AWS DevOps Interview Questions (4 Years Experience)

Infrastructure as Code (IaC) Interview Questions with Detailed Answers

Q: What is Infrastructure as Code?

A: Infrastructure as Code (IaC) is the practice of provisioning and managing infrastructure using code instead of manual processes.

Benefits:

- Version control for infrastructure
- Reproducible environments
- Automation reduces human error

Real-time example: We used Terraform to create VPCs, subnets, and EC2 instances. Version control allowed us to track infra changes over time.

Q: CloudFormation vs Terraform differences and when to use?

A: CloudFormation is AWS-native, Terraform is open-source and cloud-agnostic.

Feature	CloudFormation	Terraform
Scope	AWS-only	Multi-cloud
Language	JSON/YAML	HCL (declarative)
Modularity	Limited	Strong (modules, reusability)
State management	Handled by AWS	Manual/remote backend needed

AWS DevOps Interview Questions (4 Years Experience)

Use Terraform when multi-cloud or reusable modules are needed. Use CloudFormation for tighter AWS integration.

Real-time example: We used CloudFormation for AWS-native workloads, and Terraform for creating shared resources across AWS and GCP.

Q: How do you manage state in Terraform?

A: Terraform uses a state file to track the infrastructure.

- Local state is default (`terraform.tfstate`)
- Remote state (e.g., S3) enables team collaboration

Best practices:
- Use S3 with versioning enabled
- Use DynamoDB for state locking

Real-time example: Our team used S3 for storing state and DynamoDB for locking to avoid simultaneous updates during CI/CD runs.

Q: How do you handle secrets in Terraform?

A: Avoid hardcoding secrets in Terraform code.

Options:

- Use AWS Secrets Manager or SSM Parameter Store
- Use environment variables or `.tfvars` (not checked into Git)
- Use Vault for advanced secret management

Real-time example: We used SSM Parameter Store with encrypted values. Terraform pulled parameters during provisioning using `data` blocks.

Q: How to implement change detection and approval in IaC?

A: Change detection in Terraform:

- Use `terraform plan` to preview changes
- Use CI/CD to generate and email plan output

Approval:

- Manual approval stages in CodePipeline or Jenkins before applying

Real-time example: Our CI pipeline generated a Terraform plan. A security team reviewed it and approved via manual gate in Jenkins before applying changes.

Q: Real-time workflow for provisioning VPC using Terraform?

A: 1. Define variables for CIDR, subnets, AZs

2. Create VPC, subnets, route tables in Terraform code

3. Use `terraform init`, `plan`, and `apply`

4. Store state in S3, lock with DynamoDB

Real-time example: We had a reusable Terraform module for VPC provisioning. It was used across multiple environments (dev, staging, prod) with different variables.