Install TOMCAT In QA & Production Server

1) Select QA Server and press connect

2) Copy the SSH Command

3) Open GIT Bash & paste the SSH Command

Press Yes

4) Update the apt repository

sudo apt-get update

5) Install tomcat9

sudo apt-get install -y tomcat9

After this we need to install one more package

sudo apt-get install -y tomcat9-admin

6) Check the tomcat is intall or not

Copy the public IP of the QA Server then paste in the browser and in the end enter :8080

qa\_server\_public\_ip:8080

Setting the path of tomcat in jenkins

7) enter linux command in QA Server - cd /etc/tomcat9/

8) enter linux command in QA Server - ls

9) You will find the file tomcat-users.xml

10) Open the file -- sudo vim tomcat-users.xml

11) In the end we need to add one statement

<user username="training" password="samisami" roles="manager-script,manager-status,manager-gui"/>

save and quit

press esc

type :wq

press enter

12) When ever we do any changes done in any service we need to restart the service

sudo service tomcat8 restart

13) After this the same above 12 steps we need to do in the prod server also.

++++++++++++++++++=

Prod Instance

<user username="learning" password="samisami" roles="manager-script,manager-status,manager-gui"/>

+++++++++++++++++++++++++

First Start All the AWS Machines.

Connect Dev Server

Start the Jenkins

java -jar jenkins.war

Stage 1 : Continuous Download START CI-CD

1) Create New item as free style project

2) Click on source code managment

3) Select GIT

4) Enter the URL of github reposiditory

https://github.com/samiuddin-imaad/maven

5) Click on apply and save

6) Run the Job

7) Check the console output.

8) Connect to the dev server

9) Go to the location where code is downloaded

sudo su -

cd path of the folder

ls

Stage 2 : Continuous Build

Convert the java files in to artifact ( .war file)

10) Click on configure of the same job

11) Go to Build Section

12) Click on add build step

13) Click on Invoke top level maven targets

14) Enter the goal as package

15) click on apply and save

16) Run the Job

17) Click on number & click on console output

18) Copy the path of the war file and check the file in the linux machine

sudo su -

cd path

ls

Stage 3 :Continuous Deployment

Now we need to deploy the war file into the QA Server.

19) For this we need to install "deploy to container" plugin.

Go to Dasboard

Click on manage jenkins

Click on manage plugins

Click on avaiable section

Search for plugin ( deploy to container )

Select that plugin and click on install without restart.

20) Click on post build actions of the development job

21) Click on add post build actions

22) Click on deploy war/ear to container

23) Enter the path of the war file (or)

we can give \*\*/\*.war in war/ear files.

24) Context path: qaenv

25) Containers : select tomcat 8

Credentials : Click on add

select jenkins

enter tomcat user name and password

Click on add

Select credentials.

give the private ip of the QA server.

http://private\_ip:8080

http://172.31.32.183:8080

26) Click on apply and save

27) Run the job

28) To access the home page

public\_ip\_Qa\_server:8080/qaenv

Step 3: Create new item ( Name - testing )

Source code management tab, Git

Repository URL - https://github.com/samiuddin-imaad/test-repo

Apply -- Save

Step 4: Run the job.

Step 5: Check the path of the files which are downloaded.

/home/ubuntu/.jenkins/workspace/testing

Step 6: Configure the same job ( testing )

Build -- Add build Step -- Execute shell

( Command: java -jar testing.jar )

Command: echo " Testing passed"

Now both are independent job.

To call testing job after development job is completed

Go to first job ( demo ) -- configure

Post build actions -- add post build action -- build other project -

Projects to build - testing ( name of the job)

++++++++++++++

Copying artifacts from development job to testing job

The artifacts (war) created by the development job should be passed to the testing job so that the testing job can deploy that into tomcat in the prod environment.

Install Plugins

1) Go to Jenkins dashboard

2) Go to manage jenkins

3) Click on Manage plugins

4) Search for "Copy Artifact" plugin

5) Install the plugin

Stage 5 : Continous Delivery

1) Go to Development job

2) Go to Configure

3) Go to Post build actions tab

4) Click on add post build action

5) Click on Archive the artifacts

6) Enter \*\*/\*.war

7) Click on apply and save

8) Go to testing Job

9) Click on configure

10) Go to Build section

11) Click on add build steps

12) Click on copy artifacts from another project

13) Enter Development as project name

14) For Deployment Go to Post build actions section

15) Click on add post build action

16) Click on deploy war/ear to a container

17) Enter \*\*/\*.war in war/ear files

18) Context path : prodenv

19) Click on add container

20) Select tomcat 8

21) Select your Credentials

22) Enter private ip:8080 of the prod server

http://172.31.39.130:8080

23) Click on Apply and save