

Medium

Search

Write

Sign up

Sign in



DevSecOps &amp; AI

★ Member-only story

# DevOps Technical Interview Round 1 & 2 Accenture



Amit Singh

Follow

12 min read · Jul 3, 2025



26



These DevOps Questions were asked in Accenture Tech round 1 and 2 for 16LPA, and I have added 15 bonus interview questions after Accenture Q & A.

Must check out these Accenture Interview Questions, these questions are asked in many other interviews too for cloud and DevOps

## Q. How are your day-to-day activities as a DevOps Engineer?

### Pointers

- Monitoring infrastructure
- Writing/maintaining IaC
- Improving pipelines
- Handling releases
- Supporting dev teams
- Troubleshooting incidents

```
[ Daily ]
|
+--> Monitor clusters
+--> CI/CD maintenance
+--> Infra as Code
+--> Incident support
```

## Scenario style

“On a typical day, I monitor production clusters, manage and improve our Jenkins pipelines, work on Terraform modules for resource provisioning, and collaborate with developers for application releases. I also handle incidents like pod failures or networking issues.”

🧠 *Easy to remember:* **Monitor, Improve, Support, Troubleshoot**

## Q. What is a NAT Gateway?

### Pointers

- Allows private subnets to access the internet
- Without exposing resources to incoming traffic
- Managed by AWS



### ✓ Scenarios

“We had EC2 instances in private subnets that needed to pull Docker images from public repos. We placed a NAT Gateway in a public subnet, routing private subnet traffic through it so those instances could reach the internet safely.”

“In my last project, the private subnets for EC2 instances needed to pull OS updates from the internet. I used a NAT gateway in the public subnet, routing traffic through it, while blocking inbound access to those instances.”

### Diagram

Internet-> IGW → NAT Gateway → Private Subnet EC2

## Advanced Q&A

**Q:** Can a NAT Gateway receive inbound traffic?

**A:** No — it only handles outbound requests from private subnets.

### Best practices

- Always place a NAT gateway in a public subnet
- Remember, *private* subnet routes point to the NAT

## Q. Prerequisites to upgrade a Kubernetes cluster

### Pointers

- Backup etcd
- Drain nodes
- Check deprecated APIs
- Update kubectl
- Test in staging

### ✅ Scenario style

“We planned a Kubernetes upgrade from v1.24 to v1.27. First, we backed up etcd, verified compatibility of Ingress controllers, and checked deprecated APIs. Then we drained and upgraded worker nodes in a rolling manner.”

```
[ etcd snapshot ] --> [ kubectl upgrade plan ]
|
[ drain nodes ]
|
[ rolling node upgrade ]
```

🧠 *Easy to remember:* Backup, Drain, Check, Test

## Q. What is Pod Disruption Budget (PDB)?

### Pointers

- Defines minimum available pods during voluntary disruptions
- Prevents too many pods from going down during drain/updates

### ✅ Scenario style

1. "When upgrading a deployment, I set PDB to minAvailable: 2 to ensure at least 2 pods are always running so user traffic isn't impacted."

2. "We needed to upgrade the node pool for a production app. We set a PDB to minAvailable: 2 to keep at least 2 pods online even during the drain."

## PDB

+ → minAvailable: 2

+ → maxUnavailable: 1

## Advanced Q&A

Q: Does PDB protect from node failures?

A: No, PDB only controls voluntary disruptions (e.g., drain, upgrades).

## Best practices

- always set PDB for critical deployments
- test the effect by cordoning a node

🧠 *Easy to remember:* **Minimum running pods during changes**

## Q. Shell script for factorial of a number

```
#!/bin/bash
echo "Enter a number:"
read num
fact=1
for (( i=1; i<=num; i++ ))
do
    fact=$((fact * i))
done
echo "Factorial of $num is $fact"
```

## Q. Tell me about VPC structure in your project

### Pointers

- VPC CIDR
- Public & private subnets

- NAT gateway
- Internet gateway
- Route tables
- Security groups

### ✓ Scenario style

“We designed a VPC with /16 CIDR, split into multiple private subnets for application servers, with a NAT gateway for outbound internet, public subnets for the ALB, and separate security groups for database layers.”

