Jenkins Interview Q&A Guide

Q1) What is Jenkins? Jenkins is an open-source automation server used to automate software development processes like building, testing, and deploying code. It supports Continuous Integration (CI) and Continuous Delivery (CD), making DevOps workflows efficient.

Q2) What are the features of Jenkins? - Open-source and free to use. - Supports distributed builds across multiple machines. - Extensible via a vast number of plugins. - Easy installation and configuration. - Supports multiple languages and tools. - Provides a web-based interface and CLI. - Supports pipeline as code using Jenkinsfile.

Q3) What are the advantages of Jenkins? Why do we use Jenkins? Advantages: - Automates repetitive tasks like builds and deployments. - Enables faster integration of code changes. - Provides real-time feedback to developers. - Reduces human error in software delivery. - Easy integration with tools like Git, Docker, Maven, Kubernetes.

Why we use Jenkins: To achieve Continuous Integration and Continuous Delivery, ensuring faster and reliable software development cycles.

Q4) Mention some of the important plugins in Jenkins - Git Plugin - Maven Plugin - Pipeline Plugin - Docker Plugin - Email Extension Plugin - Slack Notification Plugin - Blue Ocean

Q5) What is Continuous Integration (CI) in Jenkins? Continuous Integration is a DevOps practice where developers frequently merge code changes into a central repository. Jenkins automatically builds and tests the code, ensuring early detection of bugs.

Q7) What is Groovy in Jenkins? Groovy is a scripting language used in Jenkins to define Jenkins pipelines. Both Declarative and Scripted pipelines are written in Groovy, allowing automation of complex tasks.

Q9) What is Jenkinsfile? A Jenkinsfile is a text file that contains the definition of a Jenkins pipeline. It allows pipeline-as-code, making builds reproducible and version-controlled.

Q10) Difference between Continuous Integration, Continuous Delivery, and Continuous Deployment

Feature	Continuous Integration (CI)	Continuous Delivery (CD)	Continuous Deployment
Purpose	Merge code and detect errors early	Automate testing & staging deployment	Automate deployment to production
Automation	Build + Unit tests	Build + Test + Staging	Build + Test + Staging + Production
Human	No	Required for production	No

Intervention		deployment	

Q11) What is Jenkins Pipeline? What is a CI/CD pipeline?

- Jenkins Pipeline: A set of automated steps defined in code (Jenkinsfile) for building, testing, and deploying software.

- CI/CD Pipeline: End-to-end automated workflow that manages code integration (CI) and delivery/deployment (CD) through multiple stages (build → test → deploy).

Q12) Difference between Scripted Pipelines and Declarative Pipelines in Jenkins

Feature	Scripted Pipeline	Declarative Pipeline
Syntax	Imperative (written in Groovy code, more flexible)	Declarative (predefined, structured syntax)
Ease of Use	Harder for beginners; requires more coding	Easier for beginners; readable and structured
Flexibility	Highly flexible; allows complex logic and dynamic behavior	Less flexible; focuses on simplicity and best practices
Error Handling	Must explicitly handle errors	Supports `post` blocks for easy handling of success, failure, always, unstable etc.

- **Scripted pipelines** = flexible, powerful, requires Groovy knowledge.
- **Declarative pipelines** = structured, simple, easier to read, maintain, and recommended for most CI/CD workflows.

Q14) What is SCM? Which SCM tools are supported in Jenkins?

- **SCM (Source Code Management):** System that manages code versions and revisions.
- Supported SCMs in Jenkins: **Git, SVN, Mercurial, Perforce, CVS, TFS.**

Q15) Which CI Tools are used in Jenkins? -

- Build Tools: **Maven, Gradle, Ant**
- Version Control: **Git, SVN**
- Testing Tools: **JUnit, Selenium**
- Containerization: **Docker, Kubernetes**
- Notification: **Slack, Email**

Q16) Where is the Jenkins password stored?

- Initial admin password: /var/lib/jenkins/secrets/initialAdminPassword - User passwords are stored encrypted in credentials.xml inside Jenkins home directory.

Q17) Where is Jenkins home directory? Default location: /var/lib/jenkins

Q18) What are the two components (pre-requisites) that Jenkins is mainly integrated with? 1. Source Code Management (SCM) – e.g., Git, SVN 2. Build Tools / Testing Tools – e.g., Maven, Gradle, Ant, Selenium

Q19) How can you clone a Git Repository via Jenkins?

- Install the **Git Plugin** in Jenkins.
- Create a new job → Configure → Source Code Management → Git.
- Enter repository URL and credentials.
- Jenkins will automatically clone the repository for build and deployment.

Q20) How can you secure Jenkins?

- Enable **authentication and authorization**.
- Use **matrix-based security** for role control.
- Use **HTTPS** instead of HTTP.
- Restrict plugin installation to trusted sources.
- Backup Jenkins regularly.
- Disable unused endpoints and anonymous access.

Q21) How to create a backup and copy files in Jenkins?

- Backup by copying JENKINS_HOME directory:

sudo cp -r /var/lib/jenkins /path/to/backup/

- Use plugins like ThinBackup or Backup Plugin

Q22) Why is Jenkins called a Continuous Delivery Tool? Because Jenkins automates the entire software delivery process—from building, testing, staging to deployment—enabling teams to release software quickly, reliably, and continuously.