Chapter 10: Monitoring & Troubleshooting

1. Why Monitoring & Troubleshooting Matters (Interview Context)

In real production environments:

- Things **will break**
- Performance **will degrade**
- Disks **will fill**
- Services **will crash**

Interviewers want to see:

- Your **approach**
- Your **order of checks**
- Your **calm, logical thinking**

Interview insight:

Good engineers don’t panic — they follow a structured troubleshooting flow.

---

2. The Golden Troubleshooting Rule (VERY IMPORTANT)

When something is wrong, always check in this order:

1. CPU
2. Memory
3. Disk
4. Network
5. Application logs

Interview line (say this confidently):

“I start with CPU and memory, then disk I/O, then network, and finally logs.”

---

3. Monitoring CPU Usage

Tools to Use

```
top
htop
uptime
```

What to Look For

- High `%CPU`
- High load average
- One process consuming most CPU

Interview insight:

High load does not always mean high CPU — it can be I/O wait.

---

4. top vs htop

top

- Default tool
- Available everywhere
- Keyboard-driven

htop

- Interactive
- Colorful
- Easier to read
- May not be installed by default

Interview line:

htop is more user-friendly, but top is universally available.

---

5. Monitoring Memory Usage

Commands

```
free -h
vmstat
```

What to Check

- **Available memory**
- **Swap usage**
- **Increasing memory over time**

Interview insight:

Low free memory is normal; low available memory is not.

---

6. Checking Disk Usage

Disk Space

```
df -h
```

Disk Usage per Directory

```
du -sh /*
```

Disk I/O Issues

```
iostat
```

Interview insight:

Disk I/O issues often cause high load with low CPU usage.

---

7. df vs du (Common Interview Question)

Command	Purpose
df	Disk free space (filesystem level)
du	Disk usage (directory/file level)

Interview line:

df shows filesystem usage, du shows directory usage.

---

8. vmstat

Purpose

vmstat provides:

- **Memory statistics**
- **CPU scheduling**
- **I/O activity**

Example:

```
vmstat 1
```

Interview insight:

vmstat is useful for spotting memory pressure and swapping.

---

9. iostat

Purpose

iostat monitors:

- Disk read/write activity
- I/O wait times

Use case:

- Slow application
- High load average
- Disk-related bottlenecks

Interview line:

iostat helps diagnose disk performance issues.

---

10. lsof (List Open Files)

Purpose

Shows which files are opened by which processes.

Example:

```
lsof -i :8080
```

Use cases:

- **Port already in use**
- **File not releasing disk space**

Interview insight:

Deleted files still consume space if a process holds them open.

---

11. strace (Advanced but Asked)

Purpose

strace traces system calls made by a process.

Example:

```
strace -p PID
```

Use cases:

- **Application hanging**
- **Debugging runtime behavior**

Interview note:

strace is powerful but used carefully in production.

---

12. Checking Logs (VERY IMPORTANT)

System Logs

```
journalctl
journalctl -xe
```

Service Logs

```
journalctl -u nginx
```

Interview insight:

Logs often tell you exactly what is wrong.

---

13. Analyzing System Slowness (Interview Favorite)

Step-by-Step Approach

1. **Check CPU (top)**
2. **Check memory (free, vmstat)**
3. **Check disk space (df)**
4. **Check disk I/O (iostat)**
5. **Check network**
6. **Check logs**

Interview line:

I isolate the bottleneck first, then fix the cause.

---

14. Real-Life Production Scenarios

Scenario 1: Server Is Slow

- Check CPU and load
- Check memory and swap
- Check disk I/O
- Review logs

Scenario 2: Disk Is 100% Full

- Identify filesystem (df)
- Find large directories (du)
- Truncate logs
- Rotate logs

Scenario 3: Service Not Responding

- Check service status
- Check port
- Check logs
- Restart if needed

Chapter 10: Interview Takeaways

After this chapter, you should be able to:

- Monitor CPU, memory, disk, and I/O
- Explain top vs htop
- Use df, du correctly
- Diagnose disk and memory bottlenecks
- Use lsof and strace when needed
- Follow a structured troubleshooting approach