“Our VPC has a /16 CIDR block, split into public subnets for the ALB and private subnets for EC2/EKS nodes. NAT gateways handle outbound traffic for private subnets. RDS is placed in private subnet.”

```
VPC
|
+--> Public Subnet (ALB, NAT GW)
|
+--> Private Subnet (EC2, EKS, RDS)
```

### Advanced Q&A

Q: Why put RDS in a private subnet?

A: For security — no direct internet access.

### Best practices

- separate public/private
- follow least privilege on security groups

🧠 *Easy to remember:* CIDR, Subnets, NAT, IGW, Routing

### Q. How is your CI/CD pipeline set up? What security tools are integrated?

#### Pointers

- Jenkins/GitLab
- Docker builds
- SonarQube (code scan)

- Trivy/Anchore (image scan)
- HashiCorp Vault (secrets)

*Code → Build → Scan (Trivy) → Test → Deploy to K8s*

### ✓ Scenario style

“Our pipeline is on GitLab CI, running Docker builds, security scanning with Trivy, static code analysis with SonarQube, and uses Vault to inject secrets. This ensures secure, consistent, automated releases.”

### Advanced Q&A

Q: How do you manage secrets?

A: Vault or SSM Parameter Store, never hard-coded.

### Best practices

- integrate image scanning
- automate secrets rotation

🧠 *Easy to remember: Build, Scan, Store secrets, Deploy*

## Q. How do you manage them?

### Pointers

- Version control (Git)
- IaC for infra
- Role-based access
- Automated tests

```
GitOps
|
Infra as Code
|
RBAC
```

### ✓ Scenario style

“We manage pipelines through version-controlled YAML, infrastructure with Terraform, and RBAC controls in Kubernetes to delegate least privilege.”

🧠 *Easy to remember: Version, Automate, Secure*

## Q. Write a rough pipeline script for microservices architecture

```
stages:
  - build
  - test
  - deploy

build:
  script:
    - docker build -t myapp:${CI_COMMIT_SHA} .
    - docker push myapp:${CI_COMMIT_SHA}

test:
  script:
    - docker run myapp:${CI_COMMIT_SHA} pytest

deploy:
  script:
    - kubectl apply -f deployment.yaml
```

### Advanced Q&A

Q: How do you handle dependencies?

A: Use semantic versioning + separate pipelines to avoid coupling.

### Best practices

- keep microservices decoupled tag images uniquely

🧠 *Remember: Build → Test → Deploy*

## Q. What is multi-stage Docker build?

### Pointers

- Separate build & run stages
- Reduce image size
- Improves security

✅ **Scenario style**

“In a microservice build, we use a Golang builder image, compile binaries, and then copy them to a scratch image in a second stage. That keeps the production image minimal.”

### Advanced Q&A

Q: Why use multi-stage?

A: Reduce attack surface and image size.

### Best practices

- keep only production dependencies
- scan final images

 *Easy to remember:* **Build once, copy artifacts, keep clean**

## Q. What are manifest files?

### Pointers

- YAML files for Kubernetes resources
- Define pods, deployments, services, etc.

### Scenario style

“We manage Kubernetes manifests for deployments and services in a GitOps workflow to apply them consistently across environments.”

“We store Deployment and Service YAMLs in Git repos. We apply them with kubectl or FluxCD.”


### Advanced Q&A

Q: How to manage multiple environments?

A: Use Kustomize or Helm.

### Best practices

- version manifests
- keep separate folders for dev/prod

 *Easy to remember:* **K8s blueprints**



## Q. What is Ansible Vault?

### Pointers

- Encrypt sensitive data (passwords, secrets)
- Stored in playbooks securely

[ ansible-playbook ] → [ encrypted vault file ]

### ✅ Scenario style

“We use Ansible Vault to encrypt DB passwords in our inventory, and decrypt only during runtime with a vault password file.”

