

Jenkins Basic Interview Questions

Q1) What is Jenkins and why to use it?

Ans. Jenkins is one of the leading open-source continuous integration tools. The main functionality of this tool is to keep track of the version control system and monitor the build system, and provide notifications and reports to alert. It enables you to deliver software by integrating with a large number of testing and deployment technologies.

The following are the reasons to use Jenkins:

- It possesses an installer package for major operating systems.
- Integrates individual projects for a larger purpose
- To keep your team in sync
- Troubleshoot and audit past jobs effortlessly
- Provides accurate data support for project management

Q2) What is continuous integration?

Ans. Continuous integration is a process of continuously checking the developer's code into a version control system several times a day and automating the build to check and detect bugs in the written code. Continuous Integration includes the following:

- Development and Compilation
- Database Integration
- Unit Testing
- Production Deployment
- Code Labeling
- Functional Testing
- Generating and Analyzing Reports

Q3) What are the features of Jenkins?

Ans. Jenkins comes with the following features:

- Free open source.
- Easy [installation](#) on various operating systems.
- Build Pipeline Support.
- Workflow Plugin.
- Test harness built around JUnit.
- Easy upgrades.
- Rapid release cycle.
- Easy configuration setup.
- Excellent community and documentation
- Extensible with the use of third-party plugins.

Q4) What are the advantages of using Jenkins?

Ans. The advantages of using Jenkins are the following:

- Open-source tool and user-friendly
- Easy to install
- Provides great collaboration between development and operations teams.
- Code deployment is easy and happens in minutes, along with the generation of reports.
- Free of cost
- Platform independent
- Rich plugin ecosystem
- Code errors can be detected as early as possible.
- Automation of integration work, thereby reducing the number of integration issues.

Q5) What are the prerequisites to use Jenkins?

Ans. We require the following to use Jenkins:

- A source code repository which is accessible, for instance, and a Git repository
- A working build script. e.g., a Maven Script, checked into the repository

Q6) Name some of the plugins in Jenkin?

Ans. Some of the important plugins in Jenkin includes:

- Maven 2 project
- Amazon EC2
- HTML publisher
- Copy Artifact
- Join
- Green Balls
- Git plugin
- Multijob plugin
- Test Results Analyzer
- Metrics

Q7) How to restart Jenkins manually?

Ans. To restart Jenkins manually, you can use any one of the following commands:

Subscribe to our youtube channel to get new updates..!

-

(jenkins_url)/safeRestart - Allows all running jobs to complete. New jobs will remain in the queue to run after the restart is complete.

- (jenkins_url)/restart - Forces a restart without waiting for builds to complete.

Q8) What are the components that you can integrate Jenkins with?

Ans. Jenkins is mainly integrated with the following:

- Version Control system like GIT, SVN
- And build tools like Apache Maven.

Q9) How does Hudson relate to Jenkins?

Ans. Jenkins was a renamed version of Hudson.

Q10) How to install Jenkins?

Ans.

- Install Java Version 8
- Install [Apache](#) Tomcat Version 9
- Download Jenkins war File
- Deploy Jenkins war File
- Install Suggested Plugins

Q11) How can you start Jenkins manually?

Ans. Jenkins can be manually started by opening the Console/Command line and using the below commands:

- Start Jenkins: jenkins.exe start
- Stop Jenkins: jenkins.exe stop
- Restart Jenkins: jenkins.exe restart

Q12) What is the difference between Jenkins, Maven, and Ant?

Ans.

Jenkins	Maven	Ant
Continuous Integration Tool	Build automation tool	Command Line/Java Library Tool.
Automates software development process through continuous integration and facilitates continuous delivery.	Describes software dependencies and explains how the software is built	Drives build process
Supports version control tools like Git, AccuRev.	Supports projects written in C#, Ruby .	Supports projects written in C and C++.

Q13) Name a few Jenkins environment variables.

Ans. By default, there are numerous variables available in Jenkins. Some of them are:

- \$NODE_NAME
- \$JOB_NAME
- \$WORKSPACE
- \$JOB_URL
- \$BUILD_URL

Q14) How can you deploy a custom build of a core plugin?

Ans. To deploy a custom build of a core plugin, you have to do the following:

- Stop Jenkins.

- Copy the custom HPI to \$Jenkins_Home/plugins.

JENKINS Certification Training!

