**JENKINS**

Jenkins with git

Now launch one instance In AWS

Connect this instance through putty

sudo -i

**Install Jenkins**

Using below command.

sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

# yum install Jenkins

# sudo yum remove java-1.7.0-openjdk.x86\_64

# sudo yum install java-1.8.0-openjdk.x86\_64

# service jenkins start

Now open inbound 8080 port in customtcp, open port 8080,set any where Aws Instance.

Now copy and paste instance public IP address in Google or any browser with 8080 port

Ex: 123.456.85.6:8080

Now Jenkins is connect

Now get key from this location

cat /var/lib/jenkins/secrets/initialAdminPassword

Now you get password

Just copy and paste in

Now Jenkins is ready click get ready

Select suggested plugins.

Now you get Jenkins Dashboard.

**1)Now Started with Jenkins with GIt**

Step1:create one job ,with free style job apply and ok

Step2: now job configuration settings for your created job

**Genearl section**

Select : GitHub project

Project url: <https://github.com/deepueng40/git-fundamentlas>

**Source Code Management**

Select git:

Repositories

Repository URL: https://github.com/deepueng40/git-fundamentlas.git

**Build Triggers**

Select : Build periodically

Set : \* \* \* \* \*

**Build**

Select Execute shell

echo “hi”

Apply And save

Now Build This Job

Now Automatically Builds Are Runing for every 1 min

Because your are select Build periodically option:

2) Now for **Poll SCM**

Go and login your Github Account and select any Repository.

Now come to jenkins

Step1:create one job ,with free style job ok

Step2: now job configuration settings for your created job

**General section**

Select: GitHub project

Project url: [\*\*\*\*\*\*\*\*\*Give](https://github.com/deepueng40/git-fundamentlas) your git hun url\*\*\*\*\*\*\*\*\*\*\*

**Source Code Management**

Select git:

Repositories

Repository URL: \*\*\*\* Give Your Repository URL \*\*\*\*\*\*

(Select any Repo: here clone option is there with https:)

**Build Triggers**

Select : **Poll SCM**

Set : \* \* \* \* \*

Apply And save

Now Automatically it clone and Build this first build

Now second build now Not build Because poll scm will check if any modification done in your Github then only build are running. Other wise not build.

Go to your github and select any repository select any file then edit and give commit ok

Now Automatically 2nd build is running bcz your modify code in GITHUB.

Differnce B/W Build periodically And Poll scm

Pollscm is for any modification done you github then only build

Build periodically is Automatically done .as per set your cron job.

**3) Maven Project**

Goto below site for java instalation

<https://tecadmin.net/install-java-8-on-centos-rhel-and-fedora/>

run below command

# cd /opt/

# wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/8u141-b15/336fa29ff2bb4ef291e347e091f7f4a7/jdk-8u141-linux-x64.tar.gz"

# tar xzf jdk-8u141-linux-x64.tar.gz

**Install Java 8 with Alternatives**

# cd /opt/jdk1.8.0\_141/

# alternatives --install /usr/bin/java java /opt/jdk1.8.0\_141/bin/java 2

# alternatives --config java

Enter to keep the current selection[+], or type selection number: 2

If you press enter (or) 2.

Now go root directory

# cd

# java –version

Now run this four commands

# alternatives --install /usr/bin/jar jar /opt/jdk1.8.0\_141/bin/jar 2

# alternatives --install /usr/bin/javac javac /opt/jdk1.8.0\_141/bin/javac 2

# alternatives --set jar /opt/jdk1.8.0\_141/bin/jar

# alternatives --set javac /opt/jdk1.8.0\_141/bin/javac

Now Path setup

**# export JAVA\_HOME=/opt/jdk1.8.0\_141**

**# export JRE\_HOME=/opt/jdk1.8.0\_141/jre**

**# export PATH=$PATH:/opt/jdk1.8.0\_141/bin:/opt/jdk1.8.0\_141/jre/bin**

For conveformation java installed or not just type**:**

**# javac**

**#Java -version**

**Now Maven Installation**

Go this below link

[**https://tecadmin.net/install-apache-maven-on-centos/**](https://tecadmin.net/install-apache-maven-on-centos/)

**execute the below commands**

# cd /opt

# wget http://www-eu.apache.org/dist/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.tar.gz

**#** sudo tar xzf apache-maven-3.3.9-bin.tar.gz

**#** sudo ln -s apache-maven-3.3.9 maven

## Setup Environment Variables for maven

sudo vi /etc/profile.d/maven.sh

when you exeute the above command by default vi editor open write below two commands then save and quit(!wq)

export M2\_HOME=/opt/maven

export PATH=${M2\_HOME}/bin:${PATH}

The below command for execute the above script

# source /etc/profile.d/maven.sh

Now Check maven version .below is command

**# mvn -version**

**Now goto Jenkins Dashboard.**

Go to managejenkins🡪global tool configuration

Now set for jdk(for java)

Name:jdk

JAVA\_HOME= /opt/jdk1.8.0\_141(this getting from go to below way)

# cd opt/ jdk1.8.0\_141

# Pwd

# /opt/jdk1.8.0\_141

**Set maven path**

Name :maven

Maven\_home: /opt/maven

Now save.

Create one job with name Maven.select freestyle job now ok.

Now goto job configuration.

**Genral**

GitHub project

Project url: <https://github.com/khasim1227/webapp>

**Source Code Management**

Select Git:

Repositories

Repository URL: <https://github.com/khasim1227/webapp.git>

**Bulid**

Select :Invoke Top-level maven targets

Maven Version:when we set maven path in gobletool configuration for maven that time your given name for maven.

Select that Name(maven)

Goals: clean package

Apply And save

Now Build this Job

Now Go and check workspace you will get target directory.

