Top 100 DevOps Engineer Interview Questions & Answers for 2025

1. DevOps Fundamentals

1. **What is DevOps?**
   DevOps is a software development methodology that integrates development (Dev) and operations (Ops) to improve collaboration, automation, and efficiency.
2. **What are the key principles of DevOps?**
   - Collaboration
   - Automation
   - Continuous Integration & Deployment
   - Monitoring & Feedback
   - Security & Compliance
3. **How does DevOps differ from Agile?**
   Agile focuses on software development processes, while DevOps extends it to operations, ensuring faster and reliable delivery.
4. **What are the key benefits of DevOps?**
   - Faster software releases
   - Improved collaboration
   - Higher efficiency and scalability
   - Better system reliability

5. **What are the key DevOps tools?**
    - CI/CD: Jenkins, GitLab CI, GitHub Actions
    - Containerization: Docker, Podman
    - Orchestration: Kubernetes, OpenShift
    - Monitoring: Prometheus, Grafana
    - Configuration Management: Ansible, Puppet, Chef
    - Version Control: Git

2. Version Control & Git

6. **What is Git?**
   Git is a distributed version control system for tracking source code changes.
7. **Difference between Git and GitHub/GitLab?**
   Git is a VCS, whereas GitHub/GitLab are web-based platforms providing repositories with collaboration features.
8. **What is the difference between git pull and git fetch?**
    - git fetch: Downloads changes but doesn't merge.
    - git pull: Fetches and merges changes into the working branch.
9. **Explain Git branching strategies.**
    - **Feature Branching**: Develop features in isolated branches.
    - **Gitflow**: Uses main, develop, feature, release, and hotfix branches.
    - **Trunk-based development**: Continuous integration into main.
10. **How do you resolve a merge conflict in Git?**
    - Identify conflicts using git status.
    - Manually edit the conflicting files.
    - Add (git add) and commit (git commit -m "Resolved conflict") the resolved files.

3. CI/CD (Jenkins, GitLab CI/CD, GitHub Actions)

11. **What is Continuous Integration (CI)?**
    CI automates code integration from multiple developers into a shared repository.
12. **What is Continuous Deployment (CD)?**
    CD automates software delivery from testing to production.
13. **Explain Jenkins Pipeline.**
    - A **Declarative Pipeline** defines the entire CI/CD process in Jenkinsfile.

- A **Scripted Pipeline** provides greater flexibility but requires Groovy scripting.

14. **How do you secure Jenkins?**
    - Use Role-Based Access Control (RBAC).
    - Encrypt secrets using Jenkins credentials.
    - Use HTTPS and limit plugin vulnerabilities.
15. **What are GitHub Actions and their advantages?**
    - A CI/CD automation tool integrated with GitHub.
    - Advantages: Easy setup, built-in marketplace, YAML-based workflows.

4. Configuration Management (Ansible, Puppet, Chef)

16. **What is Configuration Management in DevOps?**
    Managing system configurations to ensure consistency and scalability.
17. **How does Ansible differ from Puppet and Chef?**
    - **Ansible**: Agentless, YAML-based, push model.
    - **Puppet**: Agent-based, uses Puppet DSL, pull model.
    - **Chef**: Uses Ruby DSL, agent-based.
18. **What is an Ansible Playbook?**
    A YAML file defining automation tasks.
19. **Explain Infrastructure as Code (IaC).**
    Automating infrastructure provisioning using code (e.g., Terraform, Ansible).
20. **What is an Ansible role?**
    A structured way to organize Ansible Playbooks.

5. Containers & Orchestration (Docker, Kubernetes)

21. **What is Docker?**
    Docker is a containerization platform that packages applications with dependencies.
22. **What is a Dockerfile?**
    A script containing instructions to build a Docker image.
23. **What is Kubernetes?**
    An orchestration platform for managing containerized applications.
24. **What are Kubernetes Pods?**
    The smallest deployable unit in Kubernetes, containing one or more containers.
25. **What is Helm in Kubernetes?**
    A package manager for Kubernetes that simplifies application deployment.

6. Cloud & Infrastructure as Code (AWS, Azure, GCP, Terraform)

26. **What is Terraform?**
    An open-source IaC tool for managing cloud infrastructure declaratively.
27. **What is the difference between Terraform and CloudFormation?**
    - Terraform: Multi-cloud, state management, declarative.
    - CloudFormation: AWS-specific, integrated with AWS services.
28. **Explain the Terraform state file.**
    Stores infrastructure state to track changes.
29. **What is an AWS IAM role?**
    A set of permissions for AWS services to access resources securely.
30. **What is auto-scaling in AWS?**
    Automatically adjusting the number of instances based on demand.

