# DevOps and Cloud Interview Questions and Answers

## Rollback a Failed Deployment in Jenkins

- Use the "Build with Parameters" option to trigger a previous successful build.  
- Alternatively, use Git tags or commit hashes to redeploy a known good version.  
- For pipelines, implement a rollback stage that redeploys the last stable artifact.

## Docker Image vs Container

- Image: A static snapshot of an application and its dependencies.  
- Container: A running instance of an image, isolated with its own filesystem, network, and process space.

## Troubleshooting CrashLoopBackOff

- Run `kubectl describe pod <pod-name>` and `kubectl logs <pod-name>` to inspect errors.  
- Common causes: misconfigured env vars, missing files, failed health checks.  
- Fix the root cause and redeploy.

## Purpose of .gitignore

- Specifies files/folders Git should ignore (e.g., logs, secrets, build artifacts).  
- Prevents accidental commits of sensitive or irrelevant files.

## Managing Secrets in Kubernetes

- Use Kubernetes Secrets (`kubectl create secret`).  
- For enhanced security: integrate with HashiCorp Vault, Sealed Secrets, or External Secrets Operator.

## IAM Role

- Identity and Access Management role in AWS defines permissions for services/users.  
- Use it to grant least privilege access to EC2, Lambda, or Kubernetes nodes.

## Terraform Apply

- Validates and executes the plan to create/update/delete infrastructure.  
- Prompts for approval unless `-auto-approve` is used.

## Custom Domain for Kubernetes Ingress

- Create an Ingress resource with the domain in the `host` field.  
- Update DNS records to point to the ingress controller’s external IP.  
- Use TLS with cert-manager for HTTPS.

## Load Balancer in Cloud

- Distributes traffic across multiple instances.  
- Types: Layer 4 (TCP) and Layer 7 (HTTP).  
- In cloud (e.g., AWS ELB), it auto-scales and integrates with health checks.

## Find Process Using a Port (Linux)

Use the following commands:  
 sudo lsof -i :<port>  
 sudo netstat -tulnp | grep <port>

## Pipeline Stage vs Step

- Stage: Logical grouping (e.g., Build, Test, Deploy).  
- Step: Individual task within a stage (e.g., run tests, build image).

## Monitor Logs from Multiple Containers

- Use centralized logging: ELK stack, Fluentd, Loki, or Datadog.  
- Agents collect logs and push to a central dashboard.

## Horizontal vs Vertical Pod Autoscaling

- Horizontal: Adds/removes pod replicas based on metrics.  
- Vertical: Adjusts CPU/memory of existing pods.

## Update Env Vars in Running Container

- You can’t directly update env vars in a running pod.  
- Update the deployment and redeploy:  
 env:  
 - name: VAR\_NAME  
 value: "new\_value"

## No Resource Limits in K8s

- Pods can consume unlimited resources, risking node starvation.  
- Always set `requests` and `limits` to ensure fair scheduling.

## Ensure Idempotency in Terraform

- Use state files and avoid hardcoded values.  
- Use `terraform plan` to preview changes.  
- Avoid using `count` or `for\_each` with changing keys.

## Dangling Docker Image

- Image not tagged and not referenced by any container.  
 docker images -f "dangling=true"

## Prevent Accidental Deletions

- Use RBAC, approval workflows, and Terraform `prevent\_destroy`.  
- Enable backup and audit logging.

## Readiness vs Liveness Probes

- Readiness: Is the app ready to serve traffic?  
- Liveness: Is the app alive or should it be restarted?

## Debugging a Failed CI Build

- Check logs for error messages.  
- Validate environment setup, dependencies, and secrets.  
- Reproduce locally if possible.  
- Use version control to compare with last successful build.

**DevOps and Cloud Engineering Q&A Summary**

**What is the difference between hard link and soft link?**

- Hard Link: Points directly to the inode of a file. Multiple hard links share the same inode.  
 - Cannot link directories.  
 - If the original file is deleted, the data remains accessible via the hard link.  
- Soft Link (Symbolic Link): Points to the file name/path.  
 - Can link directories.  
 - If the original file is deleted, the symlink becomes broken.

**How would you monitor resources in Linux?**

- Tools include: top, htop, vmstat, iostat, netstat, dstat  
- System monitoring suites: Nagios, Zabbix, Prometheus + Grafana, Glances  
- Logs: journalctl, /var/log/syslog, /var/log/messages

**What are the different types of S3?**

- S3 Standard  
- S3 Intelligent-Tiering  
- S3 Standard-IA (Infrequent Access)  
- S3 One Zone-IA  
- S3 Glacier  
- S3 Glacier Deep Archive

**What are the purchasing methods for Amazon EC2?**

- On-Demand: Pay per hour/second.  
- Reserved Instances: 1 or 3-year commitment.  
- Spot Instances: Bid for unused capacity.  
- Savings Plans: Flexible pricing model.  
- Dedicated Hosts: Physical servers for compliance.

