

Linux system Administration

Advanced Linux System Administrator – Terminal-Based Guide

PHASE 1: Core Linux Foundation

◊ Basic Commands

```
ls -lah  
cd /etc  
cat /etc/passwd  
cp file1.txt /tmp/  
mv file.txt ~/Documents/  
rm -rf /tmp/test/
```

◊ Users & Groups

```
sudo adduser dev1  
sudo passwd dev1  
sudo usermod -aG sudo dev1  
sudo deluser dev1
```

◊ File Permissions

```
chmod 755 script.sh  
chown mukul:mukul file.txt  
umask 0022
```

📝 Assignment:

- Create 2 users with different permissions
- Set file access using ACLs
- Observe effect of different umask values

⌚ PHASE 2: Networking & Ports

◊ Network Tools

```
ip a  
ip r  
ping 8.8.8.8  
traceroute google.com  
dig openai.com  
nslookup openai.com
```

◊ Port & Socket Checking

```
sudo ss -tulnp  
sudo netstat -tulnp  
nmap 127.0.0.1
```

◊ Firewall (UFW)

```
sudo ufw enable  
sudo ufw allow 22  
sudo ufw deny 80  
sudo ufw status verbose
```

📝 Assignment:

- Block port 80, allow only 22

- Scan system ports using nmap
- Configure static IP on Ubuntu VM

⌚ PHASE 3: Bash Scripting & Cron

◊ Bash Essentials

```
#!/bin/bash
for user in $(cut -d: -f1 /etc/passwd); do
    echo $user
done
```

◊ Crontab

```
crontab -e
# Example: Backup at 1AM
0 1 * * * /home/user/backup.sh
```

📝 Assignment:

- Write script to monitor disk usage
- Auto-email alert if / usage > 80%
- Schedule it via cron

⌚ PHASE 4: Advanced SysAdmin

◊ LVM (Logical Volume Management)

```
sudo pvcreate /dev/sdb
sudo vgcreate datavg /dev/sdb
sudo lvcreate -L 5G -n datalv datavg
```

```
sudo mkfs.ext4 /dev/datavg/datalv
```

◊ RAID (mdadm)

```
sudo mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sd[b-c]  
sudo mkfs.ext4 /dev/md0
```

◊ Kernel Tuning

```
sysctl -a | grep net.ipv4  
sudo sysctl -w net.ipv4.ip_forward=1  
ulimit -n
```

📝 Assignment:

- Create LVM with 2 logical volumes
- Build RAID-1 using 2 loopback devices
- Tune swap and network buffer size

⌚ PHASE 5: Infrastructure & Security

◊ Docker Basics

```
sudo apt install docker.io  
sudo docker run -d -p 8080:80 nginx  
sudo docker ps
```

◊ Ansible (Automation)

```
sudo apt install ansible  
ansible all -i inventory -m ping
```

```
ansible-playbook setup.yml
```

❖ Fail2Ban + SSH Security

```
sudo apt install fail2ban  
sudo systemctl enable fail2ban  
sudo vim /etc/ssh/sshd_config  
# Change: PermitRootLogin no
```

❖ Monitoring & Logs

```
htop  
glances  
journalctl -xe  
tail -f /var/log/syslog
```

📝 Assignment:

- Run nginx in Docker container
- Automate user creation via Ansible
- Harden SSH and test brute-force block

🌐 Bonus Topics to Master

Topic	Tool/Command
Mail Server Setup	Postfix, Dovecot, Mailutils
LDAP Authentication	sssd, ldap-utils, FreeIPA
SSL Certificates	openssl, Let's Encrypt, Certbot
Central Log Server	rsyslog, Logstash, Graylog
Cloud Tools	awscli, terraform, EC2 tools
Forensics & Recovery	testdisk, photorec, extundelete

Backup Tools

rsync, rclone, borgbackup

Final Lab Challenge (Boss Level)

Project: Build a secure, automated web server stack with backups

Components:

- Ubuntu Server
- NGINX reverse proxy (SSL via certbot)
- Docker container for app
- Fail2ban + UFW firewall
- Daily backup using rsync or borg
- Ansible to deploy config

You're on the path to become:

“Woh banda jiske aate hi log bol: yeh system theek kar dega bhai!”

