

DevOps Interview Questions and Answers

1. What are your daily responsibilities as a DevOps engineer?
2. Which DevOps tools are you proficient with?
3. Can you describe the CI/CD workflow in your project?
4. How do you handle the continuous delivery (CD) aspect in your projects?
5. What methods do you use to check for code vulnerabilities?
6. What AWS services are you proficient in?
7. How would you access data in an S3 bucket from Account A when your application is running on an EC2 instance in Account B?
8. How do you provide access to an S3 bucket, and what permissions need to be set on the bucket side?
9. How can Instance 2, with a static IP, communicate with Instance 1, which is in a private subnet and mapped to a multi-AZ load balancer?
10. For an EC2 instance in a private subnet, how can it verify and download required packages from the internet without using a NAT gateway or bastion host? Are there any other AWS services that can facilitate this?
11. What is the typical latency for a load balancer, and if you encounter high latency, what monitoring steps would you take?
12. If your application is hosted in S3 and users are in different geographic locations, how can you reduce latency?
13. Which services can be integrated with a CDN (Content Delivery Network)?
14. How do you dynamically retrieve VPC details from AWS to create an EC2 instance using IaC?
15. How do you manage unmanaged AWS resources in Terraform?
16. How do you pass arguments to a VPC while using the 'terraform import' command?
17. What are the prerequisites before importing a VPC in Terraform?
18. If an S3 bucket was created through Terraform but someone manually added a policy to it, how do you handle this situation?
19. How do you handle credentials for a PHP application accessing MySQL or any other secrets in Docker?
20. What is the command for running container logs?
21. Have you upgraded any Kubernetes clusters?
22. How do you deploy an application in a Kubernetes cluster?
23. How do you communicate with a Jenkins server and a Kubernetes cluster?
24. How do you generate Kubernetes cluster credentials?
25. Do you only update Docker images in Kubernetes, or do you also update replicas, storage levels, and CPU allocation?
26. What types of pipelines are there in Jenkins?
27. Can you define environment variables inside your Jenkins pipeline?
28. What is the role of artifacts in Jenkins, and why do we need to push them to Nexus instead of building and storing them locally?
29. If you're developing a Python-based application, how do you separate the packages needed for your local deployment to avoid interfering with globally installed packages?
30. How do you handle error handling in Python?

ANSWERS

1. What are your daily responsibilities as a DevOps engineer?

- Automate infrastructure provisioning and management.
- Build and maintain CI/CD pipelines.
- Troubleshoot and resolve infrastructure issues.
- Collaborate with development and operations teams.
- Monitor system performance and optimize resources.

2. Which DevOps tools are you proficient with?

- Ansible, Terraform, Docker, Kubernetes, Jenkins, Git, Prometheus, Grafana, AWS CLI, Python scripting.

3. Can you describe the CI/CD workflow in your project?

- Code is committed to a version control system (Git).
- CI server (Jenkins) builds, tests, and packages code.
- Artifacts are stored in a repository (Nexus).
- CD deploys artifacts to various environments (dev, staging, prod).

4. How do you handle the continuous delivery (CD) aspect in your projects?

- Use feature flags for controlled releases.
- Implement blue/green deployments.
- Leverage canary deployments for gradual rollouts.
- Employ infrastructure as code (IaC) for environment management.

5. What methods do you use to check for code vulnerabilities?

- Static code analysis (SonarQube).
- Dynamic application security testing (DAST).
- Container vulnerability scanning (Trivy).
- Penetration testing.

6. What AWS services are you proficient in?

- EC2, S3, RDS, Lambda, IAM, VPC, CloudFormation, Route 53, EBS, ELB, CloudWatch.

7. How would you access data in an S3 bucket from Account A when your application is running on an EC2 instance in Account B?

- Use IAM roles for cross-account access.
- Configure trust relationships between accounts.
- Grant necessary permissions to the role.

8. How do you provide access to an S3 bucket, and what permissions need to be set on the bucket side?

- Use IAM policies to control access.
- Grant specific permissions (read, write, list, etc.) based on requirements.
- Employ bucket policies for additional control.

9. How can Instance 2, with a static IP, communicate with Instance 1, which is in a private subnet and mapped to a multi-AZ load balancer?

- Use NAT Gateway or Bastion Host for outbound internet access from Instance 1.
- Configure security groups to allow traffic between instances.

10. For an EC2 instance in a private subnet, how can it verify and download required packages from the internet without using a NAT gateway or bastion host? Are there any other AWS services that can facilitate this?

- Use AWS Systems Manager (SSM) Agent for package management.
- Leverage Amazon Elastic Container Registry (ECR) for private package repositories.

11. What is the typical latency for a load balancer, and if you encounter high latency, what monitoring steps would you take?

- Load balancer latency varies, but typically low.
- Monitor with CloudWatch for increased error rates, request count, and latency.
- Analyze load balancer logs for detailed insights.

12. If your application is hosted in S3 and users are in different geographic locations, how can you reduce latency?

- Use Amazon CloudFront as a CDN.
- Configure edge locations closer to users.
- Optimize content delivery with caching.

13. Which services can be integrated with a CDN (Content Delivery Network)?

- S3, CloudFront, Route 53.

14. How do you dynamically retrieve VPC details from AWS to create an EC2 instance using IaC?

- Use AWS Data Sources in Terraform or CloudFormation.
- Query VPC information using the AWS CLI.

