

DevOps Scenario Based Commands

Scenario 1: Website is DOWN (most common) ----

❶ Check server is reachable → ping server_ip

Why: Network issue or server down?

❷ Check CPU & Memory → top / htop

Why: High CPU/RAM can crash apps

❸ Check disk space → df -h

❹ Check running processes → ps -ef | grep nginx / systemctl status nginx.

Scenario 2: Application not accessible but server is UP -----

❶ Check service status → systemctl status nginx.

❷ Restart service → systemctl restart nginx.

❸ Check logs → journalctl -u nginx --since "10 minutes ago" OR
tail -f /var/log/nginx/error.log

Why: Logs show exact error (port, permission, crash)

Scenario 3: Port not working (502 / 503 error) -----

Check port listening → netstat -tulnp | grep 80 / ss -tulnp | grep 80.

Check firewall → iptables -L / ufw status.

Scenario 4: Disk FULL alert

Find large files → du -sh /* | sort -h

Check logs size → ls -lh /var/log/

Clean safely → rm -f /var/log/*.log

Scenario 5: Service not starting

Check service error → systemctl status docker

Reload config → systemctl daemon-reexec

Check failed services → systemctl --failed

Scenario 6: Docker container issue (real DevOps daily work) -----

Check containers → `docker ps -a`

Check logs → `docker logs container_id`

Restart container → `docker restart container_id`

Scenario 7: Deployment issue – “My code is not deployed”

Check Git branch → `git branch`

Pull latest code → `git pull origin main`

Check build logs → `tail -f /var/log/application.log`

Scenario 8: Permissions issue (VERY common)

Check permission → `ls -l file.sh`

Fix permission → `chmod +x file.sh`

Fix ownership → `chown user:user file.sh`

Scenario 9: Network troubleshooting -----

curl <http://localhost:8080>

telnet localhost 8080

traceroute google.com

Scenario 10: Monitoring & logs (production habit)-----

Uptime

`free -m`

`watch df -h`

`tail -f /var/log/syslog`

----- INTERVIEW TIP -----

When interviewer asks:

“Website is down, what will you do?”

ANS == Network → Server → Resource → Service → Logs → Restart → Monitor