🧠 Linux & DevOps Troubleshooting Cheat Sheet (Chat Version)

(Beginner → Interview → Real Job)

🔰 UNIVERSAL DEBUGGING RULE (MEMORIZE THIS FIRST)

> **Never run commands randomly.**
> **Always ask questions.**

```
What is broken?
Where is traffic stopping?
Which layer owns the problem?
```

🧭 TRAFFIC FLOW MODEL (ROOT OF ALL DEBUGGING)

```
Client → Network → Firewall → Service → Application → Data
```

Always debug **top to bottom**, never sideways.

1️⃣ `systemctl` — SERVICE CONTROL (FIRST COMMAND ALWAYS)

❓ Question it answers

> Is the service alive? Can it start? Will it survive reboot?

🧠 When to use

- Website down
- App not responding
- DB connection issue
- After changing configs

🔹 Commands

```
systemctl status nginx      # is it running?
systemctl start nginx       # start service
systemctl stop nginx        # stop service
systemctl restart nginx     # restart after config change
systemctl enable nginx      # start at boot
systemctl disable nginx     # disable at boot
```

🔍 Interpretation

Output	Meaning
active (running)	Service OK
inactive	Service stopped
failed	Service crashed → check logs

👉 **If service is DOWN, stop everything and fix this first**

2️⃣ `journalctl` — WHY DID IT FAIL?

❓ Question it answers

> What exactly went wrong?

🧠 When to use

- `systemctl` shows **failed**
- Service won't start
- Unexpected crash

🔹 Commands

```
journalctl -xe
journalctl -u nginx
journalctl -u httpd
```

🔥 Common errors you'll see

- Port already in use
- Permission denied
- Syntax error in config
- Missing files

👉 **journalctl explains failures — systemctl only reports them**

3️⃣ `ss` / `netstat` — PORT & PROCESS CHECK

❓ Question it answers

> Is the port open? Who is using it?

🧠 When to use

- Service is running
- But not reachable
- Suspect port conflict

🔹 Commands

```
ss -tulnp
netstat -tulnp
```

🔍 Important flags

Flag	Meaning
`-t`	TCP
`-u`	UDP

Flag	Meaning
`-l`	Listening
`-n`	Numeric
`-p`	Process

🔥 CRITICAL INSIGHT

Binding	Meaning
127.0.0.1	Local only ❌
0.0.0.0	Network accessible ✅

4️⃣ `telnet` — NETWORK CONNECTIVITY TEST

❓ Question it answers

> Can I reach this port from another machine?

🧠 When to use

- Local curl works
- Remote access fails
- Firewall suspected

🔹 Command

```
telnet server-ip port
```

🔍 Interpretation

Result	Meaning
Connected	Network OK

Result	Meaning
Connection refused	Service down
No route to host	Firewall / routing

5️⃣ `curl` — APPLICATION LAYER TEST

❓ Question it answers

> Is the application responding correctly?

🧠 When to use

- After port/network is OK
- Testing HTTP/HTTPS
- Verifying LB/backend

🔹 Commands

```
curl http://server:port
curl -I http://server
curl -Ik https://server
```

🔍 What curl tests

- HTTP status codes
- Headers
- SSL handshake

6️⃣ CONFIG VALIDATION (NEVER SKIP)

❓ Question it answers

> Is my config syntactically correct?

🧠 When to use

- Before restarting a service

🔹 Commands

```
nginx -t
apachectl configtest
```

👉 **This saves production outages**

7️⃣ FIREWALL DEBUGGING

❓ Question it answers

> Is traffic blocked?

🧠 When to use

- Service running
- Port open locally
- Remote access fails

🔹 firewalld

```
firewall-cmd --list-ports
```

🔹 iptables

```
iptables -L -n
iptables -L -n --line-numbers
```

🔥 Red flag

```
No route to host
```

→ almost always firewall

8️⃣ FILESYSTEM QUICK MAP (MENTAL MODEL)

Purpose	Path
Configs	`/etc/<service>`
Logs	`/var/log`
Binaries	`/usr/bin`, `/usr/sbin`
Web files (nginx)	`/usr/share/nginx/html`
Web files (apache)	`/var/www/html`
Temp files	`/tmp`
App data	`/var/lib`

9️⃣ SYSTEM HEALTH COMMANDS

```
top           # CPU, memory
free -m       # RAM
df -h         # Disk space
ps -ef        # Processes
```

🔟 LOAD BALANCER DEBUG FLOW

```
1. Is LB running?       → systemctl
2. Is port open?        → ss/netstat
3. Can reach backend? → curl from LB
4. Config correct?     → nginx.conf + nginx -t
5. Restart              → systemctl restart
```

🧠 GOLDEN RULES (INTERVIEW READY)

- Never guess ports
- Validate configs before restart
- Logs explain failures
- Local success ≠ network success
- Firewall issues fail silently

