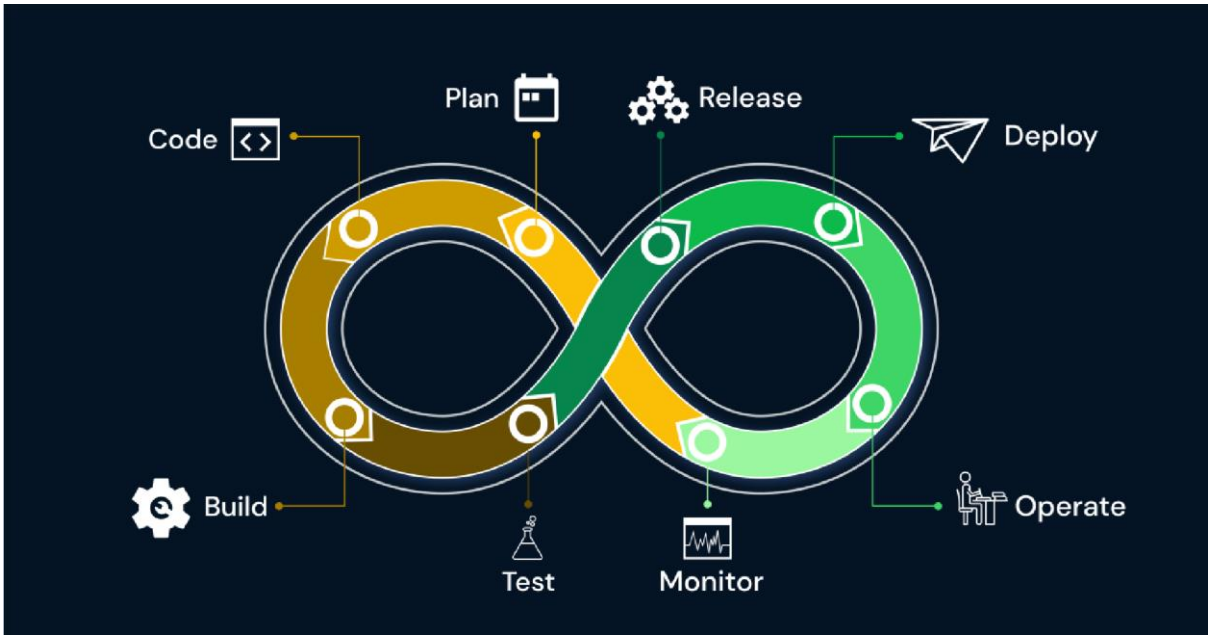
1. What does it mean with “Continuous Integration” used in Jenkins?

- Continuous Integration is a development practice where the codes can be integrated into a shared repository.
- The usual practice of Continuous Integration is to trigger a build as soon as the code is committed to the repository.
- Process before Continuous Integration has many flaws, making the software delivery slow with poor quality. Developers had a tough time locating & fixing bugs.
- Continued Integration overcame these shortcomings by adding verification, detection & consistency to the process.
- The practice uses automated verifications for early & easy detection of code problems.



2. How do you install Jenkins on Linux?

To install Jenkins on your system (Ubuntu), you need to follow these steps

1. Install Java Version 8 or 11 (the ones which have Long Term Support)- Java is must-since Jenkins is a Java-based application. Open the terminal (**Ctrl**+**Alt**+**T**) and type this to install & check the installation.

```
$ sudo apt update
$ sudo apt install openjdk-11-jre
$ java -version
openjdk version "11.0.12" 2021-07-20
OpenJDK Runtime Environment (build 11.0.12+7-post-Debian-2) OpenJDK 64-Bit Server VM (build 11.0.12+7-post-Debian-2, mixed mode, sharing)
```

2. Add Jenkins Repository

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \   https://pkg.jenkins.io/debian-stable binary/ | sudo tee \   /etc/apt/sources.list.d/jenkins.list > /dev/null sudo apt-get update sudo apt-get install jenkins
```

3. Verify Your Installation