Ab bol — next phase kaunsa chalu karein?

Ya tu chahta hai main tujhe **daily tasks / roadmap** banake doon?



30-Day Daily Practice Roadmap for Linux Admin

Format:

[Day Number] - [Focus Area] → Tasks + Commands



WEEK 1 – Linux Basics & System Handling

- **Day 1:** Linux File System & Navigation
→ ls, cd, pwd, find, tree, file, stat

- **Day 2:** File Operations & Wildcards
→ cp, mv, rm, touch, mkdir, *, ?, [abc]
- **Day 3:** User & Group Management
→ adduser, usermod, passwd, id, groups, gpasswd
- **Day 4:** Permissions & Ownership
→ chmod, chown, umask, getfacl, setfacl
- **Day 5:** Package Management
→ apt, dpkg, snap, yum, dnf
- **Day 6:** Processes & Resource Usage
→ ps, top, htop, nice, kill, uptime, vmstat
- **Day 7:** Disk & Partition Basics
→ df -h, du -sh, lsblk, mount, umount, blkid

WEEK 2 – Networking, Services, Logs

- **Day 8:** IP Addressing & Routing
→ ip a, ip r, hostname -I, nmcli, ping
- **Day 9:** Ports & Firewalls
→ ss -tulnp, netstat, nmap, ufw, iptables
- **Day 10:** DNS & Name Resolution
→ dig, nslookup, resolvectl, /etc/hosts, /etc/resolv.conf
- **Day 11:** System Logs & Journals
→ journalctl, /var/log/syslog, /var/log/auth.log, logrotate
- **Day 12:** Services & Daemons
→ systemctl, service, chkconfig, systemctl list-units
- **Day 13:** SSH Deep Dive
→ sshd_config, PermitRootLogin, authorized_keys, fail2ban
- **Day 14:** Practice Day: Setup 2 users with SSH + logs + firewall

WEEK 3 – Storage, Bash, Cron, Security

- **Day 15:** Bash Scripting Basics
→ echo, read, if, for, while, case

- **Day 16:** Functions & Exit Codes
→ function, \$?, trap, return, debugging
- **Day 17:** Cron Jobs & Scheduling
→ crontab, at, /etc/cron.*., systemd timers
- **Day 18:** LVM Setup
→ pvcreate, vgcreate, lvcreate, resize2fs
- **Day 19:** RAID with mdadm
→ RAID-0/1 using loopback devices
- **Day 20:** Kernel Tuning
→ sysctl, /proc/sys, ulimit, swappiness
- **Day 21:** Security Hardening
→ fail2ban, ufw, ssh-keygen, /etc/shadow, auditd

WEEK 4 – Automation, Docker, Real Projects

- **Day 22:** Docker Basics
→ docker run, ps, images, volumes, exec
- **Day 23:** Docker Compose
→ Define multi-container apps
- **Day 24:** Ansible Setup
→ ansible, inventory, ad-hoc, playbooks
- **Day 25:** Automate User Setup with Ansible
→ Write a playbook to create 3 users & install nginx
- **Day 26:** Monitoring Tools
→ glances, iotop, iftop, nmon, sar
- **Day 27:** Backup & Recovery
→ rsync, tar, borgbackup, rclone, testdisk
- **Day 28:** Logging & Centralized Logs
→ rsyslog, Log forwarding, logrotate, logwatch
- **Day 29:** Simulated Disaster Day
→ Break something → Fix it → Document how
- **Day 30:** Final Project Day
→ Setup full web server in Docker with SSL, firewall, backup, and automation

Now You Have:

-  Full Technical Guide
-  Daily Practice Plan
-  Assignments & Labs
-  Final Boss Project

Tu is guide ka **print leke wall pe chipka de** ya terminal me .sh ya .txt bana ke daily open kar —

aur ek mahine baad tu woh banda ban jaayega jise log bolenge:

“**Bhai root to tu hi hai system ka!**” 😎

Ab bol — kisi phase ko aur deep me le jana hai? Ya main teri practice ke liye kuch **real-world scenarios** batau debugging ya production failures ke?