**What are the different types of Auto Scaling?**

- Dynamic Scaling: Based on metrics (CPU, memory).  
- Scheduled Scaling: Based on time.  
- Predictive Scaling: Uses ML to forecast demand.  
- Manual Scaling: Admin sets desired capacity.

**What is the difference between on demand and spot instance?**

- On-Demand: Reliable, no interruption, higher cost.  
- Spot: Cheap, can be interrupted anytime, ideal for fault-tolerant workloads.

**What is the difference between application load balancer and network load balancer?**

Application Load Balancer (ALB):  
- Protocol: HTTP/HTTPS  
- Layer: Layer 7  
- Use Case: Web apps  
- Features: Path-based routing, host-based routing  
  
Network Load Balancer (NLB):  
- Protocol: TCP/UDP  
- Layer: Layer 4  
- Use Case: High-performance, low-latency  
- Features: Static IP, TLS passthrough

**What are the required pre-requisites for launching an ELB from Linux?**

- AWS CLI configured (aws configure)  
- Security groups and subnets set  
- EC2 instances running in target group  
- IAM permissions for ELB  
- Health check configuration

**What is the difference between add and copy in docker file?**

- COPY: Simple file copy.  
- ADD: More powerful—can extract archives and supports remote URLs.

**How would you remove all the unused images in docker?**

Command:  
docker image prune -a  
Or to be safe:  
docker system prune -a

**What is terraform tfstate file?**

- Stores the current state of infrastructure.  
- Used to track resource changes.  
- Should be stored securely (e.g., in S3 with state locking via DynamoDB).

**What is the use of terraform workspace?**

- Allows multiple state files in the same configuration.  
- Useful for managing environments (e.g., dev, staging, prod).

**What is the root directory of Jenkins in Linux O/S?**

- Default: /var/lib/jenkins  
- Configs: /etc/default/jenkins  
- Logs: /var/log/jenkins

**What all monitoring tools can we use instead of Jenkins?**

- Prometheus + Grafana  
- Datadog  
- New Relic  
- Zabbix  
- Nagios  
- ELK Stack (Elasticsearch, Logstash, Kibana)

**What is a parameterized job in Jenkins?**

- Allows passing parameters to a build (e.g., branch name, environment).  
- Types: String, Choice, Boolean, File, etc.  
- Enables dynamic and reusable pipelines.

DevOps, Terraform, Kubernetes, Jenkins, and AWS Interview Q&A

## 1. How do you manage Terraform state files when working in a team?

Use remote backends like AWS S3 with DynamoDB for state locking. Enable versioning on the S3 bucket. Use workspaces for environment separation (e.g., dev, staging, prod).

## 2. What approach do you take to avoid or resolve state file conflicts in Terraform?

Enable state locking via DynamoDB. Use `terraform plan` before applying changes. Communicate changes via PRs and peer reviews. If conflicts occur, manually inspect and merge state using `terraform state` commands.

## 3. Can you write a Terraform configuration to provision multiple S3 buckets?

variable "bucket\_names" {  
 type = list(string)  
 default = ["bucket-one", "bucket-two", "bucket-three"]  
}  
  
resource "aws\_s3\_bucket" "buckets" {  
 for\_each = toset(var.bucket\_names)  
 bucket = each.value  
 acl = "private"  
}

## 4. What kind of CI/CD pipelines have you built using Jenkins?

Multi-stage pipelines for Build → Test → Lint → Security Scan → Deploy. Docker image creation and push to ECR/GCR. Helm-based Kubernetes deployments. Integration with SonarQube, Nexus, Slack, and GitHub.

## 5. How would you integrate Jenkins with SonarQube for code quality checks?

Install SonarQube Scanner plugin. Configure SonarQube server in Jenkins global settings. Use pipeline steps like:  
  
withSonarQubeEnv('SonarQube') {  
 sh 'sonar-scanner -Dsonar.projectKey=my-app -Dsonar.sources=src'  
}

## 6. How does Jenkins authenticate with Docker registries when pushing images?

Use Docker credentials stored in Jenkins credentials store. Authenticate using:  
  
withCredentials([usernamePassword(credentialsId: 'docker-creds', usernameVariable: 'USER', passwordVariable: 'PASS')]) {  
 sh "docker login -u $USER -p $PASS registry.example.com"  
}

## 7. Have you worked with Kubernetes? What deployment strategy do you prefer and why?

Rolling updates for most cases. Blue-green or Canary for critical services to minimize downtime and risk.