```
$ systemctl status Jenkins
```

Once Jenkins is up & running, you will be able to see the Jenkins dashboard.

3. Mention the commands to start Jenkins manually?

To start Jenkins Manually, open your terminal/cmd/command line, go for the Jenkins installation directory and use the following commands.

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

4. What is a Jenkins Pipeline?

Jenkins Pipeline allows developers to define a complete list of events that happen in the code lifecycle. Starting from the build, all the way to testing, monitoring, and deployment. Unlike the traditional software development approach, the pipeline doesn't wait for the whole process to be completed, to look for bugs & errors. Jenkins has been a game-changer for developers. Developers can also use a set of plugins to assist in implementing the ongoing processes as a continuous delivery pipeline. A continuous delivery pipeline facilitates an automated expression that processes the software through Version Control to your users and customers.

Additionally, to implement the pipeline as a code, a Jenkinsfile needs to be present in the project's root repository.

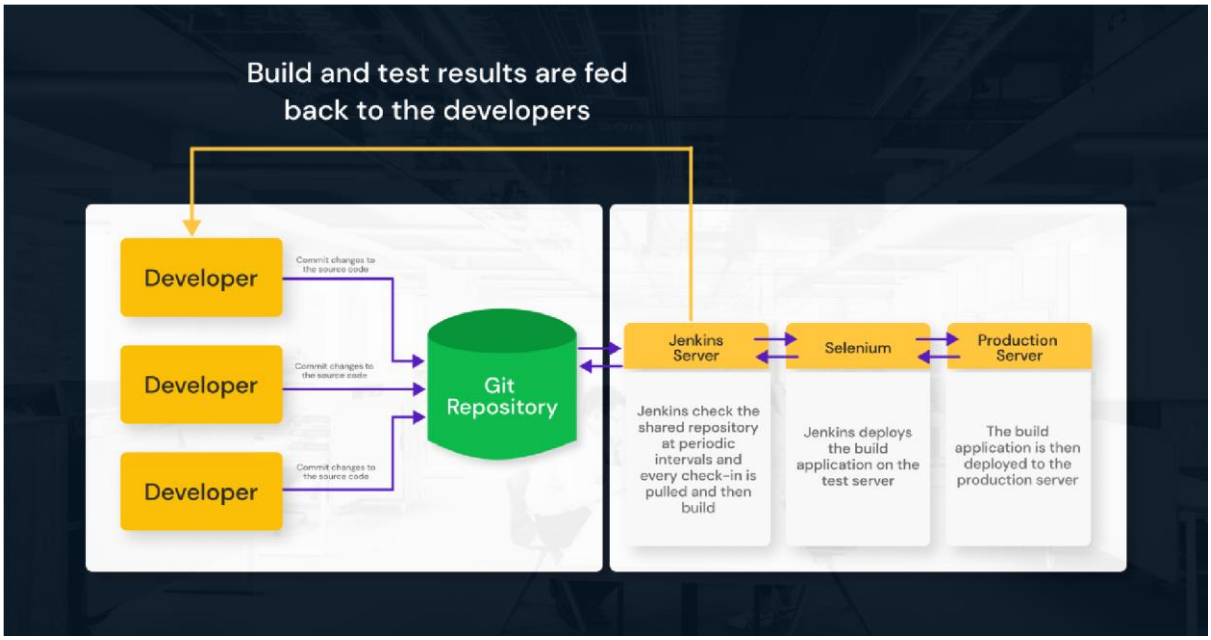
Jenkinsfile support the following syntax: Declarative & Scripted.

5. Elaborate on the process of Jenkins

Now that you've become familiar with the Jenkins Pipeline, let us offer you an insight into what actually happens.

- Initially, a developer commits the code to the repository. Meanwhile, the Jenkins Server verifies the repository at frequent intervals. If a code has been identified, then the new changes will be pulled & tested. So that the Jenkins server can start preparing a new build.
- If the builds fail, then the concerned team will get a notification.
- If it doesn't, the Jenkins server deploys the build-in test server.

- After testing, Jenkins generates feedback that notifies the developer of the test results. The process keeps on repeating.



6. What is a Jenkins File?

A **Jenkinsfile** is a type of text file that stores the complete workflow as. code. You can check the text file into an SCM or on your local system. It is written using the Groovy DSL and it can be created using text/groovy editor or you can access it through the configuration page on Jenkins Instance.

7. Any prerequisites for using Jenkins?

The answer to the above question is pretty straightforward. To use Jenkins, you need:

- A working build script, e.g., a Selenium Script, checked into the repository

- An accessible source code repository, for example, a Git or SVN Repository.

8. What is Jenkins Jobs?

Jenkins Job takes care of any event or process mentioned above. In ease words, A Jenkins job is simply a process that runs on the Jenkins server to provide the above-mentioned functionality. Jenkins offers the option to select from various types of jobs to build your project.

Freestyle

Freestyle build jobs are general-purpose build jobs, which provide maximum flexibility. It can be used for any type of project.

Multi-configuration

The multi-configuration project allows users to run the same build job on different environments. It is used for testing an application in different environments.

Folder

This project allows users to create folders to organize and categorize similar jobs in one folder or subfolder.

Multi-branch Pipeline

This project type lets you implement different Jenkinsfiles for different branches of the same project.

Pipeline

This project runs the entire software development workflow as code. Rather than creating several jobs for each stage of software development, you can now run the entire workflow as one code.

GitHub Organisation

This project scans your entire GitHub organization and creates Pipeline jobs for each repository containing a Jenkinsfile.

9. How does Jenkins fit in with DevOps

DevOps refers to software development practice that sort of blends & synchronizes the process of software development (Dev) with IT operations (Ops). Thus creating the whole development cycle feasible & shorter by constantly building fixes, builds, updates & features. Jenkins plays an imperative role as it helps in this integration by automating the build, test as well as deployment process.

10. Type of Pipelines in Jenkins?

There are basically 3 types:

- Scripted Pipeline
- Declarative Pipeline
- CI/CD Pipeline (Continuous Integration/Continuous Declarative)

11. Explain the various concepts of Jenkins Pipeline?

The pipeline is a user-defined block that contains all the processes of CI/CD such as build, test, deploy, etc. All the stages of the steps are defined in the pipeline.

A **Node** is a machine on which Jenkins runs. A node block is usually scripted in Pipeline Syntax.

A **stage** contains all the series of steps in a pipeline. i.e, build, deploy, and test processes on a stage.

Whereas a **step** is a single task that executes a particular process at a certain time. A pipeline involves various steps defined within a stage block, as illustrated below.