### Advanced Q&A

Q: What if you lose the vault password?

A: You cannot decrypt — store vault password securely.

### Best practices

- rotate vault passwords
- never commit vault keys to Git

🧠 *Easy to remember:* Encrypted secrets for playbooks

## Q. How to make a K8s cluster highly available?

### Pointers

- Multiple control plane nodes
- Etcd cluster quorum
- Load balancer in front of control plane
- Spread worker nodes across AZs

### ✅ Scenario style

“We deployed 3 control plane nodes with an external HA load balancer and spread worker nodes in 3 AZs to achieve high availability.”

```
LB
|
+--> master1
+--> master2
+--> master3
```

### Advanced Q&A

Q: How do you handle etcd failure?

A: Ensure odd number of etcd members and frequent snapshots.

### Best practices

- HA LB in front of control planes
- spread AZs

 *Easy to remember:* **Multi-master + Load Balancer + Spread**

## Q. Monitoring tools & common pod errors

### Pointers

- Prometheus, Grafana
- Alertmanager
- Common errors: CrashLoopBackOff, ImagePullBackOff

### Scenario style

“We use Prometheus + Grafana for metrics and Alertmanager for notifications. The most common pod issue I handled was CrashLoopBackOff due to wrong configmaps or missing secrets.”

### Advanced Q&A

Q: What's CrashLoopBackOff?

A: Container keeps crashing repeatedly, often due to bad configs.

### Best practices

- set up alerts
- test alert receivers regularly

 *Easy to remember:* **Prometheus + Grafana + Alerts**

## Q. Terraform script for VPC architecture

```
provider "aws" {  
  region = "ap-south-1"  
}  
  
resource "aws_vpc" "my_vpc" {  
  cidr_block = "10.0.0.0/16"  
}  
  
resource "aws_subnet" "public_subnet" {  
  vpc_id            = aws_vpc.my_vpc.id  
  cidr_block        = "10.0.1.0/24"  
  availability_zone = "ap-south-1a"  
  map_public_ip_on_launch = true  
}  
  
resource "aws_internet_gateway" "igw" {  
  vpc_id = aws_vpc.my_vpc.id  
}
```

### Advanced Q&A

Q: How to handle state files?

A: Use remote backend with S3 + DynamoDB locking.

### Best practices

- use terraform fmt
- version lock your providers

🧠 *Remember:* VPC → Subnets → IGW

## Q. How many objects can an S3 bucket store?

- Unlimited objects
- Practically limited by storage quotas

“S3 scales virtually unlimited. One bucket can store billions of objects.”

### Advanced Q&A

Q: Any hard limits?

A: Only practical ones (like request rates), no hard object limit.

### Best practices

- enable bucket versioning

- use lifecycle rules

🧠 *Easy to remember:* **Unlimited**

## Q. What are IAM roles and policies?

### Pointers

- Roles: identities with permissions
- Policies: permission documents in JSON
- Used to grant granular access

### ✅ Scenario style

“I assigned an IAM role to an EC2 instance with a policy to only allow S3 access for backup storage.”

### Advanced Q&A

Q: Difference between policy and role?

A: Role = identity; Policy = permission rules.

### Best practices

- follow least privilege
- audit IAM regularly

🧠 *Easy to remember:* **Role = identity, Policy = rules**

## Q. What are artifacts?

```
Build  
|  
Artifact  
|  
Artifactory
```

### Pointers

- Build outputs
- Stored in artifact repositories like Artifactory or S3

### ✓ Scenario style

“Our CI pipeline pushes JAR artifacts to Artifactory after a successful Maven build.”

“Our Maven builds produce JARs stored in Artifactory as versioned artifacts.”

#### Advanced Q&A

Q: Why store them?

