
DevOps Shack

Top 50 Jenkins Commands and Concepts asked in MNC interviews

1. What is Jenkins?

- Answer: Jenkins is an open-source automation server written in Java. It helps automate parts of software development, such as building, testing, and deploying.

Key Points:

- Built-in plugins for CI/CD.
- Supports distributed builds across multiple machines.

2. What is Continuous Integration in Jenkins?

- Answer: Continuous Integration (CI) is a software development practice where developers regularly integrate code changes into a shared repository.

Jenkins Role:

- Automates build and testing processes.
- Detects integration issues early.

3. How do you install Jenkins?

- Command (Linux):

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key |
```

```
sudo apt-key add -  
sudo apt update  
sudo apt install jenkins
```

Explanation:

- Installs Jenkins on a Debian-based Linux system.
- Uses the Jenkins repository for package installation.

4. How do you start/stop Jenkins?

- Commands:

```
sudo systemctl start jenkins  
sudo systemctl stop jenkins  
sudo systemctl restart jenkins
```

Explanation:

- Controls Jenkins service via **systemd**.

5. Where are Jenkins configuration files stored?

- Location:

```
/var/lib/jenkins
```

- Explanation: Jenkins home directory stores configuration files, plugins, and job data.

6. How to check Jenkins logs?

- Command:



```
sudo tail -f /var/log/jenkins/jenkins.log
```

Explanation:

- Monitors real-time logs for troubleshooting.

7. How do you secure Jenkins?

- Answer:
 1. Enable Role-based Access Control.
 2. Use an SSL certificate.
 3. Configure credentials securely.

8. How do you create a Jenkins pipeline?

- Command (Declarative Pipeline):

```
pipeline {  
  agent any  
  stages {  
    stage('Build') {  
      steps {  
        echo 'Building...'  
      }  
    }  
  }  
}
```

Explanation:

- Declarative pipelines use a script-like structure to define CI/CD processes.



9. How do you set environment variables in Jenkins pipelines?

- Command:

```
pipeline {  
    environment {  
        ENV_VAR = 'value'  
    }  
    stages {  
        stage('Test') {  
            steps {  
                echo "Environment variable: ${ENV_VAR}"  
            }  
        }  
    }  
}
```

Explanation:

- The **environment** block sets variables globally for pipeline stages.

10. What is the difference between Freestyle and Pipeline jobs?

- Answer:
 - Freestyle: Basic job setup with GUI.
 - Pipeline: Scripted or declarative workflows for complex automation.

11. How do you schedule a Jenkins job?

- Command: Use CRON syntax.



H/15 * * * *

- Explanation: Runs the job every 15 minutes.

12. How do you trigger a build remotely in Jenkins?

- Command:

```
curl -X POST http://<JENKINS_URL>/job/<JOB_NAME>/build  
--user <USER>:<TOKEN>
```

Explanation:

- Triggers builds using the Jenkins REST API.

13. What is Jenkinsfile?

- Answer: A text file that stores pipeline as code. Typically stored in a repository.

14. How do you back up Jenkins?

- Command:

```
cp -r /var/lib/jenkins /backup/location
```

Explanation:

- Copies configuration, plugins, and job data.

15. How do you install plugins in Jenkins?

- Command (CLI):



```
java -jar jenkins-cli.jar -s http://<JENKINS_URL>  
install-plugin <PLUGIN_NAME>
```

16. What are Jenkins agents and nodes?

- Answer:
 - Agent: A remote system that builds jobs.
 - Node: A machine where Jenkins runs jobs.

17. How do you configure a Jenkins agent?

- Command:

```
java -jar agent.jar -jnlpUrl <URL> -secret <SECRET> -workDir  
"/path/to/workdir"
```

18. How do you archive artifacts in Jenkins?

- Command:

```
archiveArtifacts artifacts: 'target/*.jar'
```

Explanation:

- Saves build outputs (e.g., JAR files) for future use.

19. How do you set up Git in Jenkins?

- Command:

```
git config --global user.name "Your Name"  
git config --global user.email "youremail@example.com"
```

- Add Git plugin in Jenkins.



20. What is Jenkins Blue Ocean?

- Answer: A modern UI for Jenkins pipelines with a user-friendly design.

21. How do you integrate Jenkins with Docker?

- Command (Pipeline example):