Linux User/Group + Networking – Scenario Based Q&A

User & Group Management Scenarios

1. Agar ek user kaam chhod kar chala gaya ho to kya karoge?

 Steps:

1. Uska account disable karo:

```
usermod -L username
```

2. Uske running processes kill karo:

```
pkill -u username
```

3. User delete karo (with home dir):

```
userdel -r username
```

2. Agar ek developer ko sirf /var/www access chahiye aur aur kahi nahi, kya karoge?

 Steps:

1. Ek group banao:

```
groupadd webdev
```

2. User ko group me add karo:

```
usermod -aG webdev mukul
```

3. Directory ke permissions set karo:

```
chown -R root:webdev /var/www  
chmod -R 770 /var/www
```

3. Agar ek group ka kaam khatam ho gaya ho to usko delete kaise karoge?

 Steps:

```
groupdel groupname
```

 Par pehle ensure kar lo koi active user us group ka primary group na ho.

Networking Troubleshooting Scenarios

1. Agar server internet se connect nahi ho raha ho, to steps kya honge?

 Steps:

1. Check IP:

```
ip a
```

2. Gateway check:

```
ip route show  
ping <gateway-ip>
```

3. DNS check:

```
cat /etc/resolv.conf  
ping google.com
```

4. Service port check:

```
ss -tulnp
```

2. Agar server me port 80 pe application chal rahi hai par bahar se open nahi ho rahi, kya karoge?

 Steps:

1. Service bind check:

```
sudo lsof -i :80
```

2. Firewall check:

```
sudo ss -tulnp | grep 80
sudo firewall-cmd --list-all    # (RHEL)
sudo ufw status                  # (Ubuntu)
```

3. Cloud/VPC security group check (agar AWS/Azure pe ho).

3. Agar ek IP ko block karna ho (e.g., 192.168.1.100), kaise karoge?



iptables -A INPUT -s 192.168.1.100 -j DROP

(Ya firewalld / ufw rule add karke).

4. Agar ek hi server pe 2 network interfaces hain aur tumhe ek interface se outgoing traffic route karna ho, kaise karoge?



ip route add default via 192.168.1.1 dev eth0



Interview me kya highlight karna hai:

- **User/Group:** account disable, delete, special access.
- **Networking:** IP, Gateway, DNS, port, firewall, routing.
- **Practical mindset:** “Pehle basic check, phir advanced check.”

👉 Mukul bhai, ab batao tumhe kya chahiye next step me —

1. Main tumhe **20 most asked Linux interview Q&A (advance level)** bana ke du, ya
2. Tumhare liye **daily ek-ek live troubleshooting lab** ready karu jisme tum commands run karke seekho?

You said:

troubleshooting lab bna de

ChatGPT said:

Mast choice Mukul bhai 🌟 Troubleshooting labs karne se tumhari **soch + haath dono strong** ho jaayenge. Main tumhare liye **Daily Linux Troubleshooting Lab Series** banaata hoon. Tum apne system/VM/EC2 pe try kar sakte ho.

◊ Day 1 – Linux Troubleshooting Lab

⌚ Scenario 1: Disk Full

👉 Situation: /var partition full ho gaya hai, server slow ho raha hai.

❓ Tum kya karoge?

🔧 Steps (Try Yourself):

1. Disk usage check:

`df -h`

2. Top 5 directories jo space kha rahe hain:

`du -sh /var/* | sort -h | tail -5`

3. Logs clean karo:

```
> /var/log/syslog  
> /var/log/messages
```

4. Agar space chahiye immediately:

```
rm -rf /var/log/*.gz
```

⌚ Scenario 2: High CPU Usage

👉 Situation: Server slow hai, suspected CPU load.

🔧 Steps:

1. Check load average:

```
uptime
```

2. Top processes:

```
top
```

3. Specific process kill:

```
kill -9 <PID>
```

4. Nice priority change:

```
renice -n 10 -p <PID>
```

⌚ Scenario 3: Service Down

👉 Situation: Apache/Tomcat service chal nahi raha.

Steps:

1. Status check:

```
systemctl status apache2    # Ubuntu  
systemctl status httpd      # RHEL
```

2. Logs dekhna:

```
journalctl -xe  
tail -f /var/log/httpd/error_log
```

3. Restart:

```
systemctl restart apache2
```

4. Port check:

```
ss -tulnp | grep 80
```

Scenario 4: SSH Not Working

 Situation: Tumhe server SSH se access nahi ho raha.

