DevOps Interview Questions & Answers

1. Explain your CI/CD pipeline end-to-end

Code is pushed to Git → CI tool (Jenkins/GitHub Actions) triggers → build & unit tests → static code analysis → Docker image build → image pushed to registry → deployment via Helm/Kubernetes → automated tests → monitoring & alerts.

2. How do you handle rollback in deployment?

Rollback using previous stable image tags, Kubernetes rollout undo, Helm rollback, or Blue-Green switch. Versioned artifacts help revert safely.

3. Difference between Continuous Integration & Continuous Delivery

CI focuses on frequent code integration with automated builds/tests. CD ensures code is always in a deployable state and can be released automatically.

4. What is Blue-Green deployment?

Two environments (Blue=live, Green=new). Traffic switches after validation, enabling zero-downtime and easy rollback.

5. What challenges have you faced in a pipeline?

Dependency failures, flaky tests, slow builds, secrets handling, environment drift, permission issues, and rollback failures.

6. What is the use of a Dockerfile?

A Dockerfile defines instructions to build a Docker image including base image, dependencies, configuration, and startup command.

7. What is a multi-stage build, and why do we use it?

Uses multiple FROM stages to separate build and runtime. Reduces final image size and improves security.

8. How do you reduce Docker image size?

Use slim/alpine images, multi-stage builds, remove unused packages, minimize layers, and use .dockerignore.

9. Difference between Docker Image & Container

Image is a read-only template. Container is a running instance of that image.

10. How do you clean unused containers/images?

Using docker system prune, docker image prune, and docker container prune.

11. What happens internally when you run kubectl apply?

kubectl sends request to API server → validation → etcd update → controller reconciles desired vs current state → scheduler & kubelet act.

12. Explain Pod vs Deployment vs ReplicaSet

Pod: smallest unit running containers. ReplicaSet: ensures desired pod count. Deployment: manages ReplicaSets and rolling updates.

13. How do you troubleshoot a CrashLoopBackOff?

Check logs, describe pod, verify env vars, probes, image, resource limits, and application errors.

14. What is a Readiness & Liveness probe?

Liveness checks if app should restart. Readiness checks if app can receive traffic.

15. How can you expose a service externally?

Using NodePort, LoadBalancer, or Ingress.

16. How do you find a high CPU process?

Using top, htop, ps -eo pid,cmd,%cpu.

17. How do you check disk usage?

Using df -h for filesystem usage and du -sh for directories.

18. Explain top, df -h, and netstat usage

top: process usage, df -h: disk usage, netstat: network connections and ports.

19. What is load average?

Average number of runnable or waiting processes over 1, 5, 15 minutes.

20. How do you handle log rotation?

Using logrotate with size/time-based rotation and compression.

21. What is Auto Scaling?

AWS service that automatically adjusts EC2 capacity based on demand.

22. Difference between ALB, NLB & CLB

ALB: Layer 7 HTTP routing, NLB: Layer 4 high performance, CLB: legacy load balancer.

23. How do you secure S3 buckets?

Use IAM policies, bucket policies, block public access, encryption, and versioning.

24. What are IAM Roles & Policies?

Policies define permissions; roles allow AWS services/users to assume those permissions.

25. Explain VPC, Subnets & Security Groups

VPC is a virtual network, subnets divide it, security groups act as virtual firewalls.

26. What is a Terraform state file?

It tracks infrastructure resources and their current state.

27. How do you manage secrets?

Using AWS Secrets Manager, SSM Parameter Store, Vault, or encrypted tfvars.

28. Difference between plan, apply, and destroy

plan previews changes, apply creates/updates infra, destroy removes resources.

29. What are Terraform modules?

Reusable, configurable collections of Terraform code.

30. What is drift and how do you fix it?

Drift occurs when infra changes outside Terraform. Detect via plan and fix by apply or importing resources.