Inside target directory you see war file.

**4) Webhooks**

1)Now create a new job :web hooks,select free style job🡪ok

Now go to job configuration:

General:

GitHub project

Project url: <https://github.com/khasim1227/Explore-Devops>

Source Code Management

Select Git

Repositories: <https://github.com/khasim1227/Explore-Devops.git>

Apply and save

**2**) Now Go to managejenkins-🡪system configurations

**GitHub**

Select Git hub server: add git hubserver

Click Advanced(hint right side notebook with pen images is available click that Advanced)

Click Override Hook URL : Specify another hook url for GitHub configuration

Shared secret: Add

Now enter user name and password for github accounts.

Click Additional actions :Now you get convert login id and password to token

From credentials

|  |  |
| --- | --- |
|  | Credentials :select your github username and password.  Now click **Create Token Credentails**  **Now token is created.**  **Now to** GitHub    GitHub Servers  Credentials click add --🡪 enter your git hub user id and password.  Click test connection  You get http://52.53.235.217:8080/static/4594b9ff/images/none.gifCredentials verified for user xxxxx, rate limit: 4996 |

**3) Now go to your github in google**

**And go to repositories**

**Select repositories for which repositories for you build job.**

**In this repositories -🡪**see **settings** options **.Now click this options**

It goto other page here left side

3.1)**integrations & services** options there .select that

**Add service:serch github plugib -🡪select that.** Then click **add service** button.

3.2) Now Go to webhooks.

Add webhooks

Payloadurl: <http://52.53.235.217:8080/github-webhook/>

(this Url you will get from manage Jenkins-🡪configure system-🡪git hub🡪right side notebook with pen image click Advanced--🡪see override web hook url🡪copy that paste in above mention place )

**Click Add Webhook Button.**

Now Go to job in Jenkins for webhooks

**Go to -🡪Build trigger**

**Select:** GitHub hook trigger for GITScm polling

**Now Apply And save.**

**Observation:**

**When you edit any file in this Repository will get build automatically in Jenkins dashboard for this Job.**

**5) How To add nodes in Jenkins**

Now to aws Create one Instance give Tag Name is slave1.

Now you have two nodes

1)Jenkins

2)slave1

Now steps to create Nodes in slave.

Goto manage Jenkins-🡪select manage nodes-🡪 click new node

Node node: Node1

Slect perment agent ok.

Now node configuration

Name:**node1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Remote root directory: /home/ec2-user  Labels: slave1  Usage:use node this as much as possible  Launch method:lunch slave agent via ssh.  Host:private ip address of aws slave1 instance.  Give   |  |  | | --- | --- | |  | Credentials Add Jenkins--🡪 Now redirect another page Add Credentials Kind: user name with key  User name:ec2-user(as per your select aws machine image)  Private key: select enter directly  Past **pem key**  **Ok** | |
| Host Key Verification Strategy | :select 3 option(manually trusted key) |

Availability:keep this ajent as much as possible.

Now save

Select node1 and lunch.

If it is any error(like X mark)

Set this commands in slave1 instance

Now connect slave1 instance through putty

Now install jdk 1.8

Below is link

Below is commands

# cd /opt/

# wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/8u141-b15/336fa29ff2bb4ef291e347e091f7f4a7/jdk-8u141-linux-x64.tar.gz"

# tar xzf jdk-8u141-linux-x64.tar.gz

# cd /opt/jdk1.8.0\_141/

# alternatives --install /usr/bin/java java /opt/jdk1.8.0\_141/bin/java 2

# alternatives --config java

Enter to keep the current selection[+], or type selection number: 2

If you press enter (or) 2.

Now go root directory

# cd

# java –version

Now Link once slave1in node dashaboard.

Now it is in online

**II.(Continues Deployment).**

**Install Tomcat in Linux machine.**

Go to <http://tomcat.apache.org/>

Down load option is there for version wise.

Select download tomcat 8.0

#### Binary Distributions

Core

Select tar file.right click copy location

Now come to aws Jenkins instance

Go to opt folder

# cd /opt

# wget <https://www.apache.org/dist/tomcat/tomcat-8/v8.5.20/bin/apache-tomcat-8.5.20.tar.gz.asc>

Now un tar this below is command

# tar -xvf apache-tomcat-8.5.20.tar.gz

# ls

# rm -rf apache-tomcat-8.5.20.tar.gz

# ls

# cd apache-tomcat-8.5.20/

#pwd

# ls

# cd bin/

# ls

# ./startup.sh(for start tomcat)

# ./shutdown.sh(for shutdown tomcat)

# cd ..

# ls

# cd conf

# vi server.xml

change container port 8080 to 9090

:wq!

# cd bin/

# ./startup.sh

Now go to aws instance and open 9090 port set any where

Just copy the public ip address of Jenkins server paste with :9090

Now tomcat server also running in your browser.

Now login into tomcat purpose.

Goto

# cd/opt/ apache-tomcat-8.5.20/

# ls

# cd conf/

# vi tomcat-user

<role rolename=”manager-gui”/>

<role rolename=”manager-status”/>

<role rolename=”manager-scripts”/>

<role rolename=”manager-jmx”/>

<role rolename=”viewer”/>

<role rolename=”admin”/>

<role rolename=”tomact”/>

<user username=”manager” password=”manager” roles=”manager-gui”/>

<user username=”viewer” password=”viewer” roles=”admin,tomcat,manager-gui,manager-script”/>

Acces denied error:403 error resolve.

Cd :webapps

# ls

# cd manger

# ls

#cd META-INF/

# ls

#content.xml

Just comment below lines.

<!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"

allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->

now come to bin folder

Now run ./shutdown

And then

Now start ./startup