

Devops Viva Q&A

1. Q: What is DevOps?
A: DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) to shorten the development lifecycle and deliver high-quality software continuously.
2. Q: State two main objectives of DevOps.
A: (1) Faster delivery of software, (2) Improved collaboration between teams.
3. Q: Explain the history of DevOps.
A: DevOps originated around 2009 as a response to the growing need for agile collaboration between development and operations teams.
4. Q: What is the role of DevOps in SDLC?
A: DevOps automates and integrates processes across development, testing, deployment, and monitoring to improve efficiency.
5. Q: Difference between Agile and DevOps?
A: Agile focuses on iterative development, while DevOps emphasizes continuous delivery and collaboration between dev and ops.
6. Q: What is Continuous Integration?
A: Continuous Integration is the practice of frequently merging code changes into a shared repository and automatically testing them.
7. Q: Define Continuous Deployment.
A: Continuous Deployment is the automated release of code changes to production after passing all tests.
8. Q: What is Jenkins?
A: Jenkins is an open-source automation server used for Continuous Integration/Continuous Deployment (CI/CD).
9. Q: What is a Container?
A: A container is a lightweight, portable unit that packages an application with its dependencies.
10. Q: What is Virtualization?
A: Virtualization is the creation of virtual instances (VMs) of servers, networks, or storage to maximize resource usage.
11. Q: Differentiate between Containers and VMs.
A: Containers share the host OS kernel, while VMs have their own OS making them heavier.

12. Q: Define Configuration Management in DevOps.

A: It is the practice of maintaining system configurations in a consistent state using tools like Ansible or Puppet.

13. Q: What is Ansible?

A: Ansible is an open-source tool for IT automation, configuration management, and application deployment.

14. Q: What is Cloud Computing?

A: Cloud computing is the on-demand delivery of IT resources (compute, storage, networking) via the internet.

15. Q: Differentiate between IaaS, PaaS, SaaS.

A:

- IaaS: Infrastructure (VMs, storage, networks)
- PaaS: Platform (runtime environment, middleware)
- SaaS: Software delivered over the internet (e.g., Gmail)

16. Q: What is a Public Cloud?

A: A cloud infrastructure available to the general public over the internet (e.g., AWS).

17. Q: What is a Private Cloud?

A: A cloud environment dedicated to a single organization.

18. Q: Define Hybrid Cloud.

A: A combination of public and private cloud services for flexibility.

19. Q: What is AWS?

A: Amazon Web Services, a leading cloud service provider offering IaaS, PaaS, SaaS.

20. Q: What is Microsoft Azure?

A: A cloud platform by Microsoft providing compute, analytics, storage, and networking.

21. Q: What is Google Cloud Platform?

A: Google's cloud computing platform providing various services like BigQuery, GKE, Cloud Functions.

22. Q: Mention advantages of Cloud in DevOps.

A: Scalability, cost efficiency, automation, high availability.

23. Q: What is Continuous Monitoring?

A: The practice of monitoring applications and infrastructure continuously in DevOps.

24. Q: Name two Configuration Management tools.

A: Ansible and Puppet.

25. Q: What is Vagrant in DevOps?

A: Vagrant is a tool for building and managing virtual machine environments.

Q: What is Terraform?

A: Terraform is an open-source Infrastructure as Code (IaC) tool for building, changing, and managing cloud infrastructure safely and efficiently.

26. Q: What is the architecture of Terraform?

A: Terraform consists of **Core** (execution plan, state management) and **Providers** (plugins for cloud services like AWS, Azure, GCP).

27. Q: What is a Terraform provider?

A: A plugin that enables interaction with APIs of cloud platforms (e.g., AWS provider, Azure provider).

28. Q: What is a Terraform module?

A: A container for multiple resources that are used together, making infrastructure reusable.

29. Q: What is a Terraform state file?

A: A JSON file that records the current state of infrastructure, allowing Terraform to track resources.

30. Q: What is terraform init command?

A: It initializes a working directory containing Terraform configuration files.

31. Q: What does terraform plan do?

A: It shows the execution plan before applying changes.

32. Q: What is terraform apply?

A: Executes the plan and creates/updates infrastructure.

33. Q: What is terraform destroy?

A: Removes all resources defined in Terraform configuration.

34. Q: What is Infrastructure as Code (IaC)?

A: Managing infrastructure using code rather than manual processes.

35. Q: Difference between Terraform and Ansible?

A: Terraform focuses on provisioning infrastructure (IaC), while Ansible focuses on configuration management.

36. Q: What is Ansible?

A: An open-source automation tool for configuration management, application deployment, and orchestration.

37. Q: What language does Ansible use for automation?

A: YAML (Yet Another Markup Language).

38. **Q:** What is an Ansible Playbook?
A: A YAML file containing instructions to automate tasks.
39. **Q:** What is an Ansible Inventory?
A: A file that lists all the hosts/servers Ansible manages.
40. **Q:** What are Ansible Handlers?
A: Special tasks triggered only when notified, usually used for restarting services.
41. **Q:** What are Ansible Roles?
A: A structured way of organizing playbooks into reusable components.
42. **Q:** What are Host Variables in Ansible?
A: Variables specific to a single host in the inventory.
43. **Q:** What are Group Variables in Ansible?
A: Variables applied to a group of hosts.
44. **Q:** What are Ansible Modules?
A: Reusable units of code used to automate tasks (e.g., copy, yum, service).
45. **Q:** What is ansible-galaxy?
A: A command-line tool to download and share Ansible roles.
46. **Q:** What is the difference between Playbook and Ad-hoc commands?
A: Playbooks are reusable YAML scripts, while ad-hoc commands are one-time commands for quick tasks.
47. **Q:** What is Ansible Tower?
A: An enterprise version of Ansible with a GUI, role-based access control, and scheduling.
48. **Q:** What are Ansible Conditionals?
A: Logic statements in playbooks that execute tasks based on conditions.
49. **Q:** What are Ansible Loops?
A: Used to repeat tasks multiple times in a playbook.
51. **Q:** What is Docker?
A: Docker is a containerization platform that packages applications and dependencies into lightweight containers.
52. **Q:** What is a Docker Image?
A: A read-only template used to create Docker containers.
53. **Q:** What is a Docker Container?
A: A running instance of a Docker image.