7. Monitoring & Logging (Prometheus, Grafana, ELK Stack)

31. **What is Prometheus?**
    An open-source monitoring system for collecting and querying time-series data.
32. **What is Grafana used for?**
    A visualization tool for monitoring metrics.
33. **Explain the ELK Stack.**
    - **Elasticsearch**: Search engine
    - **Logstash**: Log ingestion
    - **Kibana**: Visualization
34. **What is observability?**
    The ability to measure system health via logs, metrics, and traces.
35. **What are service-level objectives (SLOs)?**
    Targets for system performance and availability.

8. Security in DevOps (DevSecOps)

36. **What is DevSecOps?**
    Integrating security into DevOps workflows.
37. **What is OWASP?**
    Open Web Application Security Project – provides security guidelines.
38. **How do you secure a containerized environment?**
    - Use minimal base images.

- Implement RBAC in Kubernetes.
- Scan images for vulnerabilities.

39. **What is Shift-Left Security?**
    Incorporating security early in the software development lifecycle.
40. **What are secrets management tools?**
    HashiCorp Vault, AWS Secrets Manager, Kubernetes Secrets.

9. SRE & Reliability Engineering

41. **What is Site Reliability Engineering (SRE)?**
    A discipline that applies software engineering to infrastructure operations.
42. **What are SLAs, SLOs, and SLIs?**
    - **SLA**: Service Level Agreement
    - **SLO**: Service Level Objective
    - **SLI**: Service Level Indicator
43. **What is Chaos Engineering?**
    The practice of testing system resilience through controlled failures.
44. **What is an Error Budget?**
    The acceptable downtime limit before affecting SLOs.
45. **What are blameless postmortems?**
    Incident reviews focused on learning rather than blaming.

10. Advanced CI/CD Concepts

46. **How do you implement Blue-Green Deployment?**
    - Maintain two environments (Blue = live, Green = new).
    - Switch traffic after testing Green.
47. **What is Canary Deployment?**
    - Gradual release to a small subset of users before full rollout.
48. **What is a Rolling Update?**
    - Gradual replacement of old instances with new ones without downtime.
49. **How do you handle secrets in CI/CD pipelines?**
    - Use environment variables, HashiCorp Vault, or AWS Secrets Manager.
50. **How do you prevent deployment failures?**
    - Implement automated testing, rollback strategies, and feature flags.

11. GitOps & Infrastructure Automation

51. **What is GitOps?**
    - A DevOps model where infrastructure changes are managed via Git repositories.
52. **How does GitOps differ from traditional IaC?**
    - GitOps enforces version-controlled infrastructure and automatic reconciliation.
53. **What are the best practices for Terraform state management?**
    - Store state in remote backends (S3, Azure Blob).
    - Use state locking to prevent conflicts.
54. **What is a Terraform module?**
    - A reusable, parameterized collection of Terraform configurations.
55. **What is drift detection in Terraform?**
    - Detecting changes in infrastructure that are not in the Terraform state.

12. Kubernetes Advanced Topics

56. **What is a Kubernetes DaemonSet?**
    - Ensures a pod runs on every node in a cluster.
57. **What is a Kubernetes StatefulSet?**
    - Used for stateful applications, providing stable network IDs and persistent storage.
58. **How does Kubernetes Horizontal Pod Autoscaler (HPA) work?**
    - Adjusts the number of pods based on CPU/memory metrics.
59. **What is a Kubernetes Ingress?**
    - A resource managing external access to services via HTTP/HTTPS.
60. **What is Kubernetes RBAC?**
    - Role-Based Access Control for managing permissions in a cluster.

13. Monitoring & Logging Advanced Concepts

61. **What is PromQL in Prometheus?**
    - A query language for fetching Prometheus metrics.
62. **How do you monitor Kubernetes clusters?**
    - Use Prometheus, Grafana, and Kubernetes Metrics Server.
63. **How do you centralize logs in a distributed system?**

- Use the ELK Stack or Fluentd for log aggregation.