Steps (agar console access hai):

1. Check SSH service:

```
systemctl status sshd
```

2. Firewall check:

```
firewall-cmd --list-all    # RHEL  
ufw status                  # Ubuntu
```

3. Port open:

```
firewall-cmd --add-port=22/tcp --permanent  
firewall-cmd --reload
```

⌚ Scenario 5: User Access Issue

👉 Situation: Ek user bol raha hai usko /project directory me access nahi mil raha.

🔧 Steps:

1. Check user groups:

```
id username
```

2. Check directory permissions:

```
ls -ld /project
```

3. Add user to group:

```
usermod -aG projectgroup username
```

⚡ Homework for You (Mukul bhai):

- Ye 5 scenarios apne system pe try karo aur mujhe batao kahan atke.
- Main **Day 2 Troubleshooting Lab** dunga (Networking: DNS issue, Port block, Multiple NICs, IP conflict).

👉 Bolo bhai, kya main tumhe **Daily Troubleshooting Lab Plan (7 days ka)** ready kar du jisme har din alag-alag scenario milega (Day 1: Disk/CPU, Day 2: Networking, Day 3:

Permissions, Day 4: Logs, Day 5: Storage/LVM, Day 6: Security, Day 7: Mixed Mock Interview)?

You said:

hnn do

ChatGPT said:

ठीक है Mukul 🌟 चल एक **Linux Troubleshooting Lab** बनाते हैं जो तुझे interview में भी काम आएगा और real-life practice में भी। मैं step-wise दे रहा हूँ:

◊ Linux Troubleshooting Lab

1. User & Group Issues

- Create a user and group

```
sudo useradd testuser  
sudo groupadd testgroup  
sudo usermod -aG testgroup testuser
```

- Error case: User को group से निकालना

```
sudo gpasswd -d testuser testgroup
```

- Delete user और group

```
sudo userdel testuser  
sudo groupdel testgroup
```

👉 Practice:

- Check करो कि /etc/passwd, /etc/group, /etc/shadow फाइल में changes हुए या नहीं।
- कभी-कभी user login नहीं कर पाता क्योंकि shell /bin/false set होती है → fix करना सीखो।

2. File Permission & Ownership

- Create a file and play with permissions

```
touch file1.txt
chmod 000 file1.txt
```

- अब check करो कि user access कर सकता है या नहीं।
- फिर fix करो chmod 644 file1.txt और chown user:user file1.txt से।

👉 Practice:

- SetUID, SetGID और Sticky bit लगाओ।
- समझो कि /tmp में sticky bit क्यों जरूरी है।

3. Disk Space Issues

- Check disk usage

```
df -h
du -sh /var/*
```

- Error Case: Log files भर जाएं → /var/log/ के अंदर बड़ी फाइल ढूँढो और rotate/delete करो।

👉 Practice:

- /var/log/messages या syslog को truncate करना।

```
sudo truncate -s 0 /var/log/syslog
```

4. Service Troubleshooting

- Apache/Tomcat install करके service fail करवाओ।

```
sudo systemctl start apache2
sudo systemctl status apache2
```

- Error check करो: journalctl -xe
- Port busy होने पर fix करो netstat -tulnp या ss -tulnp से।

5. Networking Issues

- Ping और DNS Troubleshooting

```
ping 8.8.8.8
ping google.com
cat /etc/resolv.conf
```

- IP assign करो manually और verify करो।

```
ip addr add 192.168.1.100/24 dev eth0
ip route add default via 192.168.1.1
```

👉 Practice:

- Gateway गलत set करके troubleshoot करना।
- Firewall block करके देखना (iptables या ufw)।

6. Process Troubleshooting

- High CPU process ढूँढना

```
top  
ps -ef | grep processname  
kill -9 <pid>
```

- Zombie processes check करना और kill करना।

7. Boot Issues

- GRUB recovery mode में login करो।
- Single-user mode से root password reset करना सीखो।

⚡ Practice Plan

तू ये सब labs खुद simulate करेगा:

1. User/Group misconfigurations
2. File permissions problem
3. Disk full
4. Service not starting
5. Network unreachable
6. High CPU/memory process
7. Boot failure (password recovery)

👉 अब बता, क्या मैं तेरे लिए **step-by-step lab guide with scenarios (problem + solution)** बनाऊँ, ताकि तू खुद problem create करे और फिर solve करे?