```
pipeline {
   agent any
   stages {
         stage ('Build') {
            ...
         }
         stage ('Deploy') {
            ...
         }
         stage('Monitor'){
            ...
         }
   }
}
```

```
pipeline {
   agent any
    stages {
         stage ('Build') {
            steps {
                  echo 'Running build phase...'
            }
         }
    }
}
```

Enjoying reading Jenkins Interview Questions? You can also visit our [resource guide to crack a Web-developer interview.]

12. Are you aware of other Continuous Integration Tool other than Jenkins? How does Jenkins compare with them?

There are many CL/CD tools available online and the most prominent are:

- bamboo
- TeamCity
- Travis CI
- Circle CI
- ThoughtWorks
- Integrity
- Perforce
- Go

Each one offers its own set of functionalities. For Instance, TeamCity offers great .NET support but it is costly & complex. Travis CI is free to use, just like Jenkins, and does support good documentation. Bamboo also offers efficient and faster builds but it does levy a charge and so on.

13. What is a DSL, Jenkins?

The Domain Specific Language (DSL) in Jenkins enables users to describe jobs using a Groovy-based language. Moreover, there are two parts to Jenkins's "Job DSL/Plugin." The Second one is Jenkins Plugin, whose job is to manage the scripts and updating of Jenkins Jobs.

14. How does Jenkins Identify & authenticate users.

To authenticate a user, there are basically 3 ways.

1. By default, Jenkins store the user data & their credentials in its internal database.
2. Or you can configure Jenkins to use an authentication mechanism, defined by the application server.
3. You can also configure Jenkins to authenticate against the LDAP server.

15. Explain Various ways to schedule a build in Jenkins?

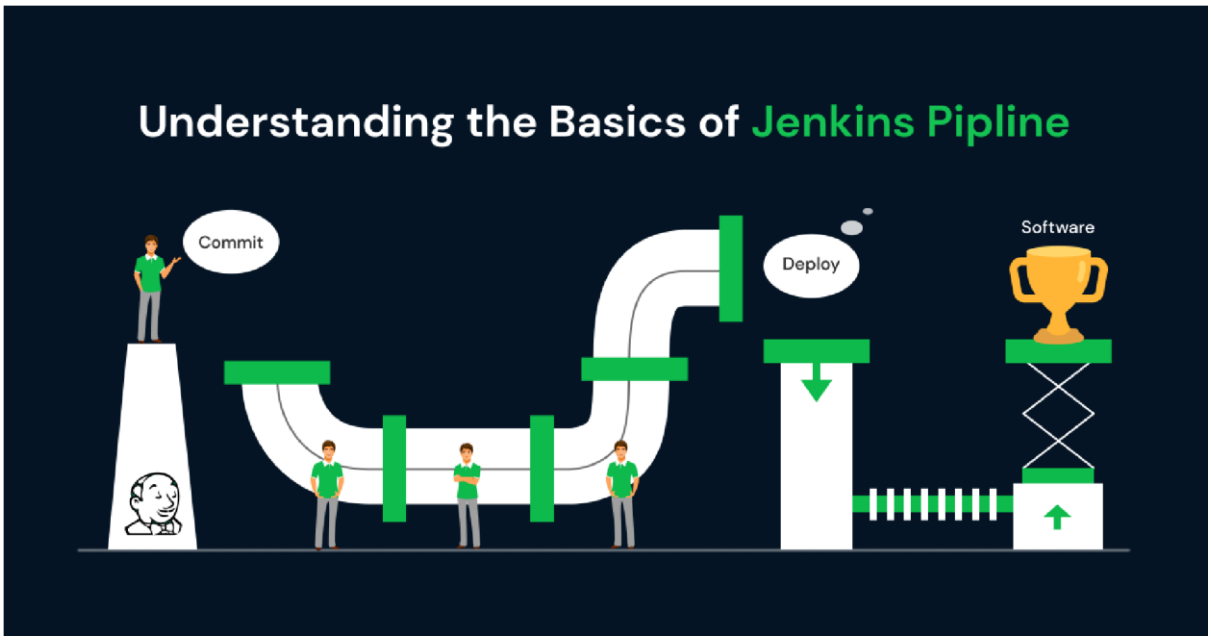
You can schedule a build in Jenkins in the following ways:

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

16. Why you should use Jenkins Pipeline?

- A pipeline can be run in a loop.
- It supports larger projects that may involve a high CPU job, provided the Jenkins infrastructure is scalable enough to support it.
- Since the Jenkins pipeline is written in code, users can use it as a template, modify it and run customized tests and processes.
