# JENKINS

- Jenkin is an open-source project written in Java that runs on windows, macOS and other unix like os. It is free, community supported and might be your first-choice tool for CI.
- Jenkins automate the entire s/w development life cycle.
- Jenkins was originally developed by sun microsystem in 2004 under the name hadson.
- The project was later named Jenkins when oracle bought microsystem.
- It can run on any major platform without any compatibility issues.
- Whenever developer write code, we integrate all that code of all developers at that point of time and we build test and deliver/ deploy to the client. This process is called as CI/CD.
- Jenkins helps us to achieve this.
- Because of CI now bugs will be reported fast and get rectified fast. So that entire software development happens fast.

Workflow of Jenkins:

- We can attach git, maven, selenium and artifactory plugins to Jenkins.
- Once developers put codes in github Jenkins pill that code and send to maven for build.
- Once build is done, then Jenkins will pull that code and send to artifactory as per requirement and so on.
- We can also deploy with Jenkins.

Advantages of Jenkins:

- It has lots of plugins available.
- You can write your own plugins.
- You can use community plugins.
- Jenkins is not just a tool. It is a framework i.e you can do whatever you want. All you need is plugins.
- We ca attach slaves (nodes) to jenkins master. It instructs other (slaves) to do job. If slaves are not available, Jenkins itself does the job.
- Jenkins also behave as crone server replacement i.e can do scheduled task.
- It can create labels.

**CI/CD Project:**

Go to google chrome – search ‘git download’ – download 2.31.1 for windows – click to download.

**Git download and install:**

Open the download file preamble – c:\programfiles\gir – select components – select start menu folder – choosing default editor vim – let git decide – git from the command line and also from the 3rd party s/w – use the open SSL library – check as-is, commit unix-style line encodings – use minty – choose default behavior – git credentials manager core – enable file system caching – install.

Go to command prompt in laptop

- Git config –global username ‘username’
- Git config –global user.email ‘mailid’
- Git config –global –list

**Download and Installation of JDK16:**

Go to google chrome – search ‘java development kit download’ – download java SE development kit 16 for windows x64.

Run and follow the steps to install.

Now go to C drive – program files – java jdk16 – select path and copy it

Search ‘edit system environment variables’ in laptop – go to use variables – new

Variable name- JAVA_HOME

Variable value- paste the path here

Now go to system variable – new

Variable name – JAVA_HOME

Variable value- paste the path here

Now go inside programfiles – bin – paste the path

Again, go to ‘edit system environment variables’ – system variable – path – new – paste path

Now verify in command promt

C:\users\home\echo% JAVA_HOME%

o/p- c:\ programfiles\java\jdk-16

**Maven download and install:**

Go to google chrome – search maven.apache.org – download – binary zip archive

Extract files – c:\devtools

Go to c:\ - devtools – apache-maven – copy the path

Now search ‘edit system environmental variables’ – system variable – new

Variable name – M2_HOME

Variable value - paste the path here

Now go inside apache maven folder – bin – copy path

Now again go to ‘environmental variable’ – system variable – path – new – paste path

Now open command prompt

C:\users\home > mvn -version

C:\users\home > echo % M2_HOME%

Now restart the laptop

**Jenkins download and install:**

Go to google chrome – Jenkins.io – download – select LTS – windows – download

Open download file – run and install

After installation it automatically open as local host: 8080

Unlock the page by using password

Now install suggested plugins

Ask for username and password

Username – admin

Password – admin123

Email address- technicalgutgu99@gmail.com

Save and continue

Start using Jenkins

Plugins: plugins are small libraries that add new abilities to Jenkins and can provide integration points to other tools.

Go to google chrome – localhost:8080 – login

Go to manage Jenkins on left side of Jenkins dashboard – manage plugins – available – select maven integration and green balls – install without restart

Go to new item – maven project

Now go to manage Jenkins – global tool configuration

Go to add JDK

Uncheck the install automatically option

NAME – JAVA

JAVA_HOME – c:\programfiles\java\jdk

Now go to maven

Name – MAVEN

MAVEN_HOME – c:\devtools\apache-maven

**Maven Project (by maven):**

Go to https://github.com/technicalguftgu/time-tracker

Click on time tracker repo

'fork' to copy this repo

Sign-in into your github account

Click on time-tracker repo

Clone

Go to c drive

Git clone <url of time-tracker repo>

Cd time-tracker

C:\time-tracker > maven clean package

**Maven Project (by Jenkins):**

Now go to Jenkins – new item – entername -> mymavenproject

Then select maven project – ok

Source code management – git – repo url

Build option – root POM – pom.xml

Goals and options – clean package – save

Go to Jenkins home page – click on mymavenproject – build now

**Scheduled Project:**

Click on any project – configure – build triggers – build periodically - * * * * * - save

Can see automatic builds after every 1 min.

You can manually trigger build as well.

**Source Code Polling (Poll SCM):**

Now go to Jenkins home page – go to mymavenproject – configure

Now go to build trigger

Enable poll SCM

Schedule * * * * * - save

Now go to github account – do some changes in README.md – commit changes.

You can see after 1 min, it builds automatically.

**User Management:**

go to Jenkins homepage – manage Jenkins – manage user

create two users – Bhupinder and Rajput

now login as Bhupinder

{by default you have all the permissions}

Login as ‘admin’ again

Go to manage Jenkins – manage plugins – search ‘role-based authorization strategy’ – install without restart.

Go to Jenkins homepage – manage Jenkins – configure global security – select role-based strategy – save.

Login as ‘bhupinder’ – access denied

Now attach permission

Go to Jenkins – manage Jenkins – manage and assign role – manage roles

Role to add – employee

Go to item project – add developer and tester [pattern – dev* test*]

Then assign roles

User/group to add Bhupinder and Rajput.

**How to install Jenkins on Ubuntu:**

- Login into AWS account and create one ubuntu instance.
- Access it through putty and login as ‘ubuntu’.
- Use given commands
  # sudo apt-get update
  # sudo apt-cache search openjdk
  #sudo apt-get install openjdk-8-jdk
  #java -version
  #sudo vi /etc/apt/source.list
  Paste at the bottom
  deb https://pkg.jenkins.io/debian-stablebinary/
  #sudo apt-get update
  # wget -q -o ---
  # sudo apt-cache search Jenkins
  # sudo apt-cache Madison Jenkins
  # sudo apt-get install Jenkins -y
  # sudo service Jenkins status -q
  Copy public ip from AWS and paste in chrome url – public:8080
  Go to instance #sudo cat /var/lib;initialpassword