You said:

hn

ChatGPT said:

ठीक है भाई 🚀 अब मैं तुझे **Linux Troubleshooting Lab** का पूरा setup और scenario बता रहा हूँ, जिस पर तू practically काम करके interview level की तैयारी कर सकेगा।

🔧 Linux Troubleshooting Lab Scenarios

1. Disk Full Problem

- Issue: Server पर application crash हो रहा है क्योंकि / partition भर गया।
- Task:
 - Disk utilization चेक करना (df -h, du -sh *)
 - बड़े files identify करना (find / -type f -size +500M)
 - Unnecessary log files clear करना (/var/log/)

2. Service Not Starting

- Issue: Apache/Tomcat service start नहीं हो रही।
- Task:
 - Service status चेक करना (systemctl status apache2)
 - Logs चेक करना (journalctl -xe, /var/log/httpd/)
 - Port check करना (netstat -tulnp | grep 80)
 - Permission fix करना (अगर config file root owned हो गई है)।

3. User/Group Permission Problem

- Issue: Developer को app folder access नहीं मिल रहा।
- Task:
 - Permission check करना (ls -ld /app)
 - User का group membership verify करना (id devuser)
 - Proper permission देना (chmod 775, chown devuser:devgroup /app)

4. Network Connectivity Issue

- Issue: Server से DB connect नहीं हो रहा।
- Task:
 - Ping check करना (ping DB_IP)
 - Port open है या नहीं (telnet DB_IP 5432 या nc -zv DB_IP 5432)
 - Firewall rules चेक करना (iptables -L, ufw status)
 - /etc/resolv.conf और DNS check करना।

5. Process Hung Issue

- Issue: Server slow हो गया है।
- Task:
 - High CPU process चेक करना (top, htop)
 - Zombie processes kill करना (ps aux | grep defunct)
 - Memory hogging process identify करना (free -m, vmstat 1)

6. SSH Login Issue

- Issue: User SSH नहीं कर पा रहा।
- Task:
 - Check authorized_keys permissions (~/.ssh/authorized_keys 600)
 - Verify sshd config (/etc/ssh/sshd_config)
 - Fail2Ban या firewall block check करना।

7. Sticky Bit / SUID / SGID Check

- Issue: Security audit में इनका इस्तेमाल validate करना।
- Task:
 - Sticky bit: /tmp directory पर check (ls -ld /tmp)
 - SUID program check (ls -l /usr/bin/passwd)
 - SGID program check (ls -l /usr/bin/locate)

8. LVM Disk Expansion

- Issue: /home partition भर गया।
- Task:
 - New disk attach करना और partition बनाना।
 - PV create → VG extend → LV extend।
 - Filesystem resize करना (resize2fs)।

💡 कैसे Lab बनाए?

- एक VM बना ले (Ubuntu या CentOS)।
- Minimum 2-3 users create कर।

- Apache install कर (service troubleshooting practice)।
- एक extra disk attach कर (LVM practice)।
- Network में 2 VM use कर (ping, ssh, DB connection troubleshoot)।

👉 अब ये सब scenarios practice कर, तेरे पास **real-life Linux admin lab** ready हो जाएगा।

क्या मैं तेरे लिए इन सबका **step-by-step hands-on guide with commands** भी बना दूँ (जैसे एक-एक scenario practical करके दिखाना)?