A: To enable rollback or re-deploys.

#### Best practices

- automate artifact cleanup
- version artifacts clearly

🧠 *Easy to remember:* **Build results**

### Q. SATS and DATS?

#### Common terms

- SATS = System Acceptance Testing
- DATS = Data Acceptance Testing

SATS → DATS

### ✓ Scenario style

“After deploying, we perform SATS to validate application behavior, and DATS to verify data correctness with staging data.”


#### Advanced Q&A

Q: Are they manual or automated?

A: Usually manual with automated test cases integrated.

#### Best practices

- always document acceptance criteria
- automate where possible

 *Easy to remember: System + Data Testing*

## Q. How do you find errors in pipelines?

### Pointers

- Logs
- CI/CD dashboard
- Alerts
- Test reports

### Scenario style

“Whenever the pipeline fails, I check Jenkins logs, review the failing stage, and correlate with Git commit changes.”

### Advanced Q&A

Q: How do you get notified?

A: Slack or email from pipeline notifications.

### Best practices

- break pipeline into small stages
- add retry steps

 *Easy to remember: Logs + Stage + Commit*

## Q. What are Ansible Roles?

### Pointers

- Reusable set of tasks
- Structured like modules
- Makes playbooks cleaner

### Scenario style

“We created roles for installing Nginx, managing users, and deploying apps, to keep our playbooks DRY and modular.”

## Advanced Q&A

Q: Benefits of roles?

A: Reusable, maintainable, clean structure.

### Best practices

- structure roles with defaults, tasks, handlers
- version control them

 *Easy to remember:* **Reusable Ansible blocks**

**Thanks Everyone!**

LinkedIn: [Amit Singh](#)

Medium: [Amit Singh — Medium](#)

## Bonus Questions 15 common DevOps questions, with:

### **1** What is a CI/CD pipeline and how would you design one for a growing team?

#### Pointers

- Automates build, test, and deployment
- Improves release speed & consistency
- Usually uses tools like Jenkins, GitLab, GitHub Actions

#### Scenario style

*“In a previous team, I set up a GitLab CI pipeline with build → security scan → test → deploy stages. As the team grew, I split pipelines by microservice, added branch protection, and introduced approvals to scale safely.”*

### Best practices

- Separate pipelines per service
- Add approvals & rollback
- Use infrastructure as code

🧠 *Easy to remember:* **Build, Test, Deploy, Secure, Scale**

## 2 How does Infrastructure as Code work and why is it important?

### Pointers

- Infra described in code (e.g., Terraform, CloudFormation)
- Enables repeatability, versioning, consistency
- Reduces manual errors

### Scenario style

*“We migrated manual AWS provisioning to Terraform modules, making it easier to spin up staging and prod environments with identical configurations.”*

### Best practices

- Use Git for IaC
- Validate changes with `plan` commands
- Modularize code

🧠 *Easy to remember:* **Code Infra → Consistent → Repeatable**

## 3 Why are containers often preferred over virtual machines?

### Pointers

- Lightweight
- Faster startup
- Better resource utilization
- Easier portability

### Scenario style



*"In our team, we moved apps from VMs to Docker containers, reducing our environment setup time from 15 minutes to 30 seconds."*

### Best practices

- Keep containers small
- Use base images with minimal footprint
- Scan regularly

🧠 *Easy to remember: Lightweight, Fast, Portable*

## 4 What do you do when a deployment fails in production?

### Pointers

- Triage quickly
- Rollback if needed
- Investigate root cause
- Add safeguards

### Scenario style

*"When a deployment failed due to a misconfigured environment variable, I rolled back using Helm, analyzed the issue, fixed the config, and then re-deployed after validating in staging."*

### Best practices

- Automate rollback
- Add health checks
- Communicate with stakeholders

🧠 *Easy to remember: Rollback, Root cause, Recover*

## 5 Difference between DevOps and Site Reliability Engineering (SRE)?

### Pointers

- DevOps: culture + practices for faster delivery
- SRE: reliability engineering with SLOs, SLIs, error budgets
- SRE uses software engineering to improve ops