54. **Q:** Difference between Docker Image and Container?
A: Image is a blueprint; Container is a running instance.
55. **Q:** What is the Docker Engine?
A: The core part of Docker that creates and manages containers.
56. **Q:** Command to list running containers?
A: `docker ps`
57. **Q:** Command to stop a container?
A: `docker stop <container_id>`
58. **Q:** Command to remove a Docker image?
A: `docker rmi <image_name>`
59. **Q:** What is Docker Compose?
A: A tool for defining and managing multi-container applications using a YAML file.
60. **Q:** What is a Dockerfile?
A: A script containing instructions to build a Docker image.
61. **Q:** What is Version Control?
A: A system to track changes in code over time.
62. **Q:** What is Git?
A: A distributed version control system used for source code management.
63. **Q:** What is GitHub?
A: A cloud-based hosting service for Git repositories.
64. **Q:** What does `git init` do?
A: Initializes a new Git repository.
65. **Q:** What does `git clone` do?
A: Creates a copy of an existing repository.
66. **Q:** What does `git add` do?
A: Stages changes to be committed.
67. **Q:** What does `git commit` do?
A: Saves staged changes into the repository history.
68. **Q:** What does `git push` do?
A: Uploads local commits to a remote repository.
69. **Q:** What does `git pull` do?
A: Fetches and merges changes from a remote repository.

70. **Q:** What is git branch?
A: Creates or lists branches in a repository.
71. **Q:** What is Git Merge?
A: Combines changes from one branch into another.
72. **Q:** What is Git Rebase?
A: Moves commits from one branch to another base.
73. **Q:** What is Git Stash?
A: Temporarily saves uncommitted changes without committing them.
74. **Q:** Difference between Git Fetch and Git Pull?
A: Fetch only downloads changes; Pull downloads and merges.
75. **Q:** What is a GitHub Pull Request?
A: A request to merge changes from one branch/repository into another.
76. **Q:** What is Jenkins?
A: Jenkins is an automation server used for Continuous Integration and Continuous Deployment.
77. **Q:** What is a Jenkins Job?
A: A Jenkins job is a task that Jenkins runs, such as building or testing software.
78. **Q:** What is a Jenkins Pipeline?
A: A set of automated steps in Jenkins that define how to build, test, and deploy software.
79. **Q:** What are Jenkins Plugins?
A: Extensions that add extra functionalities to Jenkins.
80. **Q:** What is Jenkins Master-Slave Architecture?
A: A setup where the master controls build jobs, and slaves execute them.
81. **Q:** What is Jenkins Authentication?
A: The process of verifying users before granting access to Jenkins.
82. **Q:** What is Jenkins Authorization?
A: The process of defining what actions authenticated users can perform.
83. **Q:** What are Sequential Builds in Jenkins?
A: Builds executed one after another in a defined order.
84. **Q:** What are Parallel Builds in Jenkins?
A: Builds executed simultaneously to save time.
85. **Q:** What is Jenkins Workspace?
A: A directory on the Jenkins server where build files are stored.

86. **Q:** What is Jenkins SCM plugin?
A: A plugin that integrates source control systems (Git, SVN) with Jenkins.
87. **Q:** What is Maven Build in Jenkins?
A: A build automation process using Apache Maven integrated into Jenkins.
88. **Q:** What is Jenkins Scheduling?
A: Setting up jobs to run at specific times using CRON syntax.
89. **Q:** What is a Jenkins Build Trigger?
A: An event that starts a Jenkins job automatically (e.g., Git commit).
90. **Q:** How do you secure Jenkins?
A: By enabling authentication, role-based access control, and using HTTPS.
91. **Q:** What is Kubernetes?
A: An open-source system for automating deployment, scaling, and management of containerized applications.
92. **Q:** What is a Kubernetes Pod?
A: The smallest deployable unit in Kubernetes, containing one or more containers.
93. **Q:** What is a Kubernetes Node?
A: A worker machine in Kubernetes that runs Pods.
94. **Q:** What is a Kubernetes Cluster?
A: A set of nodes managed by a Kubernetes control plane.
95. **Q:** What is the Kubernetes Control Plane?
A: The central component managing cluster state and scheduling.
96. **Q:** What is a Kubernetes Deployment?
A: A controller that manages stateless applications and ensures desired replicas.
97. **Q:** What is a Kubernetes Service?
A: An abstraction that defines a logical set of Pods and a way to access them.
98. **Q:** What is a Kubernetes Ingress?
A: Manages external access to services, typically via HTTP/HTTPS.
99. **Q:** What is a Kubernetes ConfigMap?
A: A way to store non-confidential configuration data in key-value pairs.
100. **Q:** What is a Kubernetes Secret?
A: A Kubernetes object used to store sensitive information like passwords, tokens, and keys.