mv file1 file2  # Move
rm file.txt     # Delete
```

Process & Memory Management

```
ps aux          # View processes
```

```
top              # Real-time system monitor
free -m          # Memory usage
df -h            # Disk usage
kill -9 <PID>    # Kill process
```

Networking Tools

```
ping google.com
curl https://example.com
wget http://file.com/file.txt
netstat -tuln
```

Package Management

- **Ubuntu/Debian:** `apt`, `dpkg`
- **RHEL/CentOS:** `yum`, `dnf`, `rpm`

🐚 Topic 3: Shell Environment and Scripting Basics

What is a Shell?

Shell is the command interpreter; (Bourne Again Shell) is the most used shell in Linux systems.

Environment Variables

```
echo $USER
echo $PATH
```

Script Structure

```
#!/bin/
echo "Hello $USER"
```

Variables & User Input

```
name="DevOps"
echo $name

read -p "Enter your name: " username
echo "Hello $username"
```

Aliases

Define shortcuts in `~/.rc`:

```
alias ll='ls -la'
```

🔁 Topic 4: Conditional Statements and Loops

If-Else Condition

```
if [ "$USER" == "root" ]; then
  echo "Welcome root"
else
  echo "You are $USER"
fi
```

For Loop

```
for i in {1..5}
```

```
do
  echo "Number $i"
done
```

While Loop

```
count=1
while [ $count -le 5 ]
do
  echo "Count: $count"
  ((count++))
done
```

🔧 Topic 5: Functions, Arguments, and Exit Status

Function

```
greet() {
  echo "Hello $1"
}
greet DevOps
```

Script Arguments

```
echo "Script Name: $0"
echo "First Arg: $1"
echo "All Args: $@"
```

Exit Status

```
exit 0     # Success
exit 1     # Failure
echo $?    # Returns exit status of last command
```

Use Case: File Exists Check

```
if [ -f "$1" ]; then
  echo "File Exists"
else
  echo "File Not Found"
  exit 1
fi
```

📦 Topic 6: Working with Files, Archives, and Processes

Compression & Archiving

```
tar -cvf archive.tar /dir
gzip file.txt
zip files.zip *.txt
```

Searching

```
find /var/log -name "*.log"
grep "ERROR" /var/log/syslog
```

Background Jobs

```
sleep 60 &     # Run in background
jobs           # List background jobs
```

```
fg %1          # Bring to foreground
kill %1        # Kill background job
```

Cron Jobs

```
crontab -e
# Run script every day at midnight
0 0 * * * /path/to/script.sh
```

🚀 Topic 7: Real-Time DevOps Shell Script Use Cases

🧹 Log File Cleanup

```
find /var/log -name "*.log" -mtime +7 -exec gzip {} \;
```

💽 Disk Usage Alert

```
disk_usage=$(df / | awk 'NR==2 {print $5}' | sed 's/%//')
if [ "$disk_usage" -gt 80 ]; then
  echo "Disk usage alert: $disk_usage%"
fi
```

🔁 Nginx Service Health Check

```
if ! systemctl is-active --quiet nginx; then
  echo "Restarting Nginx..."
  systemctl start nginx
fi
```