Explore Curriculum

-

Delete the previously expanded plugin directory.

- Make an empty file called . hpi. pinned.
- Start Jenkins.

Q15) What are the Parameters in Jenkins?

Ans. Parameters are supported by the Agent section. They are used to support several use-cases pipelines and are defined at the top-level of the pipeline or inside an individual stage directive.

Q16) What is the use of Agent Directive?

Ans. The agent directive specifies Jenkins how and where to execute the entire pipeline. The directive is specified at the top-level inside the pipeline block and stage-level usage is optional.

Q17) What is flow control in Jenkins?

Ans. Flow control supports the pipeline structure (scripted pipeline) for executing the top to bottom of the Jenkins file.

Q18) What are the ways to configure Jenkins node agent to communicate with Jenkins master?

Ans. Below mentioned ways help you to start the node agent –

- Command-line – For this, the client needs the executable agent.jar file. It helps to launch the process on the client to communicate with the Jenkins master to run build jobs.
- Browser – This file launches a new process on the client machine to run jobs.

Q19) What is a Jenkins Pipeline?

Ans. Jenkins Pipeline is a set of features of Jenkins, which are installed as plugins to enable continuous delivery pipelines implementation. These are the automated processes to get the software from source control through deployment to end-users.

Q20) What is the use of JENKINS HOME directory?

Ans. JENKINS_HOME directory is used to store all the settings, configurations, and logs.

Jenkins Advanced Interview Questions

Q21) How do I move Jenkins from one server to another?

Ans. There are multiple ways to do that as mentioned below:

- Move a job from one installation of Jenkins to another by simply copying the corresponding job directory.
- Make a copy of an existing job by making a clone of a job directory by a different name.
- Rename an existing job by renaming a directory.

Q22) Explain how to create a backup and copy files in Jenkins?

Ans. Jenkins saves all the build logs and configurations in the home directory. It includes all the slave node configurations and job configurations. For performing back of Jenkins, you should back up the JENKINS_HOME directory frequently. For copying Jenkins jobs, you should replicate the job directory.

Q23) Explain how to Setup Build jobs in Jenkins.

Ans. The following steps will help you to build jobs in Jenkins:

Step 1: First, go to the Jenkins dashboard and click on the New Item.

Step 2: Enter the Item name and choose the 'Freestyle project option'.

Step 3: Specify the details of the job.

Step 4: Next, specify the location of files which need to be built.

Step 5: If your repository is hosted on Github, you can also enter the URL of that repository here.

Step 6: Build section and click on the Add build step.

Step 7: In the command window, enter the following commands and then click on the Save button.

```
( Javac HelloWorld.java
```

Java HelloWorld)

Step 8: You can click on the Build Now option to see if you have successfully defined the job.

Step 9: Once the build is scheduled, it will run.

Step 10: Click on the Console Output link to see the details of the build

Q24) What is Groovy in Jenkins?

Ans. Apache Groovy is the default scripting language used for the JVM platform. It has some of useful features like DSL support, dynamic typing, and closures.

Some of the major features of using Groovy are:

- Groovy is a dynamic and agile language, offers seamless integration with all the existing Java libraries and objects.
- We can use scopes to define Collections or Arrays.
- Allows you to add or remove collections

Q25) How do I clone a git repository using Jenkins?

Ans. You have to enter the user name and email for your Jenkins system and turn into your job directory and execute the “git config” command for that.

Q26) What are the ways by which build can be scheduled in Jenkins?

Ans. You can schedule a build in Jenkins in the following ways:

- By source code management commits
- After completion of other builds
- Can be scheduled to run at a specified time (crons)
- Manual Build Requests

Q27) How to configure and use a third-party tool in Jenkins?

Ans. The procedure to work with a third-party tool in Jenkins is:

- First, the third-party tool must be installed.

- Download the plug-in that supports the third-party tool through the Jenkins administrator console.
- The third-party tool must be configured in the administrator console.
- At last, the plug-ins can be utilized from inside a Jenkins build job.

Q28) What are the different ways of scheduling a build in Jenkins?

Ans.

- Builds can be triggered by source code management commits sequentially.
- Manual requests the Builds.
- Can be scheduled to run at a specified time using the CRON jobs.

Q29) How will you secure Jenkins?