```
pipeline {  
  agent {  
    docker { image 'node:14' }  
  }  
  stages {  
    stage('Build') {  
      steps {  
        sh 'npm install'  
      }  
    }  
  }  
}
```

22. How do you restart Jenkins?

- Command:

```
sudo systemctl restart jenkins
```

23. What is Jenkins CLI?

- Answer: Command-line interface for managing Jenkins.

Example Command:

```
java -jar jenkins-cli.jar -s http://<JENKINS_URL> list-jobs
```



24. What is the purpose of Jenkins pipeline syntax generator?

- Answer: Helps users create pipeline scripts using a GUI.

25. How do you deploy code using Jenkins?

- Command:

```
sh 'scp target/app.war user@server:/path/to/deploy'
```

26. What are shared libraries in Jenkins?

- Answer: Reusable pipeline code stored in a Git repository.

27. How do you monitor Jenkins jobs?

- Answer:
 - Use the Build History and Dashboard.
 - Integrate monitoring tools like Prometheus.

28. How do you configure Jenkins to run in Docker?

- Command:

```
docker run -d -p 8080:8080 -p 50000:50000  
jenkins/jenkins:lts
```

29. What are the stages in Jenkins pipelines?

- Answer:
 - Stages represent the steps of the CI/CD process (e.g., Build, Test, Deploy).

30. How do you parameterize a Jenkins build?

- Command:




```
parameters {  
    string(name: 'BRANCH', defaultValue: 'main',  
description: 'Branch to build')  
}
```

31. How do you use a Docker agent in Jenkins?

- Command:

```
agent { docker { image 'maven:3.8.4' } }
```

32. How do you perform post-build actions in Jenkins?

- Command:

```
post {  
    always {  
        mail to: 'team@example.com', subject: 'Build  
Complete', body: 'Details...'  
    }  
}
```

33. How do you implement parallel execution in Jenkins?

- Command:

```
parallel {  
    stage('Test A') { steps { echo 'Running Test A' } }  
}
```



```
    stage('Test B') { steps { echo 'Running Test B' } }  
}
```

34. How do you integrate Jenkins with Maven?

- Command:

```
sh 'mvn clean install'
```

35. What is the use of workspace in Jenkins?

- Answer: Temporary location for job execution files.

36. What is the Jenkins environment variable **BUILD_NUMBER**?

- Answer: Represents the current build number.

37. How do you troubleshoot Jenkins pipeline errors?

- Answer:
 - Check the Console Output.
 - Analyze logs in **/var/log/jenkins**.

38. What is a webhook in Jenkins?

- Answer: A trigger mechanism for CI when changes are pushed to the repository.

39. How do you handle secrets in Jenkins?

- Answer:
 - Use the Credentials plugin.
 - Mask secrets in pipelines.

40. What is the **JENKINS_HOME** variable?

- Answer: Points to the Jenkins root directory.



41. How do you debug Jenkins pipelines?

- Answer: Add **echo** and use **debug** options.

42. How do you trigger a downstream job in Jenkins?

- Command:

```
build job: 'Downstream_Job_Name'
```

43. How do you integrate Slack with Jenkins?

- Answer: Use the Slack plugin and configure it with a webhook.

44. What is the role of a Jenkinsfile?

- Answer: Pipeline-as-code for version-controlled CI/CD.

45. How do you test Jenkins pipelines locally?

- Answer: Use tools like **Jenkinsfile Runner**.

46. What is the difference between scripted and declarative pipelines?

- Answer:
 - Scripted: More flexible, written in Groovy.
 - Declarative: Easier to use, structured syntax.

47. How do you integrate Jenkins with AWS?

- Answer:
 - Use the AWS CLI and plugins like S3 Publisher.

48. How do you install Jenkins as a Windows service?



- Command:

```
java -jar jenkins.war
```

49. What are Jenkins plugins?

- Answer:
 - Add additional features like Git integration, Slack notifications, etc.

50. How do you configure email notifications in Jenkins?

- Command:

```
emailx to: 'team@example.com', subject: 'Build Status',  
body: 'Build details here...'
```