THE FINAL BOSS PROJECT: Build & Secure a Real Production-Style Linux Server
*Agar tu isse crack kar gaya, to tu **full-fledged system admin** banne ke liye taiyaar hai — no more theory, only real-world skill.*

💧 Project Name: Iron Fortress Deployment

🎯 Goal:

Set up a secure, automated, production-ready **web server stack** using:

- **NGINX (reverse proxy + SSL)**
- **Docker container (backend app)**
- **Firewall (UFW)**
- **Fail2ban (brute-force protection)**
- **Auto backups with rsync**
- **Ansible (to deploy and configure all)**
- **Monitoring (glances + logs)**

💻 System Requirements

Component	Tool / Service
t	

OS	Ubuntu Server (EC2 or VM)
Web Server	NGINX (reverse proxy)
App Server	Docker container (Node.js or Flask app)
Security	UFW + Fail2ban
Monitoring	htop, glances, journalctl
Automation	Ansible
Backup	rsync (daily cron)

Step-by-Step Implementation Plan

STEP 1: Base OS Setup

```
sudo apt update && sudo apt upgrade -y  
sudo hostnamectl set-hostname iron-fortress
```

Create non-root sudo user:

```
adduser devadmin  
usermod -aG sudo devadmin
```

STEP 2: Secure SSH

```
sudo vim /etc/ssh/sshd_config  
# Set:  
PermitRootLogin no  
PasswordAuthentication no  
  
# Reload SSH  
sudo systemctl reload sshd
```

Setup key-based SSH login from your laptop.

STEP 3: Setup Docker App

1. Install Docker:

```
sudo apt install docker.io -y  
sudo usermod -aG docker devadmin
```

2. Pull sample app container:

```
docker run -d -p 5000:5000 --name webapp ghcr.io/heroku/python-getting-started
```

3. Check:

```
curl http://localhost:5000
```

STEP 4: Install & Configure NGINX

```
sudo apt install nginx -y
```

Set up reverse proxy:

```
# /etc/nginx/sites-available/webapp  
server {  
    listen 80;  
    server_name your_domain_or_ip;  
  
    location / {  
        proxy_pass http://localhost:5000;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
    }  
}
```

Enable it:

```
sudo ln -s /etc/nginx/sites-available/webapp /etc/nginx/sites-enabled/  
sudo nginx -t  
sudo systemctl restart nginx
```

STEP 5: SSL with Certbot (Let's Encrypt)

```
sudo apt install certbot python3-certbot-nginx -y  
sudo certbot --nginx
```

Set auto-renewal:

```
sudo crontab -e  
# Add:  
0 0 * * * certbot renew --quiet
```

STEP 6: Enable Firewall & Fail2ban

```
sudo ufw allow OpenSSH  
sudo ufw allow 'Nginx Full'  
sudo ufw enable  
  
sudo apt install fail2ban -y  
sudo systemctl enable fail2ban
```

STEP 7: Configure Backups with rsync

Create backup script:

```
vim /opt/backup.sh
```

```
#!/bin/bash
rsync -a /var/www/ /backups/$(date +%F)
```

Make it executable:

```
chmod +x /opt/backup.sh
```

Schedule cron job:

```
crontab -e
# Add:
0 2 * * * /opt/backup.sh
```

STEP 8: System Monitoring

Install tools:

```
sudo apt install glances htop -y
```

Check logs:

```
journalctl -xe
tail -f /var/log/syslog
```

STEP 9: Automate With Ansible (Optional Boss Mode)

On your control machine (not server):

```
sudo apt install ansible -y
```

```
# Create inventory:
echo "iron ansible_host=<IP> ansible_user=devadmin" > hosts

# Run ping test:
```

```
ansible -i hosts all -m ping
```

Write a playbook to:

- Install NGINX
- Deploy Docker app
- Copy NGINX config
- Enable UFW
- Schedule backup

❖ FINAL VALIDATION:

Checkpoint	Status
🔒 SSH only via key	✓
🔥 Firewall enabled	✓
🐳 Docker app running	✓
🌐 NGINX reverse proxy working	✓
🔒 SSL working	✓
⚠️ Fail2ban banning SSH brute-force	✓
📅 Daily backups running	✓
📈 Monitoring active	✓
⚙️ Ansible deployment tested	✓

██ YOU MADE IT

Tere system ka naam ab **Iron Fortress** hai —

kyunki tu us server ko aise secure, automate aur deploy kar raha hai jaise **kisi devops god ka haath ho tere upar.** 😊

Ab agla level kya?

- Kubernetes this whole thing?
- Setup CI/CD?
- Use Terraform for infra automation?
- Centralized logging with ELK?

Tu bas bol, agla boss project bhi line me hai.

Bole to — "Agla Level Lao Bhai 🔥 "