# Linux Scenario Practice Guide for DevOps Engineers

This guide is designed for DevOps engineers to practice real-world Linux scenarios. Each scenario includes the situation, question, and answer with practical commands you'll use daily in production environments.

## 1. Disk Space Alert

**Situation:** Jenkins jobs are failing due to low disk space.

**Solution:**
```
df -h; du -sh /* 2>/dev/null | sort -hr | head -10; find /var/log -type f -name
'*.log' -mtime +7 -delete; docker system prune -a --volumes
```

## 2. High CPU Usage

**Situation:** Web server is slow due to CPU hogging.

**Solution:**
```
top; ps -eo pid,comm,%cpu,%mem --sort=-%cpu | head; kill -9
```

## 3. SSH Access Denied

**Situation:** A new DevOps user cannot SSH into the server.

**Solution:**
```
id devopsuser; ls -ld /home/devopsuser/.ssh; cat
/home/devopsuser/.ssh/authorized_keys; systemctl status sshd
```

## 4. Application Port Not Listening

**Situation:** Nginx running but website unreachable.

**Solution:**
```
ss -tulnp | grep nginx; curl -I http://localhost:80; firewall-cmd --permanent
--add-port=80/tcp; firewall-cmd --reload
```

## 5. Service Fails After Reboot

**Situation:** Docker doesn't start automatically.

**Solution:**
```
systemctl enable docker; systemctl start docker; systemctl status docker
```

## 6. Log File Analysis

**Situation:** Application crashed last night.

**Solution:**
```
grep -i 'error' /var/log/syslog | tail -20; journalctl -u nginx --since 'yesterday'
```

## 7. Network Connectivity Issue

**Situation:** Ping to internal server fails.

**Solution:**
```
ping ; ip a; ip route; systemctl status NetworkManager
```

### 8. File Permission Error

**Situation:** Jenkins reports 'Permission denied'.

**Solution:**
```
ls -l /var/lib/jenkins; chown -R jenkins:jenkins /var/lib/jenkins; chmod -R 755
/var/lib/jenkins
```

### 9. Automation with Cron

**Situation:** Run cleanup script daily.

**Solution:**
```
crontab -e; 0 0 * * * /usr/local/bin/cleanup.sh
```

### 10. Package Installation

**Situation:** Install jq and git on RHEL.

**Solution:**
```
yum install -y jq git
```

### 11. Check System Uptime

**Situation:** Check how long system has been up.

**Solution:**
```
uptime; who -b
```

### 12. Find Recently Modified Files

**Situation:** Find modified files in last 24h.

**Solution:**
```
find /path/to/dir -type f -mtime -1
```

### 13. Port Conflict

**Situation:** Port 8080 is busy.

**Solution:**
```
lsof -i :8080; netstat -tulnp | grep 8080
```

### 14. Copy Files Between Servers

**Situation:** Copy backup file to another node.

**Solution:**
```
scp /backup/db_backup.sql user@10.10.10.2:/data/backups/
```

### 15. File System Read-Only Error

**Situation:** Can't save logs: Read-only FS.

**Solution:**
```
mount -o remount,rw /; df -hT; dmesg | tail -20
```

### 16. Docker Container Not Starting

**Situation:** Container exits immediately.

**Solution:**
```
docker ps -a; docker logs ; docker inspect | grep Error
```

### 17. Docker Image Consuming Space

**Situation:** Disk full due to Docker images.

> **Solution:**
```
docker system df; docker system prune -a --volumes; du -sh /var/lib/docker/*
```

### 18. Docker Network Issue

**Situation:** Containers can't reach each other.

> **Solution:**
```
docker network ls; docker network inspect ; docker exec -it ping
```

### 19. Ansible Permission Error

**Situation:** Playbook fails with permission issue.

> **Solution:**
```
ansible-playbook site.yml -vvv; ansible all -m ping; ansible all -a 'whoami'
```

### 20. Dynamic Inventory

**Situation:** Auto-discover EC2 instances via Ansible.

> **Solution:**
```
ansible-inventory -i aws_ec2.yml --list
```

### 21. Kubernetes Pod CrashLoopBackOff

**Situation:** Pod restarts repeatedly.

> **Solution:**
```
kubectl describe pod ; kubectl logs --previous
```

### 22. Node NotReady

**Situation:** Kubernetes worker node shows NotReady.

> **Solution:**
```
kubectl describe node ; df -h; du -sh /var/lib/containerd /var/lib/kubelet
```

### 23. Pod External Connection Issue

**Situation:** Pod can't connect to external API.

> **Solution:**
```
kubectl exec -it -- curl -v https://api.example.com; kubectl get networkpolicy
```

### 24. ConfigMap Not Updated

**Situation:** Pods not showing latest config.

> **Solution:**
```
kubectl rollout restart deployment
```

### 25. High System Load

**Situation:** Prometheus alert: high load average.

> **Solution:**
```
uptime; top; vmstat 1 5; iostat -xz 1 5
```

### 26. Time Desync

**Situation:** Timestamps differ between servers.

```
Solution:
timedatectl; systemctl restart chronyd
```

## 27. SELinux Blocking App

**Situation:** Permission denied due to SELinux.

```
Solution:
getenforce; setenforce 0
```

## 28. Firewall Blocking App

**Situation:** Grafana port not accessible.

```
Solution:
firewall-cmd --list-all; firewall-cmd --permanent --add-port=3000/tcp; firewall-cmd
--reload
```

## 29. Deleted Log File Still in Use

**Situation:** File missing but process writing.

```
Solution:
lsof | grep deleted
```

## 30. Backup Database Script

**Situation:** Automate PostgreSQL backups daily.

```
Solution:
pg_dump -U postgres mydb > /backup/mydb_$(date +%F).sql; find /backup -type f -mtime
+7 -delete
```

**Keep Practicing!**
Use these scenarios daily to strengthen your Linux troubleshooting and automation skills.
Remember — mastery comes from repetition, observation, and automation.