**JENKINS**

1. which tool have you used for implement CI/CD ? --> Jenkins

Jenkins is an open source continuous integration tool. It keeps a track on version control system and to initiate and monitor a build system if changes occur.

2. Any alternate tool do you know for CI/CD ? --> bamboo/udeploy/teamcity/tfs...

3. what is Continuous Integration, Continuous delivery and Continuous deployment? get scm --> compile -->test -->build -->static code

analysis -->nexus --> build failure send notification (CI)

Ans:

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

**Continuous deployment** is the process of moving software that has been built and tested successfully into production.

Continous deployment is that code is released continously as part of an automated pipeline

**Continuous delivery** is the important process of delivering the software/Updates to production in smaller increments, ensuring that the software can be released at any time. With this approach of **DevOps**, the team will be always ready on 'Delivering any time' to the production.

4. what type of jobs have you configured in jenkins?

**Jenkins** supports several **different types** of build **jobs**. The two most commonly-used **are** the freestyle builds **and** the Maven 2/3 builds.

5. what are the types of jobs are available in jenkins?

6. what is difference b/w freestyle and pipeline?

A **freestyle project** is used to build a **job** or task. This could be as simple as running tests, building or packaging an application, sending a report, or even running some commands.

**Pipeline** is a workflow with group of events or jobs that are chained and integrated each other in a sequence.

In the Freestyle job everything is executed in the agent, but for the Scripted Pipeline Job, the pipeline code is translated to atomic commands in the master and that are sent to the agents.

7. what is pipeline? -->

**Pipeline** is a workflow with group of events or jobs that are chained and integrated each other in a sequence.

what is jenkins file

**Jenkinsfile** is a text **file** that contains the definition of a **Jenkins** Pipeline and is checked into source control.

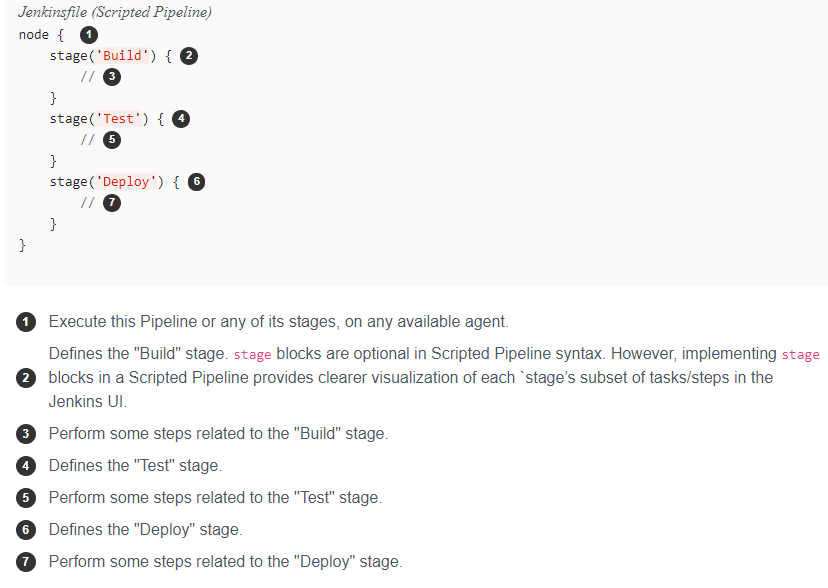
8. what is declarative pipeline?

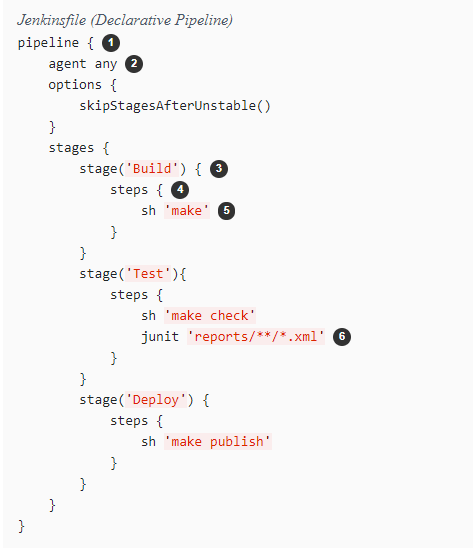
**Declarative pipeline** is a relatively new feature that supports the **pipeline** as code concept. It makes the **pipeline** code easier to read and write. This code is written in a Jenkinsfile which can be checked into a source control management system such as Git.