- Multiple jobs can run parallelly.

- Jenkins Pipeline is robust. The pipeline can automatically be resumed from it might have stopped for any reason.



17. How do you create a multi-branch Jenkins pipeline?

In a Multi-branch Pipeline project, Jenkins automatically discovers, manages, and executes Pipelines for branches that contain a Jenkinsfile in source control.

To create a Multi-branch Jenkins Pipeline

- Open Jenkins Dashboard and create a new item by clicking **“New Item”**
- Enter the project name, and from the options select **“Multi-branch pipeline”**
- Once you confirm, select the repository location, and branch source and add the branch source credentials.

- After saving the project, Jenkins automatically creates new Multibranch Pipelines for repositories
- Then to connect to the GitHub repo, we need the HookURL
- To get this URL from the repository settings, add this HookURL to the Webhooks section
- Once the jobs are created, Jenkins will automatically trigger the build.

18. How do you integrate GIT with Jenkins?

- Once you are at Jenkins Dashboard click on the “Manage Jenkins” button.
- Click on Manage Plugins
- On the Plugin page, select the GIT Plugin.
- Install the Git plugin and restart your Jenkins.
- Once you install the plugins, go to **Manage Jenkins** on your Jenkins dashboard. You will see your plugins listed among the rest.

19. What are some of the default variables in Jenkins? (One of the most asked Jenkins Interview Questions)

- $JOB_NAME — The name that you give your job when it is first set up.
- $NODE_NAME — This is the name of the node on which the current build is running.

- $JENKINS_URL — This is set to the URL of the Jenkins master that is responsible for running the build.
- $BUILD_URL — Indicates the URL where the results of the builds can be found.
- $WORKSPACE — Refers to the path of the workspace

20. How Jenkins can be used for testing in different environments?

- Jenkins checks the Git repository at periodic intervals for any changes made in the source code.
- Each builds requires a different testing environment which is not possible for a single Jenkins server. To perform testing in different environments, Jenkins uses various Slaves as shown in the below diagram.
- Jenkins Master requests these Slaves to perform testing and generate test reports.