## 8. How have you implemented blue-green deployment in a Kubernetes environment?

Deploy two versions (e.g., v1 and v2) with separate services. Use Ingress or service routing to switch traffic. Example: Update Ingress backend from blue to green.

## 9. What is HPA in Kubernetes, and when would you use it?

HPA (Horizontal Pod Autoscaler) automatically scales pods based on CPU/memory/custom metrics. Use when workloads are variable and need elastic scaling.

## 10. Suppose a new deployment introduces issues — how would you roll back to the previous stable version using Kubernetes?

Use: kubectl rollout undo deployment <deployment-name>. Or redeploy a previous version using Helm or GitOps.

## 11. What is the purpose of StatefulSets in Kubernetes, and how are they different from Deployments?

StatefulSets are for stateful apps needing stable network IDs, persistent storage, and ordered deployment. Deployments are for stateless apps with replica management and rolling updates.

## 12. What types of IAM policies exist in AWS? Can you explain their differences?

Managed Policies: AWS or customer-created reusable policies. Inline Policies: Embedded directly in a user/group/role. Permissions Boundaries: Limit max permissions a role/user can have. Service Control Policies (SCPs): Used in AWS Organizations.

## 13. How are S3 bucket policies different from ACLs?

Bucket Policies: JSON-based, fine-grained access control. ACLs: Legacy, less flexible, used for basic permissions.

## 14. What’s your understanding of dynamic auto scaling in AWS?

Uses CloudWatch metrics to scale EC2 instances, ECS tasks, or ASGs. Can be scheduled or event-driven. Works with Target Tracking, Step Scaling, or Simple Scaling.

## 15. Can you explain the key differences between AWS Security Groups and Network ACLs?

Security Groups: Stateful, instance-level firewalls. Network ACLs: Stateless, subnet-level firewalls. SGs are easier to manage; NACLs offer granular control.

## 16. Which AWS services have you worked with most, and how have you used them in your DevOps projects?

EC2: CI/CD runners, build agents. ECS/EKS: Container orchestration. S3: Artifact storage, Terraform state. CloudWatch: Monitoring and alerting. IAM: Role-based access control. CodePipeline/CodeBuild: Native CI/CD.

Q: How do you expose a Kubernetes application to the outside world?  
  
A: Use a Service of type LoadBalancer or configure an Ingress. LoadBalancers provide a direct public IP, while Ingress supports advanced routing, TLS termination, and centralized traffic management.  
  
Q: What causes pods to go into a CrashLoopBackOff state?  
A: Common reasons include incorrect image versions, missing environment variables, failing readiness/liveness probes, or insufficient resource limits. Checking logs with kubectl logs and describing the pod helps pinpoint the issue.  
  
Q: How do you scale an application in Kubernetes?  
A: Use kubectl scale to manually adjust replicas or define an HorizontalPodAutoscaler (HPA) for CPU/memory-based auto-scaling. This ensures apps can handle traffic spikes without downtime.  
  
Q: What’s the difference between a Deployment and a StatefulSet?  
A: Deployments are ideal for stateless workloads where replicas are identical. StatefulSets are used for stateful apps (like databases) requiring stable network IDs, persistent storage, and ordered scaling.  
  
Q: How do you secure sensitive data in Kubernetes?  
A: Store credentials, API keys, and certificates in Secrets rather than ConfigMaps. Mount them as environment variables or volumes, and use encryption at rest with RBAC for strict access control.  
  
Q: What’s the best way to monitor Kubernetes workloads?  
A: Integrate Prometheus for metrics collection, Grafana for visualization, and Alertmanager for proactive notifications. This trio gives end-to-end visibility into cluster health and app performance.  
  
Q: How do you perform a rolling update in Kubernetes?  
A: A Deployment supports rolling updates by gradually replacing old pods with new ones, ensuring zero downtime. You can control the pace using maxUnavailable and maxSurge parameters.  
  
Q: What’s the purpose of a DaemonSet?  
A: A DaemonSet ensures a copy of a pod runs on all (or selected) nodes in the cluster. It’s commonly used for logging agents, monitoring tools, and security daemons.  
  
Q: How do you handle persistent storage in Kubernetes?  
A: Use PersistentVolumes (PV) and PersistentVolumeClaims (PVC) to abstract storage from pods. This allows apps to survive restarts and rescheduling without losing data.