# DevOps Interview questions

DevOps Overview

1. What is DevOps?
2. What is Benefits of DevOps?
3. What is Agile?
4. Difference between Agile & DevOps?
5. What is Shift-Left in DevOps?
6. What is DevOps Life Cycle?
7. What is Blue-Green deployment?
8. What is SDLC?

Aws Fundamentals

1. What is AWS?
2. What is Region, Availability Zone and Edge Location and its Benefits?
3. Difference between Region, Availability Zone and Edge Location?

DevOps on Cloud (AWS)

1. What are Services of AWS?
2. What Service are provided by AWS?
3. What are top 5 Services?
4. What is name of DevOps Tools is AWS?
5. What types of Data Storage Aws provide?
6. Which type of Data Storage is S3?
7. Which type of Data Storage is EBS?
8. Create aEc2 by using Terraform?
9. What is VPC and What and where it is Used?
10. What is Subnet and Why and Where it is Used?
11. What is IP And What is IP Address and What it is used?
12. What is most used Cloud Platform?
13. What are the resource creation tool/cli/program in cloud and full form?
14. What is most common used resource creation tool in cloud platforms?
15. What is pipeline creation tool like Jenkins used in AWS?
16. What is cloud?
17. What is type of clouds?
18. Difference between Private, Public and Hybrid cloud and what is used?
19. What type of EC2 instances and Which type instance family and its reason?
20. What Option we get on EC2 instances?
21. Where we stored Data in cloud?
22. What is VPC and What are the types of VPC?
23. Difference between Default VPC and Custom VPC?
24. What is VPC Pairing?
25. What is IAM?
26. What is IAM Policies?
27. What Is IAM Roles and what is importance?
28. What is the common security measure used in AWS?
29. What is Monolithic and Microservices?
30. Difference between Monolithic and Microservices?
31. What is load-Balancer and Types of Load-Balancers?
32. What is Identity Federation and Describe it uses?
33. Difference between Application Load-Balancer and Networking Load-Balancers?
34. What is list of key-value pairs that are commonly used in **pom.xml** files for projects deployed on AWS?
35. How can we be setting up a cloud-based repository (repo) on AWS or Repo cloud in AWS?
36. What Algorithms used in load-balancing?

Linux Fundamentals

1. What is Linux?
2. What is versions in Linux?
3. What is Directory in Linux and Describe it uses in Details?
4. What is Client-Server Model?
5. What is root Directory and What its work?
6. What id Bin Directory and What its work?
7. Name some commonly used Command in Linux?
8. What is kernel and what its work?
9. What is Shell Scripting and what its work?
10. What is Sudo Su?
11. What is Linux File permission and their concept ?

Application Development Fundamentals

1. What is Application Development and What are type of Application Developments?
2. What is Browser?
3. What is DNS?
4. What is TCP?
5. What is IP?
6. What is application development, and why is it important in the software industry?
7. Describe the software development life cycle (SDLC) and its phases.
8. What are the key differences between frontend and backend development?
9. Explain the concept of version control and its importance in application development.
10. What are the advantages of using a version control system like Git?
11. Explain the difference between waterfall and agile software development methodologies.
12. Describe the concept of DevOps and its impact on application development.
13. What is continuous integration (CI), and how does it improve the development process?
14. Explain the importance of testing in application development.
15. What are the different types of testing, and when would you use each?
16. Describe the difference between unit testing, integration testing, and end-to-end testing.
17. What is the role of a build automation tool (e.g., Maven, Gradle) in application development?
18. Explain the concept of continuous deployment (CD) and its benefits.
19. Describe the role of containers and container orchestration tools (e.g., Docker, Kubernetes) in modern application development.
20. What is microservices architecture, and how does it differ from monolithic architecture?
21. Explain the importance of scalability and performance optimization in application development.
22. What are some best practices for writing clean, maintainable, and efficient code?

Java Concepts

1. What is Java and Where its used?
2. What is concept of OOPS in java?
3. What is Operators in java and Types of operator in java?
4. Can we Used java dynamic?
5. What are the Features of Java?
6. What is JDK, JRE & JVM and Its Differences?

Understanding and Using Build Tools

1. What is a build tool, and why is it important in software development?
2. Can you name some popular build tools used in Java development?
3. Differentiate between compilation and building in the context of software development.
4. Explain the role of a build lifecycle in Apache Maven.
5. What are the main components of a Maven project?
6. Describe the purpose of a POM (Project Object Model) in Maven.
7. How do you define dependencies in a Maven project?
8. What is the purpose of a build configuration file, such as ‘pom.xml’ in Maven or ‘build.gradle’ in Gradle?
9. Compare and contrast Maven and Gradle as build tools.
10. Explain the concept of a plugin in the context of build tools.
11. How do you execute a Maven build from the command line?
12. What is a Maven repository, and why is it important in dependency management?
13. Describe the concept of transitive dependencies in Maven.
14. How do you specify the version of a dependency in a Maven project?
15. What is a snapshot version in Maven, and when would you use it?
16. Explain the purpose of Maven profiles and when you might use them.
17. What is the purpose of a build script in Gradle, and how is it different from Maven's approach?
18. How do you define tasks and dependencies in a Gradle build script?
19. Describe the Gradle wrapper and its benefits.
20. What are some best practices for managing dependencies and configuring builds in Maven and Gradle projects?
21. Differentiate between compile-time and runtime dependencies in a software project. How do build tools manage these dependencies?
22. Difference between Local Repositories, Central Repositories and Maven Repositories?

