

## General & Problem-Solving Questions

### ◇ **Introduce yourself.**

A DevOps Engineer with expertise in CI/CD, automation, cloud computing, and container orchestration. Passionate about streamlining deployments, optimizing infrastructure, and enhancing security.

### ◇ **Describe a complex situation using the STAR method.**

- **Situation:** A production deployment failed due to misconfigured Kubernetes secrets.
- **Task:** Identify and resolve the issue quickly to minimize downtime.
- **Action:** Used `kubectl describe pod` to find the error, updated the secret using `kubectl apply`, and redeployed.
- **Result:** Deployment was restored in 10 minutes, and post-mortem led to implementing automated secret validation in CI/CD.

◇ **Was this situation completely new, or had it occurred before?** It had occurred before, but a lack of automation prolonged resolution. The experience led to improvements in pipeline security.

---

## Git & Version Control

### ◇ **Difference between git push and git fetch with examples?**

- `git push` uploads local changes to the remote repository (`git push origin main`).
- `git fetch` retrieves the latest changes without merging (`git fetch origin main`).  
Use `git fetch` when you want to inspect changes before integrating them.

### ◇ **Recover deleted changes using Git commands?**

- **If unstaged:** `git checkout -- <file>`
  - **If committed but not pushed:** `git reset --hard HEAD~1`
  - **If pushed:** `git revert <commit>`
- ### ◇ **Useful but often ignored Git commands?**
- `git stash`: Save work without committing.
  - `git bisect`: Find bugs by binary search.
  - `git cherry-pick`: Apply a specific commit.
-

## CI/CD & Security

- ◇ **How do you secure your CI/CD pipeline?**
    - Use **IAM roles** to restrict access.
    - Scan code with **SAST/DAST tools** (SonarQube, Trivy).
    - Use **signed container images** (Notary, Sigstore).
    - Encrypt secrets with **Vault or AWS Secrets Manager**.
  - ◇ **Why is open-source preferred for cloud deployment?**
    - Cost-effective and flexible.
    - Community-driven innovation. ☑ Avoids vendor lock-in.
- 

## Containerization & Kubernetes

- ◇ **What is containerization?**

It is the process of packaging applications with dependencies into lightweight, portable containers (e.g., Docker).

- ◇ **Kubernetes architecture & node communication?**

- **Control Plane:** Manages the cluster.
- **Worker Nodes:** Run application workloads.
- Nodes communicate via API server using **Kubelet**.

- ◇ **Pod lifecycle stages?**

Pending → Running → Succeeded/Failed → Terminating

- ◇ **Do pods always contain a single container?**

No. Pods can have:

- **Single-container pods** (most common).
- **Multi-container pods** (sidecar, init containers).

- ◇ **How to write a Dockerfile?**

FROM python:3.9

WORKDIR /app

COPY . .

RUN pip install -r requirements.txt

CMD ["python", "app.py"]

◇ **Secure container images for production?**

- Use **distroless** or minimal base images (Alpine, Ubuntu Core).
  - Regularly **scan images** for vulnerabilities.
- 

## **AWS & Infrastructure**

◇ **Fault tolerance & disaster recovery in AWS?**

- **Multi-AZ & Multi-Region deployments.**
- **Auto Scaling & Load Balancing.**
- **Backups using AWS Backup & RDS Snapshots.**

◇ **What is a NAT Gateway?**

Allows private subnet instances to access the internet without exposing them directly.

◇ **Why aren't you using WAF?**

WAF protects against web attacks, but alternative security layers (e.g., CloudFront, Shield) may be used.

◇ **Where are your databases hosted?**

Depends on setup:

- **Amazon RDS** (managed relational DBs).
- **DynamoDB** (NoSQL).
- **Self-hosted on EC2** (custom setups).

◇ **Does caching reduce costs?**

Yes, caching reduces database/API calls using **Amazon ElastiCache** (**Redis/Memcached**) or **CloudFront**.

◇ **Ingress traffic setup for ELB?**

- Security Groups & Network ACLs. ☑ Restrict IPs, enforce HTTPS.

◇ **Handling large data storage?**

- Use **S3 for object storage.**
- **AWS EMR** for big data processing.

◇ **What is a CDN & why use it?**

A **Content Delivery Network (CDN)** caches content at edge locations to reduce latency (e.g., AWS CloudFront).

◇ **Still experiencing latency after implementing a CDN?**

Check:

- **Origin server response time.**
  - **Cache expiration settings.**
  - **Network congestion.**
- 

## Networking & Linux

◇ **Common HTTP status codes?**

- 200 OK – Success
- 404 Not Found – Resource missing
- 500 Internal Server Error – Server failure
- 502 Bad Gateway – Bad upstream response
- 516 Unknown Error – Custom application error

◇ **Replicating an EC2 instance in another VPC?**

- Create an **AMI** and launch in the target VPC.
- Use **AWS DMS or S3 Sync** for data migration. ? **Cross-account AMI sharing** is possible.

◇ **Check open ports on Linux?**

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

◇ **Difference between curl and wget?**

- curl fetches data but doesn't save by default.
- wget downloads files and supports resumption.

◇ **Schedule backups using cron?**

```
crontab -e  
0 2 * * * tar -czf /backup/data.tar.gz /var/www  
(Runs at 2 AM daily)
```

◇ **EBS volume full—what do you do?**

- Check logs (df -h, du -sh).
- Remove unnecessary files.
- If needed, resize volume via AWS Console.

◇ **Check if a Linux service is running?**

```
systemctl status nginx ps aux |  
grep nginx
```

◇ **Troubleshoot slow applications?**

- **Check CPU/Memory** (top, htop).
- **Analyze logs** (journalctl, dmesg).
- **Monitor network latency** (ping, traceroute).

◇ **What is DNS caching?**

DNS caching stores resolved domain names locally to reduce lookup time, improving performance.

---

## Final Thoughts

Mastering these **DevOps, AWS, and Linux** concepts will boost your expertise and help troubleshoot real-world issues effectively! 💡

#DevOps #AWS #Git #Kubernetes #Docker #CloudComputing #Linux #Automation  
#Networking #CI\_CD #CloudSecurity #Infrastructure #Coding #Programming  
#Python #Scripting #SoftwareDevelopment #DevOps #Automation  
#CloudComputing #InfrastructureAsCode #Kubernetes #Docker #Git #CI\_CD  
#AWS #Linux #ShellScripting #Tech #Cloud #Developer #CodeNewbie  
#100DaysOfCode #TechCommunity #OpenSource #FullStack #Backend #Frontend  
#IT #CyberSecurity #MachineLearning #AI #DataScience #WebDevelopment  
#CodingLife