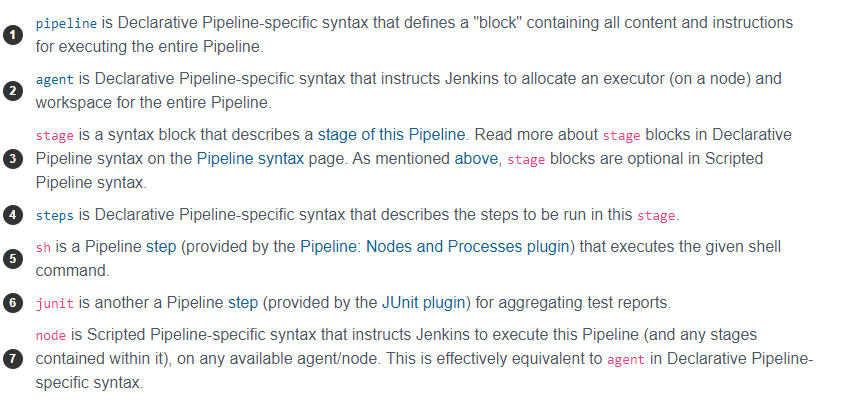
9. what is difference between declarative pipeline and script based pipeline?

* Declarative pipeline is a relatively new feature that supports the pipeline as code concept. It makes the pipeline code easier to read and write. This code is written in a Jenkinsfile which can be checked into a source control management system such as Git.
* Whereas, the scripted pipeline is a traditional way of writing the code. In this pipeline, the Jenkinsfile is written on the Jenkins UI instance.
* Though both these pipelines are based on the groovy DSL, the scripted pipeline strictly uses groovy based syntaxes because it was the first pipeline to be built on the groovy foundation. Since this Groovy script was not typically desirable to all the users, the declarative pipeline was introduced to offer a simpler and more optioned Groovy syntax.
* The declarative pipeline is defined within a block labelled ‘pipeline’ whereas the scripted pipeline is defined within a ‘node’.

10. write the pipeline syntax?







11. what is master/slave architecture? what is the use of master/slave?

**Jenkins Master** and **Slave Architecture** The **Jenkins master** acts to schedule the jobs and assign slaves and send builds to slaves to execute the jobs. It will also monitor the **slave** state (offline or online) and getting back the build result responses from slaves and the display build results on the console output.

**Use** **of** **Jenkins master slave**, developer commits the code and each builds requires a different testing environment which is not possible for a single Jenkins server or In a huge project if multiple commits are happening we need to build parallel. So to **perform testing in multiple environments or to run parallel builds Jenkins uses various Slaves.**

**Jenkins Master**

Your main Jenkins server is the Master. The Master’s job is to handle:

* Scheduling build jobs.
* Dispatching builds to the slaves for the actual execution.
* Monitor the slaves (possibly taking them online and offline as required).
* Recording and presenting the build results.
* A Master instance of Jenkins can also execute build jobs directly.

**Jenkins Slave**

A Slave is a Java executable that runs on a remote machine. Following are the characteristics of Jenkins Slaves:

* It receive the requests from the Jenkins Master instance.
* Slaves can run on a variety of operating systems.
* The job of a Slave is to do as they are told to, which involves executing build jobs dispatched by the Master.
* You can configure a project to always run on a particular Slave machine or a particular type of Slave machine, or simply let Jenkins pick the next available Slave.

12. How many ways we can connect the slaves?

13. How many ways we can provide security for your jenkins server?

14. what is sonarqube ? have you configure ? How you configure ?

15. what type of artifactory repository tool have you used? --> nexus/jfrog/s3 oss -- pro

16. what is the use of artifactory tools?

17. How you declare a variables in pipeline?

Variables in a Jenkinsfile can be defined by using the def keyword.

Such variables should be defined before the pipeline block starts.