### Scenario style

*"I see DevOps as an umbrella culture, while SRE formalizes reliability with error budgets and service-level objectives."*

### Best practices

- Learn SLIs/SLOs
- Measure error budgets
- Share learnings between teams

 Easy to remember: DevOps = Speed, SRE = Reliability

## 6 What is GitOps and how does it improve deployment processes?

### Pointers

- Declarative infrastructure in Git
- Git is the single source of truth
- Reconciled continuously by tools like ArgoCD

### Scenario style

*"We used ArgoCD to sync manifests from a Git repo to Kubernetes, reducing manual mistakes and ensuring consistency."*

### Best practices

- Always protect the Git repo
- Use PR reviews
- Automate sync

🧠 *Easy to remember: Git as the Truth*

## 7 How do you make sure your CI/CD pipeline is secure?

### Pointers

- Secrets management (Vault, SSM)
- Image scanning (Trivy, Clair)
- Least-privilege permissions
- Signing artifacts

### Scenario style

*"We integrated Trivy to scan Docker images, HashiCorp Vault for secrets, and enforced least-privilege permissions for the pipeline's IAM role."*

### Best practices

- Rotate secrets
- Automate scanning
- Log all pipeline steps

🧠 *Easy to remember: Scan, Secrets, Least Privilege*

## 8 What are blue-green and canary deployments? When to use each?

### Pointers

- Blue-green: swap entire traffic between old/new versions
- Canary: slowly shift % of traffic to new version

- Reduce downtime and risk

### Scenario style

*“For a risky feature, I chose a canary rollout shifting 10% traffic to v2 while monitoring errors. For stable routine upgrades, I prefer blue-green.”*

### Best practices

- Use automated traffic shifting
- Rollback quickly
- Monitor carefully

🧠 *Easy to remember: Blue-green = switch, Canary = gradual*

## 9 How does Kubernetes fit into modern DevOps workflows?

### Pointers

- Automates container orchestration
- Handles scaling, rolling updates, self-healing
- Integrates with CI/CD and GitOps

### Scenario style

*“We migrated from ECS to EKS for better portability, advanced networking, and standardized Kubernetes skills across the team.”*

### Best practices

- Use RBAC
- Monitor with Prometheus
- Automate manifests

🧠 *Easy to remember: Scale, Heal, Automate*

## 10 Difference between Terraform and Ansible?

### Pointers

- Terraform: declarative, infra provisioning
- Ansible: procedural, configuration management
- Often used together

### Scenario style

*"I use Terraform to provision EC2 and VPCs, then Ansible to configure packages and users on top."*

### Best practices

- Separate provisioning and config
- Store states securely
- Modularize

🧠 *Easy to remember: Terraform = infra, Ansible = config*

## 11 What metrics matter most in DevOps and why?

### Pointers

- Deployment frequency
- Lead time to change
- Mean time to recover
- Change failure rate

### Scenario style

*"We tracked DORA metrics to measure if our delivery was improving month over month."*

### Best practices

- Automate metric collection
- Visualize trends
- Align with business goals

🧠 *Easy to remember:* DORA 4 Metrics

## **1 2** How do you handle secrets and sensitive data in pipelines?

### Pointers

- Never hardcode
- Use Vault / SSM / Sealed Secrets
- Rotate regularly

### Scenario style

*"We stored DB creds in Vault, injected them at runtime, and rotated them quarterly."*

### Best practices

- Audit secret usage
- Least privilege
- Encrypt at rest & in transit

🧠 *Easy to remember:* Store, Inject, Rotate

## **1 3** What is Shift Left Security and how do you implement it?

### Pointers


- Security practices early in the SDLC
- Static code scanning, IaC scanning
- Developer awareness

