**Jenkins**

What is CI/CD?

CI/CD stands for continuous integration and continuous deployment/delivery.

Whenever developer writes the code, we integrated all that code of all developers at that point of time and we build, test and deliver or deploy to the client this process is called as

CI/CD.

Jenkins helps us to achieve this CI/CD process.

Because of CI error will be reported fast and get rectified fast so the entire software development happens fast.

What is Jenkins?

Jenkins is an open source CI tool written in Java that runs on any OS.

It is free with huge community support.

Why Jenkins is so popular?

Because of strong community support and lots of plugins available.

Jenkins automates the entire software development life cycle.

Jenkins runs on 8080 port.

Jenkins paid version is Hudson.

CI Tools

Bamboo, Travis CI, Buildbot and Jenkins.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Workflow of Jenkins

We can attach Git, Maven, Selenium and artifactory plugins to Jenkins.

Once developer push code in GitHub Jenkins pull that code and send to Maven for build.

Once build is done Jenkins pull that code and send to Selenium for testing.

Once testing is done then Jenkins pull that code and send to artefactory as per a requirement and so on.

We can also deploy with Jenkins.

Jenkins architecture is a master’s slave architecture.

Advantages of Jenkins

1) Free and open source

2) Lots of plugins available in community.

3) You can write your own plugin

4) We can attach slaves (nodes) to Jenkins master. Jenkins master instruct slaves to do job. If slaves are not available Jenkins master himself do the job.

5) Jenkins also behave as crone server replacement i.e. can do job scheduling.

6) It can create labels i.e. it can assign particular task to particular slaves.

What is Integrate?

Combine all code written by developers.

What is Build?

Compile the code, code review, unit testing, integrate testing and make executable package is Build. (Combined code and make executable package)

What is Archived?

Store executable package in artifactory.(End produced)

What is Delivery?

Handling product to client.

What is deploy?

Installing product in client machine.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Lab (Windows)**

Download Git from Git website and install and configure (set username and email id).

Download Java from java website and install and configure.

Download Mavan from Mavan website and install and configure.

Download Jenkins from Jenkins website and install and configuration from internet localhost:8080 .

**Lab (Linux)**

1) First create is Linux EC2 instance with SSH, HTTP and 8080 port.

2) Update EC2 Instance yum update -y.

3) Install java package: yum install java-1.8\* -y.

4) Now set path of java

Find path

#find /usr/lib/jvm/java-1.8\* | head -3

#vim .bash\_profile

JAVA\_HOME = <path>

PATH=$PATH:$HOME/bin:$JAVA\_HOME

Restart server

5) Now download Jenkins

From Jenkins website provided command

#wget ....

To set key

#rpm ...

Install Jenkins

#yum install jenkins -y

6) Start service

#systemctl start jenkins.service

#systemctl enable jenkins.service

Now access using web

Ec2Publicip:8080

Now unlock using command

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Plugin**

Plugins provide integration point to other tools (Like connector between Jenkins and other tools).

Plugins are small library that add new ability to Jenkins.

Jenkins comes up with some default plugins.

Click on manage Jenkins.

Click on plugin manage.

Here you can install updated plugin

**Global tool configuration**

Provide address of other tools like Git,  Maven, Java to Jenkins which are already installed in your machine.

When you run any job and job required any tool them Jenkins can make use of those already available tool.

If you not provide address of those tools then Jenkins will install those tools and after use it will uninstall the tool

Click on manage Jenkins.

Click on global tool configuration.

Provide any name and path.

Unchecked install automatically.

Click on same.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Creating maven job**

Provide item/job name

Select maven job

Click on source code management and select git

Provide git repository URL.

Click on build

Provide maven build command in goals and options tab (mvn clean package or clean package)

Click on save and run job.

When we run the job it will pull the code from github repository and build that with the help of Maven.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Schedule Jobs**

After certain amount of time if you want to perform any task then we go for jobs scheduling.

We have two types of scheduled Job

**1) Build periodically**

After every minute or mention time Jenkins pull code from GitHub repository and build.

**2) Poll SCM**

After every minute or mention time if new changes are added in Central repository then only Jenkins pulled code and build.

Select respective check box in Build Trigger.

**Related / Linked Jobs**

We are having two types of Linked job

**1) Downstream**

After first job complete second job start,

We need to mention second job name in first job.( You can mention any number of job by using comma , )

Post-build action

Select build other projects

Provide the second job name.

**2) Upstream**

After first job complete second job start,

We need to mention first job name in second job.

Select build after other projects are build check box in Build Trigger and mentioned first job name.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**View**

You can create separate view for your job. eg. Maven related Job in one view. Git related job in one view

Deleting view doesn't mean that to delete job.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**User management**

In company Jenkins installed in one server and everyone accessing that server being Norma user.

Creating user in Jenkins

Click on manage Jenkins

Click on manage user

Create user

Now to give limited access to user first revoke complete accept then give required access.

For that we need one plugin - Role-based authorization strategy.

Install above plugin.

Now

Click on manage Jenkins

Click on configure global security

Select Role-Based Strategy in Authorization Tab.

Click on save

Now created role and attached role to users. They will get limited as well as full access for few things.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Master Slave**

Jenkins is master doing lots of hard work so we can create slave to do some task.

We can create in same server (EC2) or in different server (EC2).

Create slave

Click on manage Jenkins.

Click on manage nodes a d clouds.

Click on new node

Provide node name any (slave1)

Select permanent agent.

Click on ok

Provide description

Prove of Executor (how may task slave can perform at one time (max 2))

Provide root directory just like master workspace.

Select launch method : launch agent via execution of command on master.

Provide launch command as agent location.

Click on save.

**Label**

Using label we can assign specific job to specific slave.

Provide label in slave.

Save.

Select restrict where this project cab be run.

Provide label name.

Save.