**SIMPLE DEVOPS PROJECT 1**

**CI/CD THROUGH JENKINS**

**Steps 1:**

**Installing Tomcat server**

1.Creating a new instance in aws as **Web Server**

2.Copy the IPV4 address and go and open the putty and configure the ip address with the respective .ppk file.

3.Then login it with the **username: ec2-user.**

4.Then move to the root directory by giving the command **sudo su**

5.Then install java in it using the command **yum install java-1.8\***

6.Then for a confirmation check the version of java using the command **java -version**

7.Then download the tomcat 8 version by copying the url file and download it using the command **wget https://downloads.apache.org/tomcat/tomcat-8/v8.5.79/bin/apache-tomcat-8.5.79.tar.gz**

8.Unzip the tomcat file using the command **tar -zvxf apache-tomcat-8.5.79.tar.gz**

9. Then get into the extracted folder using the **cd** command

10.Then to start the server go to the bin folder using the command **cd bin**

11. Then check whether the tomcat is running or not before starting it , using the command **ps -ef | grep tomcat**

12.Use the **ls -lrt** command to check all the file permissions that are under the bin folder

13.Then we are changing the permissions for the startup.sh and shutdown.sh files for everyone to execute it using the command **chmod +x startup.sh and chmod +x shutdown.sh**

14.Now we are creating a softlinks to start and stop the tomcat server, using the command **ln -s /opt/apache-tomcat-8.5.79/bin/startup.sh /usr/local/bin/tomcatup and ln -s /opt/apache-tomcat-8.5.79/bin/shutdown.sh /usr/local/bin/tomcatdown(**After executing this commands we can just give tomcatup and tomcatdown to start and stop the server**)** or use **./startup.sh and ./shutdown.sh** to start and stop the server

15.Then again check whether the tomcat server is running or not using the command **ps -ef | grep tomcat**

16. Then add the port number **8080** to the inbound rules which is available under security groups and check whether the page of tomcat is opening by giving Public IPV4 address like **172.31.41.132:8080**

17. To change the port from 8080 to 8090 go to the conf file and inside that edit the server.xml file using the command **vi server.xml.** While editing, change the port inside the server.xml file from 8080 to 8090 and save it using **:wq** . Asusual add the port 8090 in the inbound rules and give **172.31.41.132:8090** and check whether tomcat page is getting opened.

18. To access the sub modules inside the tomcat page, we need to edit the context.xml file as (<!- -and -- >) for values that are available in webapps like **/opt/apache-tomcat-8.5.79/webapps/host-manager/META-INF/context.xml** and **/opt/apache-tomcat-8.5.79/webapps/manager/META-INF/context.xml.**

19.Then you need to add the username and password to access the **manager app page.**

20.To add username inside the conf folder edit the **tomcat-users.xml** file with the code like

**<role rolename="manager-gui"/>**

**<role rolename="manager-script"/>**

**<role rolename="manager-jmx"/>**

**<role rolename="manager-status"/>**

**<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>**

**<user username="deployer" password="deployer" roles="manager-script"/>**

**<user username="tomcat" password="s3cret" roles="manager-gui"/>**

Then save the **tomcat-users.xml** using **:wq** command.

**Steps 2:**

**Installing Jenkins Server**

1.Creating a new instance in aws as **Jenkins-Server**

2. Copy the IPV4 address and go and open the putty and configure the ip address with the respective .ppk file.

3.Then login it with the **username: ec2-user.**

4.Then move to the root directory by giving the command **sudo su**

5.Then install java in it using the command **yum install java-1.8\***

6.Then for a confirmation check the version of java using the command **java -version**

7.First install the wget to install jenkin server using the command **yum -y install wget**

8.To install Jenkins,

**wget -O /etc/yum.repos.d/jenkins.repo** [**https://pkg.jenkins.io/redhat-**](https://pkg.jenkins.io/redhat-) **stable/jenkins.repo**

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

**yum -y install Jenkins**

9.To start the Start jenkins service using the command **systemctl start Jenkins**

10. Setup Jenkins to start at boot use the command **systemctl enable jenkins**

11.To access the Jenkins use [**http://YOUR-SERVER-PUBLIC-IP:8080**](http://YOUR-SERVER-PUBLIC-IP:8080)

12.To test the Jenkins jobs,

1. Create **“new item**”
2. Enter an item name – My-First-Project
   * Chose Freestyle project
3. Under Build section Execute shell : echo "Welcome to Jenkins Demo"
4. Save your job
5. Build job
6. Check "console output"

**Steps 3 :**

**CI/CD THROUGH JENKINS**

1.First login to the aws account and start the instances for both Tomcat and Jenkin servers **(Web Server & Jenkins Server2)**

2.Then, with the **IPV4 address** of both the servers. Login to the putty and start the servers using the commands **./startup.sh in bin folder & systemctl start Jenkins** .

3.By having the localhost address like **65.2.9.133:8090 and 65.2.37.21:8080**. Then check in the browser whether the page is running or not.

4.Now install the git inside the Jenkins server using the commands **sudo su and yum install git** and go to /usr/bin/git folder and copy the path using the command pwd

5.Now go to the **/opt folder** and create a directory called **mkdir maven**

6.Go to the maven folder by using the command **cd maven**

7.Now copy the link to download the maven from the webpage and download it in jenkin server using the command **wget**[**https://dlcdn.apache.org/maven/maven-3/3.8.5/binaries/apache-maven-3.8.5-bin.tar.gz**](https://dlcdn.apache.org/maven/maven-3/3.8.5/binaries/apache-maven-3.8.5-bin.tar.gz)

8. Now unzip the file apache-maven-3.8.5-bin.tar.gz using the command **tar -xvzf apache-maven-3.8.5-bin.tar.gz**

9.Then go to the apache-maven-3.8.5 folder using the command **cd apache-maven-3.8.5**

10. Copy the path of maven using the command **pwd**

11.Login to the **Jenkins webpage 🡪 Give the respective username and password 🡪Get into the Jenkins dashboard**.

12. Before creating the job, install the needed plugins like **maven integration, deploy to container and some plugins needed for git and maven**.

13. Now go to **Manage Jenkins 🡪 Global tool Configuration** and then give the name and add the path that are copied using the **pwd command while installing the git & maven**

14.Now apply and save the **Global tool configuration**

15.Now create a Jenkins job in the name of **DEVOPS CS 001** and select the option **maven project** then click ok

16. Now do the needful configurations

1. **SOURCE CODE MANAGEMENT** 🡪 Select Git

2. **GOALS AND OPTIONS** 🡪 Give clean install package

3. **POLL SCM** 🡪 Give it as \*/1 \* \* \* \* (Continuous integration)

**NOTE** : [ \*/1 \* \* \* \* ] 🡪 it will build automatically after a min when the code is changed.

17. **POST BUILD ACTION** 🡪 Click on add Post-Build action and select the option Deploy war / ear to a container

18. Now apply and save the **DEVOPS CS 001 job**.

19. As we have set the **Poll SCM**, edit the code file and **commit the changes then it will build automatically after a min**

20. Then check whether the build is **FINISHED : SUCCESS** and now give the tomcat IPV4 address like **3.110.81.7:8090/webapp**

21. Finally check the webpage whether the code you have changed is reflecting as the **desired output** in the address **3.110.81.7:8090/webapp**