Jenkins Interview Questions

1) What is Jenkins?

A) Jenkins is an open source Continuous Integration and Continuous Delivery Tool.

2) What is the programming language used to build Jenkins?

A) Jenkins is an open source automation server written in Java.

3) What are the features of Jenkins?

A) Jenkins has many features like:

Continuous Integration and Continuous Delivery - As an extensible automation server, Jenkins can be used as a simple CI server or turned into the continuous delivery hub for any project.

Easy installation - Jenkins is a self-contained Java-based program, ready to run out-of-the-box, with packages for Windows, Mac OS X and other Unix-like operating systems.

Easy configuration - Jenkins can be easily set up and configured via its web interface, which includes on-the-fly error checks and built-in help.

Plugins - With hundreds of plugins in the Update Center, Jenkins integrates with practically every tool in the continuous integration and continuous delivery toolchain.

Extensible - Jenkins can be extended via its plugin architecture, providing nearly infinite possibilities for what Jenkins can do.

Distributed - Jenkins can easily distribute work across multiple machines, helping drive builds, tests and deployments across multiple platforms faster.

Are you interested in taking up for Jenkins Certification Training? Enroll for Free Demo on **Jenkins Training!**

4) What is Continuous Integration (CI)?

A) Continuous integration (CI) is a software engineering practice in which isolated changes are immediately tested and reported on when they are added to a larger code base.

Continuous Integration (CI) is a development practice where developers integrate code into a shared repository frequently, preferably several times a day. Each integration can then be verified by an automated build and automated tests.

5) What is a Jenkins Pipeline?

A) Jenkins Pipeline (or simply "Pipeline") is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.

Jenkins Interview Questions And Answers

6) What is a Continuous delivery pipeline?

A) A continuous delivery pipeline is an automated expression of your process for getting software from version control right through to your users and customers.

7) What is Jenkinsfile and what it does?

A) The definition of a Jenkins Pipeline is typically written into a text file called a Jenkinsfile which in turn is checked into a project's source control repository.

8) Can you write a simple Jenkins Pipeline Code for Java?

A) Here is the simple Jenkins Pipeline Code for Java:

```
Jenkinsfile                                    (Declarative                                    Pipeline)
pipeline                                                                                               {
                     agent        {         docker           'maven:3.3.3'            }
                                                       stages                               {
                                                    stage('build')                          {
                                                              steps                          {
                                            sh        'mvn       --version'
                                                                                             }
                                                                                             }
```

```
                                                                                }
}
```

9) What is Declarative Pipeline in Jenkins?

A) Declarative Pipeline is a relatively recent addition to Jenkins Pipeline [1] which presents a more simplified and opinionated syntax on top of the Pipeline sub-systems.

All valid Declarative Pipelines must be enclosed within a pipeline block, for example:

```
pipeline                                                                        {
                    /*        insert        Declarative        Pipeline        here        */
}
```

10) What is the agent directive in Jenkins?

A) The agent directive tells Jenkins where and how to execute the Pipeline, or subset thereof. As you might expect, the agent is required for all Pipelines.

Underneath the hood, there are a few things agent causes to happen:

All the steps contained within the block are queued for execution by Jenkins. As soon as an executor is available, the steps will begin to execute.

A workspace is allocated which will contain files checked out from source control as well as any additional working files for the Pipeline.

Interview Questions on Jenkins

11) What is agent in Jenkins?

A) The agent section specifies where the entire Pipeline, or a specific stage, will execute in the Jenkins environment depending on where the agent section is placed. The section must be defined at the top-level inside the pipeline block, but stage-level usage is optional.

12) What are Parameters in Jenkins?

A) In order to support the wide variety of use-cases Pipeline authors may have, the agent section supports a few different types of parameters. These parameters can be applied at the top-level of the pipeline block, or within each stage directive.

13) What is post?

A) The post section defines one or more additional steps that are run upon the completion of a Pipeline's or stage's run (depending on the location of the post section within the Pipeline). post can support one of the following post-condition blocks: always, changed, failure, success, unstable, and aborted. These condition blocks allow the execution of steps within the post section depending on the completion status of the Pipeline or stage.

14) What are stages?

A) Containing a sequence of one or more stage directives, the stages section is where the bulk of the "work" described by a Pipeline will be located. At a minimum it is recommended that stages contain at least one stage directive for each discrete part of the continuous delivery process, such as Build, Test, and Deploy.

15) What is environment directive?

A) The environment directive specifies a sequence of key-value pairs which will be defined as environment variables for the all steps, or stage-specific steps, depending on where the environment directive is located within the Pipeline.

16) What are triggers?

A) The triggers directive defines the automated ways in which the Pipeline should be re-triggered. For Pipelines which are integrated with a source such as GitHub or BitBucket, triggers may not be necessary as webhooks-based integration will likely already be present. The triggers currently available are cron, pollSCM and upstream.

17) What is input directive?

A) The input directive on a stage allows you to prompt for input, using the input step. The stage will pause after any options have been applied, and before entering the stage`s `agent or evaluating its

when condition. If the input is approved, the stage will then continue. Any parameters provided as part of the input submission will be available in the environment for the rest of the stage.

18) What is Parallel in Jenkins?

A) Stages in Declarative Pipeline may declare a number of nested stages within them, which will be executed in parallel. Note that a stage must have one and only one of either steps or parallel. The nested stages cannot contain further parallel stages themselves, but otherwise behave the same as any other stage. Any stage containing parallel cannot contain agent or tools, since those are not relevant without steps.

19) What is Scripted Pipeline in Jenkins?

A) Scripted Pipeline, like Declarative Pipeline, is built on top of the underlying Pipeline sub-system. Unlike Declarative, Scripted Pipeline is effectively a general purpose DSL [2] built with Groovy. Most functionality provided by the Groovy language is made available to users of Scripted Pipeline, which means it can be a very expressive and flexible tool with which one can author continuous delivery pipelines.

20) What is Flow Control in Jenkins?

A) Scripted Pipeline is serially executed from the top of a Jenkinsfile downwards, like most traditional scripts in Groovy or other languages.