## Scenario style

*"We added SAST tools in our PR process and trained devs on secure coding."*

## Best practices

- Educate dev teams
- Automate security checks
- Fix vulnerabilities early

 *Easy to remember: Security Early*

## 1 4 Tools for logging, monitoring, and alerting?

### Pointers


- Logging: ELK, Loki
- Monitoring: Prometheus, Datadog
- Alerting: Alertmanager, PagerDuty

## Scenario style

*"We used Prometheus + Grafana for metrics, Loki for logs, and Alertmanager to page on-call engineers."*

## Best practices

- Test alerts
- Centralize logs
- Avoid alert fatigue

 *Easy to remember: Logs, Metrics, Alerts*

## 1 5 What is your approach to rolling back a failed release?

## Pointers

- Automated rollback strategies
- Keep previous artifact versions
- Validate before redeploying

## Scenario style

*“When a new deployment caused API errors, I rolled back by redeploying the previous Helm chart and confirmed the fix with health checks.”*

## Best practices

- Test rollback in staging
- Automate rollbacks
- Keep version history

🧠 *Easy to remember: Plan, Rollback, Validate*

[Ansible](#)[Kubernetes Cluster](#)[Aws S3](#)[Devops Practice](#)[Cloud Services](#)

26



### Published in DevSecOps & AI

267 followers · Last published 9 hours ago

The place where DevSecOps & AI unite

[Follow](#)

### Written by Amit Singh

88 followers · 10 following

DevOps & Cloud Engineer | AWS | Azure | Terraform | Docker | Ansible | CI/CD  
Jenkins | Github Actions

[Follow](#)

No responses yet




Write a response



What are your thoughts?

More from Amit Singh and DevSecOps & AI


 In DevSecOps & AI by Amit Singh

Capstone DevSecOps with Jenkins, Kubernetes, and Secure...

Bringing DevSecOps and Cloud DevOps Together: Real-World Capstone Experience


May 31  8




 In DevSecOps & AI by Obafemi

9 Bash Scripts to Build Interactive Command-Line Tools with Menus...

build deployment menus, diagnostic dashboards, and safe automation triggers al...

 Jun 26  51



 In DevSecOps & AI by Obafemi

10 Shell Scripts to Automate DevSecOps Workflows

ready-to-use bash scripts to bridge gaps between DevOps and security


 Jun 17  62



 Amit Singh

Master AWS: Real-World Scenario-Based Interview Questions

Ace your next DevOps or Cloud interview with practical EC2 troubleshooting cases and...

 Jun 19  4



See all from Amit Singh

See all from DevSecOps & AI

Recommended from Medium

 Devops Diaries

DevOps Interview Questions

Are you preparing for a DevOps interview and feeling overwhelmed by the vast topics to...


 Jul 8  7  1 

 In AWS in Plain Eng... by Sandesh I DevOps | AWS...

AWS Laid Off 40% of Its DevOps Staff—What They’re Using Instea...

- By Sandesh (5 + Years of DevOps | CI/CD | AWS | k8 | DevSecOps)

 5d ago  427  136 

 Zudonu Osomudeya

50 AI Prompts to Automate Everything You Hate as a DevOps...

DevOps automation with AI

 Jul 5  177  2 

 In SysopsMicro by @pramodchandrayan

Securing EKS Kubernetes Cluster: The Definitive Guide for Engineer...


Amazon Elastic Kubernetes Service (EKS) makes it easy to run Kubernetes on AWS, bu...

 Jul 7  18 

 Nidhi Ashtikar

32 DevOps & Cloud Interview Questions with Answers

1. What is your role and responsibility in your current or previous project?

 Sanjay Kumar PhD

DevOps Interview Questions and Answers

1. What is the difference between a container and a virtual machine?

Jul 9  15



Mar 6  13  2



See more recommendations

[Help](#) [Status](#) [About](#) [Careers](#) [Press](#) [Blog](#) [Privacy](#) [Rules](#) [Terms](#) [Text to speech](#)