64. **What is the difference between tracing and logging?**
    - Logging captures discrete events; tracing follows a request's lifecycle.
65. **How does OpenTelemetry help with observability?**
    - Provides unified telemetry (logs, metrics, traces) across services.

14. Security Best Practices in DevOps

66. **How do you implement DevSecOps in a pipeline?**
    - Integrate security scanning tools (SAST, DAST) into CI/CD.
67. **What is a SAST tool?**
    - Static Application Security Testing (e.g., SonarQube, Snyk).
68. **What is a DAST tool?**
    - Dynamic Application Security Testing (e.g., OWASP ZAP, Burp Suite).
69. **How do you implement least privilege access in DevOps?**
    - Use IAM roles, RBAC, and enforce MFA.
70. **What is container image scanning?**
    - Scanning Docker images for vulnerabilities using tools like Trivy or Clair.

15. Cloud-Native & Serverless

71. **What is a serverless architecture?**
    - Running applications without managing infrastructure (e.g., AWS Lambda).
72. **What are the benefits of serverless computing?**
    - Auto-scaling, cost efficiency, and reduced operational overhead.
73. **How does Kubernetes compare to AWS Lambda?**
    - Kubernetes runs containerized apps, whereas Lambda is event-driven and serverless.
74. **What is a CloudFormation Stack?**
    - A collection of AWS resources managed as a single unit.
75. **What is an API Gateway in cloud environments?**
    - A managed service for routing, securing, and monitoring API requests.

16. Advanced Networking & Security

76. **What is a Service Mesh?**

- A dedicated infrastructure layer for managing service-to-service communication (e.g., Istio, Linkerd).

77. **How do you secure microservices communication?**
    - Use TLS encryption, API gateways, and mutual TLS authentication.
78. **What is a Zero Trust security model?**
    - A model where no one is trusted by default, requiring strict identity verification.
79. **What is a WAF (Web Application Firewall)?**
    - Protects applications from web-based threats like SQL injection and XSS.
80. **How do you protect against DDoS attacks?**
    - Use CDNs, rate limiting, and AWS Shield/Cloudflare protections.

17. DevOps Culture & Processes

81. **How do you implement DevOps in a large enterprise?**
    - Start with CI/CD adoption, IaC, monitoring, and DevSecOps practices.
82. **What are key DevOps KPIs?**
    - Deployment frequency, mean time to recover (MTTR), change failure rate.
83. **How do you handle failures in a DevOps environment?**
    - Implement rollback strategies, blameless postmortems, and chaos engineering.
84. **What is a postmortem in DevOps?**
    - A retrospective analysis of an incident to prevent recurrence.
85. **What are feature flags?**
    - A mechanism for toggling features on/off in production.

18. Performance Optimization

86. **What is a CDN, and why is it used?**
    - A Content Delivery Network caches content to reduce latency.
87. **How do you optimize CI/CD pipeline performance?**
    - Use parallel builds, caching, and selective testing.
88. **How do you optimize database performance in DevOps?**
    - Indexing, caching, and database partitioning.
89. **What is a sidecar pattern in microservices?**

- Deploying an auxiliary container alongside the main service for logging, monitoring, or security.

90. **How do you reduce cloud costs in a DevOps environment?**
    - Use auto-scaling, spot instances, and cost monitoring tools.

19. Incident Management & Disaster Recovery

91. **What is a runbook in DevOps?**
    - A predefined set of procedures for handling incidents.
92. **What is a playbook in incident response?**
    - A detailed action plan for mitigating security or system issues.
93. **How do you handle rollback in Kubernetes?**
    - Use kubectl rollout undo to revert to the previous deployment.
94. **What is RTO and RPO in disaster recovery?**
    - RTO: Recovery Time Objective (time to restore services).
    - RPO: Recovery Point Objective (maximum acceptable data loss).
95. **How do you test disaster recovery in DevOps?**
    - Conduct failover tests and simulate outages.

20. Miscellaneous & Future Trends

96. **What is FinOps in cloud computing?**
    - Financial operations to optimize cloud spending.
97. **What is Chaos Engineering?**
    - Deliberate testing of system failures to improve resilience.
98. **What is Policy-as-Code?**
    - Defining security and compliance policies in code (e.g., OPA, AWS SCPs).
99. **What is AIOps?**
    - AI-driven operations that automate incident detection and resolution.
100. **What are emerging trends in DevOps for 2025?**

- AI-driven automation, GitOps adoption, enhanced Kubernetes security, and observability improvements.