When variable is defined, it can be called from the Jenkins declarative pipeline using ${...} syntax.

// Define variable

def myVariable = "foo"

// Print variable

pipeline {

agent any

stages {

stage ("Print variable") {

steps {

echo "My variable is ${myVariable}"

}

}

}

}

18. what is DSL language? domain spcific language?

19. what is upstream/downstream projects? what is the use of it? which scenario you

configure?

The **upstream** job is the one that is triggered before the actual job is triggered. The **downstream** job is the one that is triggered after the actual job is triggered.

20. If you have 10 repositories in github how many jobs you can configure?

21. Dou you have experience to install jenkins?

22. How you configure jdk,maven,gradle...etc?

23. in my environment i have different version for java implementation projects is there

? How you configure multiple jdk's?

24. what are the plugins have you used in your project?

25. How to take backup my jenkins? thin backup plugin

26. what is jenkins Home directory? .jenkins

The Jenkins home directory contains all the details of your Jenkins server configuration, details that you configure in the Manage Jenkins screen. These configuration details are stored in the form of a set of XML files. Much of the core configuration, for example, is stored in the config.xml file.

| **Directory** | **Description** |
| --- | --- |
| .jenkins | The default Jenkins home directory (may be .hudson in older installations). |
| fingerprints | This directory is used by Jenkins to keep track of artifact fingerprints. We look at how to track artifacts later on in the book. |
| jobs | This directory contains configuration details about the build jobs that Jenkins manages, as well as the artifacts and data resulting from these builds. We look at this directory in detail below. |
| plugins | This directory contains any plugins that you have installed. Plugins allow you to extend Jenkins by adding extra feature. Note that, with the exception of the Jenkins core plugins (subversion, cvs, ssh-slaves, maven, and scid-ad), plugins are not stored with the jenkins executable, or in the expanded web application directory. This means that you can update your Jenkins executable and not have to reinstall all your plugins. |
| updates | This is an internal directory used by Jenkins to store information about available plugin updates. |
| userContent | You can use this directory to place your own custom content onto your Jenkins server. You can access files in this directory at *http://myserver/hudson/userContent* (if you are running Jenkins on an application server) or *http://myserver/userContent* (if you are running in stand-alone mode). |
| users | If you are using the native Jenkins user database, user accounts will be stored in this directory. |
| war | This directory contains the expanded web application. When you start Jenkins as a stand-alone application, it will extract the web application into this directory. |

The **workspace directory** is where **Jenkins** builds your project: it contains the source code **Jenkins** checks out, plus any files generated by the build itself. This **workspace** is reused for each successive build

27. how to deleted old builds automatically?

28. how to configure multiple environment deployment?

29. Where is Jenkins build history stored?

**Jenkins** stores the configuration for each job within an eponymous directory in jobs/ . The job configuration file is config. xml , the **builds** are **stored** in **builds**/ , and the working directory is workspace/ .

30. How do I stop Jenkins?

**Execute the following commands respectively:**

1. To stop: jenkins.exe stop.
2. To start: jenkins.exe start.
3. To restart: jenkins.exe restart.

31. How do I restart Jenkins from GUI?

**Restart Jenkins** from Web Interface. Navigate to: [jenkins\_url]/safeRestart. – This will **restart Jenkins** after the current builds have completed.

What is DevOps lifecycle?

**DevOps Lifecycle**. **DevOps** defines an agile relationship between operations and Development. It is a process that is practiced by the development team and operational engineers together from beginning to the final stage of the product.

**Linux Commands to work on Jenkins**

Ps -ef | grep java - to check whether the Jenkins is running or not.

Systemctl stop jenkins - to stop Jenkins

Systemctl start jenkins - to start Jenkins

Wget (Jenkins url) – to download the updated Jenkins

Yum upgrade (Jenkins file) – to upgrade in another method

### How can you move or copy Jenkins from one server to another?

Follow these steps to move or copy Jenkins from one server to another:

* First, copy the related job directory and slide a job from one installation of Jenkins to another.
* Make a copy of an already existing job by making clone of a job directory by a different name.
* Renaming an existing job by rename a directory.