15. How do you manage unmanaged AWS resources in Terraform?

- Use the terraform import command to bring existing resources under management.
- Consider custom resources or third-party providers.

16. How do you pass arguments to a VPC while using the 'terraform import' command?

- **Indirectly through state modification:** While not recommended, it's technically possible to modify the Terraform state file to add arguments after import. However, this approach is prone to errors and should be avoided.

17. What are the prerequisites before importing a VPC in Terraform?

- **Terraform state file:** A valid state file is required to track imported resources.
- **Resource address:** The unique identifier of the VPC to be imported.
- **Matching configuration:** The Terraform configuration should align with the existing VPC's attributes.

18. If an S3 bucket was created through Terraform but someone manually added a policy to it, how do you handle this situation?

- **Use Terraform's lifecycle block:** Configure the lifecycle block to replace the existing policy with the desired one.
- **Consider custom resources:** For more complex scenarios, create a custom resource to manage the bucket policy.

19. How do you handle credentials for a PHP application accessing MySQL or any other secrets in Docker?

- **Use environment variables:** Store sensitive information as environment variables within the Docker container.
- **Leverage secret management tools:** Tools like AWS Secrets Manager or HashiCorp Vault can securely store and manage secrets.
- **Avoid hardcoding credentials:** Never commit credentials directly to code.

20. What is the command for running container logs?

- **docker logs <container_name_or_id>:** This command displays the logs of a running container.

21. Have you upgraded any Kubernetes clusters?

- **Yes, I have upgraded Kubernetes clusters.** The process typically involves careful planning, creating backups, updating control plane components, and then upgrading worker nodes.

22. How do you deploy an application in a Kubernetes cluster?

- **Create deployment manifests:** Define the desired state of the application using YAML or JSON files.
- **Apply manifests:** Use kubectl apply to create or update the deployment.
- **Monitor deployment status:** Use kubectl describe deployment or kubectl get pods to track the deployment process.

23. How do you communicate with a Jenkins server and a Kubernetes cluster?

- **Jenkins plugins:** Use plugins to integrate with Kubernetes (e.g., Kubernetes Pipeline plugin).
- **Kubernetes API:** Interact with the Kubernetes API directly using the kubectl command or libraries.

24. How do you generate Kubernetes cluster credentials?

- **Use kubeconfig file:** Create a kubeconfig file containing authentication and authorization information.
- **Leverage IAM roles for service accounts:** Assign IAM roles to service accounts for secure access.

25. Do you only update Docker images in Kubernetes, or do you also update replicas, storage levels, and CPU allocation?

- **Update as needed:** Docker images are typically updated for code changes, while replicas, storage, and CPU allocation are adjusted based on workload requirements.

26. What types of pipelines are there in Jenkins?

- **Freestyle pipeline:** For simple jobs.
- **Pipeline as code:** Defines pipelines using Groovy or Jenkinsfile.
- **Multibranch pipeline:** Automatically creates pipelines for different branches.

27. Can you define environment variables inside your Jenkins pipeline?

- **Yes:** Environment variables can be defined within the Jenkinsfile or passed as parameters to the pipeline.

28. What is the role of artifacts in Jenkins, and why do we need to push them to Nexus instead of building and storing them locally?

- **Artifacts are build outputs:** They can be packages, test results, or other files.
- **Nexus provides centralized artifact management:** It offers features like versioning, search, and security, improving artifact management and distribution.

29. If you're developing a Python-based application, how do you separate the packages needed for your local deployment to avoid interfering with globally installed packages?

- **Create virtual environments:** Use tools like venv or virtualenv to isolate project dependencies.

30. How do you handle error handling in Python?

- **Use try-except blocks:** Enclose code that might raise exceptions in a try block and handle them in an except block.
- **Raise custom exceptions:** Create custom exception classes for specific error conditions.
- **Utilize logging:** Log errors for debugging and monitoring purpose

31. Tools for Dynamic Code Analysis in DevOps

Dynamic code analysis focuses on testing the application while it's running, unlike static analysis which examines the code without execution.

Here are some prominent tools for dynamic code analysis that can be integrated into a DevOps CI/CD pipeline:

Dynamic Application Security Testing (DAST) Tools

- **AppScan:** Offers comprehensive vulnerability scanning, including web application and API testing.
- **Burp Suite:** Popular for penetration testing and web application security, it can be integrated into CI/CD for automated vulnerability scanning.
- **OWASP ZAP:** Open-source tool for web application security testing, suitable for integration into CI/CD pipelines.
- **Checkmarx:** Provides both static and dynamic analysis, offering a holistic approach to security testing.
- **Veracode:** Another comprehensive platform that includes DAST capabilities.

Interactive Application Security Testing (IAST) Tools

- **Contrast Security:** Offers runtime application self-protection (RASP) and IAST capabilities.
- **Synopsys Seeker:** Provides IAST and runtime application protection.

Other Dynamic Analysis Tools

- **Selenium:** While primarily a test automation tool, it can be used for dynamic testing scenarios.
- **JMeter:** For performance testing, which can indirectly identify potential vulnerabilities.
- **Chaos Engineering Tools:** Such as Chaos Monkey, can be used to test system resilience under unexpected conditions.