**JENKINS**

To install Jenkins please follow the steps in below link:

<https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-20-04>

Which type of jobs you have created?

1. Freestyle: lets you create general-purpose build jobs with maximum flexibility.
2. Pipeline
3. Multibranch pipeline
4. Github organization
5. **How to install the plugins?**

We will download the plugin from the Jenkin update server (updates.jenkins.IO)

We will add the plugin from the manage Jenkins, manage plugins in advance tab we are going to upload a file and install it without restarting.

1. **How we are going to secure Jenkins or how do you secure Jenkins?**

We used matrix-based security it provides checkboxes, whichever boxes we check users will have those permissions. We can provide permissions by selecting check boxes.

Manage Jenkins ->Configure global security->Select matrix-based security -> Add users -> Select the checkboxes

**Note: 3 types of security**

1. Matrix based
2. Project based
3. Role based

**Master Slave:**

Why are slave machines required?

Master slaves are used to distribute the load to different servers. And also, we can run specific jobs on specific servers.

Eg: If I want to compile a “c source code” on the server which provide a “c environment” only.

**Build triggers:**

1. Build Periodically

It will trigger the build based on the time, whether commit or no commit

1. Poll SCM

It will trigger the Jenkin job based on the commit, if there are any commits within a specific time, it will trigger Jenkins Job.

1. Webhooks

It will trigger the job based on the commit and push in the git repo.

**Pluggin Installation:**

CVS, upstream, downstream, quality gate, artifactory, blueocean, selenium, gearman, publish over CIFS, parameterized plugin, docker pipeline, sonar qube, docker many more.

**\*Parameterized plugin:**

It is used to pass parameters to the Jenkins job. I configured parameterized plugins for the deployment job. The deployment job has 2 parameters

1. Server name
2. Build number

These values will be passed to the deployment script inside the Jenkins job.

**\*Gearman Plugin (high availability plugin)**

It is a high-availability plugin if jenkins master goes down, Jenkins will be up and running automatically on other servers. Jenkins will be highly available.

Gearman plugin will allow us to add other server details (2nd master). When Jenkins is running on first master both servers will be in sync. As soon as Jenkin master goes down Jenkins will be up and running on another server.

**\*Publish over CIFS:**

Used to copy files from linux to Windows

**\*Sonar qube**

Sonar cube is used to check the quality of the source code. If the quality of the source code is more than 70% then we consider the quality of the source code to be good.

We integrated sonar qube with a Jenkins sonar qube works as a vulnerability tool in terms of scanning the source code.

The unit test case results would be uploaded into sonar qube, upon the unit test case results we will get Metrix from the quality profile and quality gates.

**How do you trigger jobs based on commit?**

We use Webhooks.

**\*\*\*How do you take Jenkins’s backup? \*\*\***

We need to take back up from the path /var/lib/Jenkins. We take Jenkins’s backup and store it in a separate Git repository. We have written a script to do this automatically and schedule in Crontab.

The script will push Jenkins’s job, and Jenkins configuration files to the git repository.

**\*\*\*CICD Pipeline (end-to-end process) \*\*\***

Once the developer completes his work, he will raise a PR (pull request) as soon as the developer raises the PR local build job will get triggered. Once the build is successful, the reviewer or approver will merge the code to the central repo by clicking on the merge button (the merge button will not be enabled in the pull request if the local build fails). As soon as merges to the central repo Jenkins’s pipeline job will get triggered.

It has 3 stages.

1. Build stage: It will checkout the source code from GitHub to Jenkins’s local workspace. It compiles the source code and generates the build. Once the build stage is a success it will automatically trigger the deploy stage.
2. Deploy stage: It will deploy the build to the QA server and runs the sanity test(bvt). Once the deploy stage is a success it will trigger the test stage automatically.
3. Test stage: Here we run the test cases which are given by the testing team. We configure these test cases and trigger them automatically. If there are any failures, they will be taken care of by testers.

**Jenkins pipeline (Jenkins file):** Jenkins file is where we put all our job configuration to achieve multiple instance execution in a single pipeline job (by default it will take one executor unless we configure more executors)

**Declarative pipeline:** [Pipeline Syntax (jenkins.io)Pipeline Syntax (jenkins.io)](https://www.jenkins.io/doc/book/pipeline/syntax/)

2 types of Jenkins pipeline:

1. Declarative pipeline:
2. Agent: It is a block to define which jobs to run in which machine.

We can define 4 types of agents:

1. Any- It is defined at the pipeline level. It can be run in any slave based on the load.

Syntax: agent any

1. None- If I define an agent as none at the pipeline level, then it is mandatory to define an agent on each and every stage.
2. Docker- Here we are giving the image name

Syntax: agent {image name}

1. Label- If I want to run jobs on a particular slave then we will use the label.

Syntax: agent { label ‘slavename’ }

1. Scripted pipeline

Graphical user interface, text, application, email

Description automatically generated

**What Is the path to change the port in Jenkins?**

Etc/default/Jenkins

**What Is Groovy? Types of groovy**

Groovy is used to define a new pipeline as a script and there are two types of Groovy

Declarative pipeline

Scripted pipeline

**What is the difference between declarative & scripted pipelines?**

Declarative pipeline:

It is a blocked label pipeline & it is a simple & more optimized groovy syntax

It is a relatively new feature to support pipeline code.

The code is written in a Jenkin file & which can be checked in source code management.

Scripted pipeline:

It is the traditional way of writing code.

The code is written in a Jenkin file on the Jenkins UI interface.

The scripted pipeline was not typically desirable.

**How are you going to manage Jenkins in master? OR what will do if Jenkin’s master goes down?**

We are using primary & secondary masters and whenever the primary master goes down then the secondary master will be up & running,

Both the primary & secondary masters are controlled by JENKIN OPERATION CENTER.

**What is an artifact?**

It is code ready to conduct deployment.

Sample jenkins pipeline [Comprehensive Guide To Jenkins Declarative Pipeline [With Examples] (lambdatest.com)](https://www.lambdatest.com/blog/jenkins-declarative-pipeline-examples/)