JENKINS Interview Questions & Answers (2022 Update)

1) Mention what is Jenkins?
Jenkins is an open source tool with plugin built for continuous integration purpose. The principle functionality of Jenkins is to keep a track of version control system and to initiate and monitor a build system if changes occur. It monitors the whole process and provides reports and notifications to alert.

2) Explain what is continuous integration?

In software development, when multiple developers or teams are working on different segments of same web application, we need to perform integration test by integrating all modules. In order to do that an automated process for each piece of code is performed on daily bases so that all your code get tested.

3) What is the requirement for using Jenkins?

To use Jenkins you require

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

4) Mention what are the advantages of Jenkins?

Advantage of Jenkins include

- At integration stage, build failures are cached
- For each code commit changes an automatic build report notification generates
- To notify developers about build report success or failure, it is integrated with LDAP mail server
- Achieves continuous integration agile development and test driven development
- With simple steps, maven release project is automated
- Easy tracking of bugs at early stage in development environment than production

5) Explain how you can move or copy Jenkins from one server to another?

- Slide a job from one installation of Jenkins to another by copying the related job directory
- Make a copy of an already existing job by making clone of a job directory by a different name
- Renaming an existing job by renaming a directory.

6) Mention what are the commands you can use to start Jenkins manually?
To start Jenkins manually, you can use either of the following

- (Jenkins_url)/restart: Forces a restart without waiting for builds to complete
- (Jenkin_url)/safeRestart: Allows all running builds to complete

7) Mention some of the useful plugins in Jenkin?

Some of the important plugins in Jenkin includes

- Maven 2 project
- Amazon EC2
- HTML publisher
- Copy artifact
- Join
- Green Balls

8) Explain how you can deploy a custom build of a core plugin?
To deploy a custom field of a core plugin, you have to do following things

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

9) Explain how can create a backup and copy files in Jenkins?

Jenkins saves all the setting, build artifacts and logs in its home directory, to create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

10) Explain how you can clone a Git repository via Jenkins?

To clone a Git repository via Jenkins, you have to enter the e-mail and user name for your Jenkins system. For that, you have to switch into your job directory and execute the “git config” command.

11) Explain how you can set up Jenkins job?

To create a project that is handled via jobs in Jenkins. Select New item from the menu, once this done enter a name for the job and select free-style job. Then click OK to create new job in Jenkins. The next page enables you to configure your job.

12) Mention what are the two components Jenkins is mainly integrated with?

Jenkin is mainly integrated with two components

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

13) How does Jenkins work?

Jenkins is a server-based application and requires a web server like Apache Tomcat to run on various platforms like Windows, Linux, macOS, Unix, etc. To use Jenkins, you need to create pipelines which are a series of steps that a Jenkins server will take. Jenkins Continuous Integration Pipeline is a powerful instrument that consists of a set of tools designed to **host**, **monitor**, **compile** and **test** code, or code changes, like:

- **Continuous Integration Server** (Jenkins, Bamboo, CruiseControl, TeamCity, and others)
- **Source Control Tool** (e.g., CVS, SVN, GIT, Mercurial, Perforce, ClearCase and others)
- **Build tool** (Make, ANT, Maven, Ivy, Gradle, and others)
- **Automation testing framework** (Selenium, Appium, TestComplete, UFT, and others)

Why use Continuous Integration with Jenkins?

Some people might think that the old-fashioned way of developing the software is the better way. Let’s understand the advantages of CI with Jenkins with the following example

Let us imagine, that there are around 10 developers who are working on a shared repository. Some developer completes their task in 25 days while others take 30 days to complete.

Before Jenkins	After Jenkins
Once all Developers had completed their assigned coding tasks, they used to commit their code all at same time. Later, Build is tested and deployed. Code commit built, and test cycle was very infrequent, and a single build was done after many days.	The code is built and test as soon as Developer commits code. Jenkin will build and test code many times during the day If the build is successful, then Jenkins will deploy the source into the test server and notifies the deployment team. If the build fails, then Jenkins will notify the errors to the developer team.
Since the code was built all at once, some developers would need to wait until other developers finish coding to check their build	The code is built immediately after any of the Developer commits.

It is not an easy task to isolate, detect, and fix errors for multiple commits.	Since the code is built after each commit of a single developer, it’s easy to detect whose code caused the built to fail
Code build and test process are entirely manual, so there are a lot of chances for failure.	Automated build and test process saving timing and reducing defects.
The code is deployed once all the errors are fixed and tested.	The code is deployed after every successful build and test.
Development Cycle is slowThe development cycle is fast. New features are more readily available to users. Increases profits.	Development Cycle is slowThe development cycle is fast. New features are more readily available to users. Increases profits.

Jenkins Plugins

By default, Jenkins comes with a limited set of features. If you want to integrate your Jenkins installation with version control tools like Git, then you need to install plugins related to Git. In fact, for integration with tools like Maven, Amazon EC2, you need to install respective plugins in your Jenkins.

Advantages of using Jenkins

- Jenkins is being managed by the community which is very open. Every month, they hold public meetings and take inputs from the public for the development of Jenkins project.
- So far around 280 tickets are closed, and the project publishes stable release every three months.
- As technology grows, so does Jenkins. So far Jenkins has around 320 plugins published in its plugins database. With plugins, Jenkins becomes even more powerful and feature rich.
- Jenkins tool also supports cloud-based architecture so that you can deploy Jenkins in cloud-based platforms.
- The reason why Jenkins became popular is that it was created by a developer for developers.

Disadvantages of using Jenkins

Though Jenkins is a very powerful tool, it has its flaws.

- Its interface is out dated and not user friendly compared to current UI trends.
- Though Jenkins is loved by many developers, it’s not that easy to maintain it because Jenkins runs on a server and requires some skills as server administrator to monitor its activity.

- One of the reasons why many people don’t implement Jenkins is due to its difficulty in installing and configuring Jenkins.
- Continuous integrations regularly break due to some small setting changes. Continuous integration will be paused and therefore requires some developer attention.