Ans. The following ways will help you to secure Jenkins:

- Check the global security is on.
- Make sure Jenkins is integrated with appropriate login with my company's user directory.
- Project matrix/matrix is allowed to fine-tune access.
- Automate the process of setting rights/privileges with custom version controlled script in Jenkins.
- Limit physical access to Jenkins data/folders.
- Run security audits periodically on the same.

Q30) What are the steps involved in deploying a custom build of a core plugin?

Ans. To deploy a custom build of a core plugin, you should follow the below steps:

- Stop Jenkins.
- Copy the custom HPI to \$Jenkins_Home/plugins.
- Delete the previously expanded plugin directory.
- Make an empty file called . hpi. pinned.
- Start Jenkins

Q31) What are declarative pipelines?

Ans. A declarative pipeline is a new feature in Jenkins that maintains the pipeline as a code and makes it easier to write and read. It is defined within a block labelled pipeline.

Syntax:

The common syntax is:

```
1 | pipeline {  
2 |   /* Declarative Pipeline */  
3 | }
```

Structure of the declarative pipeline:

- any – It represents the whole pipeline will run on any available agent.
- docker – This is to run the pipeline in the Docker environment.
- none – It indicates all the stages under the block will have to be declared with the agent separately.
- label – This is just a label for the Jenkins environment

Q32) How to define parameters for a build in Jenkins?

Ans. Build can use various input parameters for execution. For suppose, you have different test suites and you want to run only one then you can select a parameter to decide which one should be run. To define parameters for your job, first, you need to define the same while specifying the parameter. There are different parameter types like string, file, or a custom.

Q33) What are scripted pipelines?

Ans. Scripted pipelines are writing Jenkins pipelines as code in a traditional way. The scripted pipeline is defined within a 'node'.

Syntax of the Scripted Pipeline:

The node is a part of the Jenkins architecture, where there are two types of a node, the agent or node will run the part of the workload of the jobs and master node will handle the configuration of the job.

Q34) What does SCM mean in Jenkins?

Ans. Source Code Management (SCM) specifies the source code location in Jenkins. The entry point to SCM is defined as `jenkins_jobs.scm`. The job defined with the SCM attribute accepts many numbers of SCM definitions. Some of the SCM tools are CVS, Git, Perforce, AccuRev, Subversion, Clearcase, RTC, and Mercurial.

SCM can be defined as:

```
1 scm:
2   name: eloc - scm
3   scm:
4     git:
5     url: ssh://Jenkins.org/eloc.git
```

Q35) What is the difference between Agent, post-section, and Jenkinsfile?

Ans.

- Agent: A directive which specifies the Jenkins on how to execute the pipeline in a particular order or manner.
- Jenkinsfile: The text file which defines all the pipelines is called Jenkinsfile. It is being checked in the source control repository.
- Post-section: It runs at the end of every pipeline's execution.

Q36) How to achieve continuous integration using Jenkins?

Ans.

- Developers must commit their source code changes to the shared Git repository.
- The test results and build results are shared with the respective developers.
- Jenkins server checks the shared Git repository at particular time periods and identifies the changes taken in the build.
- The clean and tested build is deployed to the production server.

Q37) How do you define a Continuous Delivery Workflow?

Ans. The flowchart below shows the Continuous Delivery Workflow.

Q38) How do you integrate Git with Jenkins?

Ans. The below steps show you how to integrate Git with Jenkins:

1. Manage Jenkins button on your Jenkins dashboard.
2. Click on Manage Plugins.
3. In the Plugins Page
 - Select the GIT Plugin and click on Install without restart.
 - You can also select the option Download now and install after restart.
 - If you already have the Git plugin installed, you will see a “No updates available” message.

Go to Manage Jenkins on your Jenkins dashboard.

Q39) Explain Continuous Testing.

Ans. Continuous Testing is the process of executing automated tests of the software delivery pipeline. It helps you to identify the business risks associated with software as early as possible. In this, testing takes place on a continuous basis without any disruption.

Explore Jenkins Sample Resumes! Download & Edit, Get Noticed by Top Employers!

Download Now!

Q40) What is DSL Jenkins?

Ans. Job DSL is one of the most important plugins for Jenkins that allows managing configuration as code. It is made up of two parts. First, Domain Specific Language (DSL) that enables users to define jobs utilising a Groovy-based language, and second, a Jenkins plugin that manages the scripts and the updates the created Jenkins jobs.