🎯 ONE-LINE DEVOPS FORMULA

```
Service → Logs → Port → Network → Firewall → App
```

Perfect 👍
Here are **10 REAL DevOps / Linux troubleshooting scenarios**, exactly like what you'll see in:

- KodeKloud Engineer
- Interviews
- Real production incidents

For **each scenario**, I'll give you:

- ❓ Problem
- 🧠 How to THINK (mental model)
- 🛠️ What to CHECK (commands)
- 🎯 Root cause (typical)
- ✅ Fix (high level)

🔥 10 Real Troubleshooting Scenarios (Beginner → Pro)

🧩 Scenario 1: Website Not Loading

❓ Problem

> `curl http://server` fails

🧠 Think

- Is the web service running?

- Is port open?
- Network or firewall?

🛠️ Check

```
systemctl status nginx
ss -tulnp | grep nginx
curl http://localhost
```

🎯 Root Cause

- Service stopped
- Wrong port
- Config error

✅ Fix

- Start service
- Fix config
- Restart nginx

🧩 Scenario 2: Service Won't Start After Restart

❓ Problem

> `systemctl start httpd` fails

🧠 Think

- Why did it fail?
- Port conflict?
- Config error?

🛠️ Check

```
journalctl -xe
apachectl configtest
```

```
ss -tulnp
```

🎯 Root Cause

- Port already in use
- Syntax error

✅ Fix

- Free port
- Correct config

🧩 Scenario 3: Works Locally but Not from Another Server

❓ Problem

> `curl localhost` works
> `curl server-ip` fails

🧠 Think

- Firewall
- Bind address

🛠️ Check

```
ss -tulnp
iptables -L -n
firewall-cmd --list-ports
```

🎯 Root Cause

- Bound to 127.0.0.1
- Firewall blocking

✅ Fix

- Bind to 0.0.0.0
- Open port

🧩 Scenario 4: Apache Running but Site Down

❓ Problem

> `systemctl status httpd` = running
> Page not loading

🧠 Think

- Port mismatch
- Document root wrong

🛠️ Check

```
ss -tulnp | grep httpd
ls /var/www/html
```

🎯 Root Cause

- Apache listening on different port
- Missing index.html

✅ Fix

- Use correct port
- Fix content

🧩 Scenario 5: Load Balancer Returns 502 Bad Gateway

❓ Problem

> LB up but backend not reachable

🧠 Think

- Can LB reach backend?
- Correct upstream?

🛠️ Check

```
curl http://backend-ip:port
nginx -t
```

🎯 Root Cause

- Wrong backend port
- Backend service down

✅ Fix

- Fix upstream
- Start backend service

🧩 Scenario 6: SSH Login Fails Suddenly

❓ Problem

> `ssh user@server` fails

🧠 Think

- Service running?
- Firewall?
- Auth issue?

🛠️ Check

```
systemctl status sshd
journalctl -u sshd
iptables -L -n
```

🎯 Root Cause

- sshd stopped
- Firewall blocking port 22

✅ Fix

- Start sshd
- Open port 22

🧩 Scenario 7: Disk Full – Services Fail Randomly

❓ Problem

> Services crash unexpectedly

🧠 Think

- Disk space?
- Logs filling disk?

🛠️ Check

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

🎯 Root Cause

- Disk 100% full

✅ Fix

- Clean logs

- Rotate logs

🧩 Scenario 8: App Can’t Connect to Database

❓ Problem

> DB connection error

🧠 Think

- DB running?
- Port reachable?

🛠️ Check

```
systemctl status mariadb
ss -tulnp | grep 3306
telnet db-ip 3306
```

🎯 Root Cause

- DB down
- Firewall blocking

✅ Fix

- Start DB
- Open DB port

🧩 Scenario 9: Cron Job Not Running

❓ Problem

> Scheduled task not executing

🧠 Think

- Cron running?
- Correct user?
- Permission?

🛠️ Check

```
systemctl status crond
crontab -l
ls -l script.sh
```

🎯 Root Cause

- crond stopped
- Script not executable

✅ Fix

- Start crond
- chmod +x script

🧩 Scenario 10: Service Works Until Reboot

❓ Problem

> Works now, fails after reboot

🧠 Think

- Enabled at boot?

🛠️ Check

```
systemctl is-enabled nginx
```

🎯 Root Cause

- Service not enabled

✅ Fix

```
systemctl enable nginx
```

🧠 HOW TO PRACTICE THESE (IMPORTANT)

For each scenario:
1️⃣ Read the problem
2️⃣ Say **out loud** what layer is failing
3️⃣ Choose commands **before typing**
4️⃣ Explain WHY each command is used

🎯 Interview Tip

If you explain **Scenario + Thought Process**,
interviewers stop asking more questions.
