AWS DevOps Interview Questions (4 Years Experience)

Security & Compliance Interview Questions with Detailed Answers

Q: How do you secure access to AWS resources?

A: Use IAM (Identity and Access Management).

Best practices:

- Use IAM roles over users
- Enable MFA (Multi-Factor Authentication)
- Apply least privilege principle
- Use AWS SSO for centralized access

Real-time example: Developers used IAM roles to assume temporary access during CI jobs. MFA was required for console access, and policies were scoped per environment (dev, prod).

Q: How do you handle secrets securely in AWS?

A: Options:

- AWS Secrets Manager: Automatic rotation, encryption
- AWS SSM Parameter Store (with SecureString)

Best practices:

- Never hardcode in codebase
- Limit access via IAM policies

Real-time example: Our application pulled DB passwords from Secrets Manager during deployment, ensuring

they were not stored in Git or environment files.

Q: How can you enforce compliance across AWS accounts?

A: Use tools like:

- AWS Config: Tracks configuration changes
- AWS Organizations + SCPs: Enforce org-wide rules
- AWS Audit Manager and Security Hub: Continuous compliance checks

Real-time example: We enforced no public S3 buckets across accounts using SCPs and monitored compliance violations with AWS Config rules.

Q: What is SCP (Service Control Policy) and how is it used?

A: SCPs are policies attached to AWS Organizations to restrict permissions across member accounts.

Use cases:

- Deny use of specific services (e.g., EC2)
- Prevent disabling CloudTrail

Real-time example: We used SCPs to block root users from deleting CloudTrail logs and restrict IAM creation in non-production accounts.

Q: How do you audit activity in AWS accounts?

A: Use AWS CloudTrail.

Features:

- Records all API calls (who, what, when, where)
- Delivers logs to S3 or CloudWatch

Real-time example: We investigated unauthorized access using CloudTrail logs to trace IAM login attempts and S3 data access.

Q: How do you secure CI/CD pipelines?

A: Steps:

- Use roles with limited permissions for pipeline stages
- Store credentials in Secrets Manager or Parameter Store
- Enable artifact encryption
- Sign artifacts (CodeSign)

Real-time example: Our CodePipeline used isolated IAM roles per stage, and credentials were injected from SSM. Artifacts were encrypted and stored in versioned S3.