Continuous testing with Selenium

1. What is continuous testing, and why is it important in software development?
2. Explain the role of Selenium in continuous testing.
3. What are the advantages of using Selenium for automated testing in a continuous integration/continuous delivery (CI/CD) pipeline?
4. How does Selenium WebDriver facilitate browser automation for testing web applications?
5. What programming languages are supported by Selenium WebDriver?
6. Describe the difference between Selenium IDE, Selenium RC (Remote Control), and Selenium WebDriver.
7. How do you handle dynamic elements on a web page using Selenium WebDriver?
8. Explain the concept of Page Object Model (POM) and how it helps in writing maintainable Selenium tests.
9. What are implicit and explicit waits in Selenium WebDriver, and when would you use each?
10. How do you handle pop-up windows and alerts in Selenium tests?
11. What is headless testing, and how can it be useful in a continuous testing environment?
12. Explain how you would integrate Selenium tests into a continuous integration (CI) pipeline.
13. What are some common challenges you might face when running Selenium tests in parallel, and how can you address them?
14. What reporting tools or frameworks can you use to generate test reports for Selenium tests?
15. How would you handle cross-browser testing with Selenium WebDriver?
16. Describe the process of setting up and configuring a Selenium Grid for distributed testing.
17. How do you handle authentication (e.g., login forms) in Selenium tests?
18. What are some best practices for writing maintainable and reliable Selenium tests?
19. Explain the concept of data-driven testing with Selenium and how you would implement it.
20. What are some limitations or drawbacks of using Selenium for automated testing, and how would you mitigate them?
21. What is the testing and what is types of testing?
22. Difference between Junit testing and Test-NG report testing?
23. What is regression testing?

Overview of Python

1. Tell us Something about Python?
2. What type language is Python and What are its Feature and What its used for?
3. What python is called Platform Independent?
4. What are operator types?
5. What is function in python?
6. What are variables in python?
7. What are Data types in Python?
8. What is Mutable and Immutable in Python?
9. What is Array and what are it Types of arrays in Python?
10. Difference between List and Tuple?

Structure Query Language (SQL)

1. What is SQL?
2. What is the type of SQL?
3. What types of Database SQL support?
4. What is Scaling and What types of Scaling?
5. Difference between Horizontal Scaling and Vertical Scaling?
6. Difference between Alter and Update
7. What are join and what are types of joins?
8. What type of Data types Non-Relational Database stores?
9. What is Schema?
10. What are Attributes?
11. What is Tables?
12. Write a query for CREATE, DROP, ALETR, TRUNCATE, INSERT, DELETE, UPDATE, GRANT, REVOKE, SELECT, COMMIT, ROLLBACK, SAVE POINT?

Managing Source Code – Git and GitHub

1. What Is GIT?
2. What is GITHUB?
3. Difference between GIT and GITHUB?
4. Difference between GIT and SVN?
5. What is .gitignore and what its purpose?
6. What is Git Reset, Git Revert, Git Merge, Git Pull, Git Push, Git Fetch, Git Commit, Git Rebase, Git Branch, Git Cherry-pick, Git log, etc.?
7. Difference Between Git pull and Git push
8. Difference Between Git Reset & Git Revert?
9. Difference Between Git Pull & Git Fetch?
10. Difference Between Git Fork and Git Clone?
11. What is Tags and What are their purpose?
12. What is Distributed and Centralized repository and What are their differences?
13. What is Branching in Git and How do you create a new branch?
14. What is a merge conflict? How do you resolve it?
15. Explain the difference between git revert and git rebase?
16. What is a Git fork, and how is it different from a clone?
17. Explain the difference between git cherry-pick and git rebase?
18. Explain the difference between git merge and git rebase?
19. What is Git Stash, and how do you use it?
20. How does Git integrate with CI/CD pipelines?
21. How do you manage secrets in Git Repository?
22. How do you handle versioning in a Git project?
23. How does Git integrate with CI/CD pipelines?

Continuous Integration Using Jenkins

1. What is Master-slave model in Jenkins?
2. Difference between virtualization and containerization?

Containerization, Docker and Docker Hub

1. What is Docker?
2. What are the components of Docker Architecture?
3. On what platforms does Docker run?
4. What is the Lifecycle of Docker container?
5. What is the purpose of Docker\_Host?
6. What is docker in AWS?
7. Explain Docker Architecture?
8. Difference between virtualization and containerization?
9. Difference between a Container and a Virtual Machine (VM).
10. How does Docker work?
11. What is a Docker Image?
12. What is a Docker container?
13. Explain the role of Docker Engine?
14. Is Docker a Microservice?
15. What is the command to pull a Docker image from Docker Hub?
16. How to run a Docker container?
17. Explain the purpose of docker ps command
18. How to stop a running Docker container?
19. What is the difference between docker rm and docker rmi?
20. What is a Dockerfile and its use?
21. Explain the significance of the ENTRYPOINT and CMD instructions in a   
    Dockerfile
22. How to build a Docker image from a Dockerfile?
23. What is the purpose of the .dockerignore file?
24. What is Networking in Docker?
25. Explain the default network mode in Docker
26. How to expose ports in a Docker container?
27. What is Docker Compose, and why would you use it?
28. How to create Docker container?
29. How to stop and restart the Docker container?
30. Why are Docker volumes used?
31. How to create a Docker volume?
32. What are the networks that are available by default?
33. Explain the difference between a bind mount and a Docker volume
34. What is Docker Swarm and How to initialize a Docker Swarm?
35. Explain the role of managers and workers in Docker Swarm?
36. How can you improve Docker container security?
37. How to define services in a Docker Compose file?
38. Explain the use of the docker-compose up command.
39. How to scale services in Docker Compose?
40. What is a Docker Registry?
41. How to push a Docker image to Docker Hub?
42. How to check container logs in Docker?
43. Explain the purpose of the docker stats command.
44. What to do if a container exits immediately after starting?
45. How to remove all stopped containers?
46. How to manage secrets in Docker Swarm?
47. How does Docker fit into a CI/CD pipeline?
48. Explain the concept of blue-green deployment with Docker
49. How is Docker used in cloud platforms like AWS, Azure, or GCP?
50. What is Docker Swarm Mode, and how does it differ from standalone   
    Docker Swarm?
51. How to secure Docker containers and images?
52. What are some challenges when working with Docker, and how can they   
    be mitigated?
53. Can json be used instead of yaml for compose file?
54. How to use JSON instead of YAML compose file?
55. Explain the process of scaling your Docker containers
56. Explain Docker object labels.
57. Write a Docker file to create and copy a directory and built it using python modules?
58. What are the three main types of Docker components?
59. What is client?
60. What is ‘NameSpaces’ used for?
61. Mention some commonly used Docker Commands?
62. How do I run multiple copies of a Compose file on the same host?
63. What’s the difference between up, run, and start?
64. What are the differences between the ‘docker run’ and the ‘docker create’?
65. What command should you run to see all running container in Docker?
66. What is the command to run the image as a container?
67. What are the common instruction in Dockerfile?
68. How can you monitor the docker in production environments?
69. What the states of Docker container?
70. What are the steps for the Docker container life cycle?
71. List out some important advanced docker commands
72. Where the docker volumes are stored?
73. Explain Implementation method of Continuous Integration(CI) and Continues Development (CD) in   
    Docker?
74. What is CNM?
75. Does Docker offer support for IPV6?

Kubernetes

1. What is Kubernetes, and why is it important for container orchestration?
2. What is Kubernetes, and how does it relate to Docker?
3. How to use Docker containers in a Kubernetes cluster?
4. Explain the key components of a Kubernetes cluster?
5. What is a Pod in Kubernetes, and why is it the smallest deployable unit?
6. what is Kubernetes architecture?
7. What are the secrets in Kubernetes?
8. What is the difference between Pod and Container
9. What is Service in Kubernetes? How many types are there?
10. What is Replica Set in Kubernetes?
11. What is Deployment in Kubernetes?
12. What is a Daemon Set in Kubernetes?
13. How can you provide env variables to the Pod in a better way
14. How to configure a health check in Pod
15. What is minikube?
16. How to run any pod through kubectl?
17. What do you mean by exposing the pod?

Container Automation Using Ansible

1. What is Ansible?
2. Why Ansible its used?
3. What is Playbook and which programing language it is written?
4. What is the Benefits of Playbook?
5. What is most popular configuration tools?
6. Difference between Chef & Ansible?
7. What programing language Ansible is written in?
8. What is Architecture of Ansible?
9. What is command to run Ansible Playbook?
10. What is called “Linux command used in ansible”?

Terraform Overview

1. What is Terraform?
2. What are the Benefits of Terraform?
3. What is the History of Terraform?
4. Which cloud providers support Terraform?
5. What are provides and resources in Terraform?
6. Which programing language Terraform is written?
7. Which most used Command of execution of Terraform?

Continuous Monitoring using Prometheus and Grafana

Project