**5. Building CI/CD pipeline to deploy new version of Application (Jenkins)**

Jenkins is a self contained, open source automation server which can be used to automate all tasks related to building , testing and delivery activities.

Jenkins can be installed even on standalone be any machine with a java runtime envirowment (JRE) Installed.

Jenkins is a tool for Implmenting CI-CD (Continuous Integration - Continuous Delivery)

Stages in CI-CD

Stage 1 : Continuous Download

Stage 2: Continuous Build

Stage 3: Continuous Deployment

Stage 4: Continuous Testing

Stage 5: Continuous Delivery

1-4 ----- Continuous Integration

5 ---- Continuous Delivery

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

Diagram

Description automatically generated with low confidence

Create Instance in AWS

1) Create the AWS Account

2) Login with your aws account

3) Click on Services

4) Click on EC2

5) Click on Instance

6) Click on launch instance

7) Select Ubuntu Server 18 (Free For Eligible)

8) Select t2.micro (Free For Eligible)

9) Click on Next: Configure Instance Details

10) Enter 3 in Number of Instance

11) Click on Add storage

12) Click on Next : Add Tags

13) Click on Next : Configure Security Group

14) Click on Add Rule

15) Select Type as All Traffic

16) Select Source as Anywhere

17) Click on Review and launch

18) If you are doing first time then you need to select Create a new key pair

19) Enter any name in key pair name

20) Click on download key pair

This key pair helps us to connect us to our data center.

21) Click on launch instance

22) Give all 3 instance proper name ( Dev Server, QA Server, Prod Server )

A picture containing graphical user interface

Description automatically generated

How to Connect with the AWS Instance

1) Select that Instance

2) Press Connect

3) You will get the SSH Command

4) Copy the SSH Command

5) Go to the folder where you have place your key then Open GITBASH in local machine

Git Bash you will get automatically when you have install GIT in your local machine.

6) Check your current location using pwd command

7) Now Paste the SSH Command in GITBASH

8) Just Type Yes

9) Now you are connect to the AWS Instance

Now we run any command that command run in the AWS Intance

Important Point : If you are not doing practice in AWS Stop all the instance.

Install Jenkins in AWS Instance

To install Jenkins the first thing we need java file so first we need to install java like we have done in the local instance.

We need to download Java 1.8 or more.

1) Update the apt repository

sudo apt update

2) sudo apt install openjdk-8-jdk -y

3) Check the Java Version

java -version

4) Install Maven & Git

sudo apt-get install -y git maven

5) Check the Verion of Git & Maven

For Git : git --version

For Maven : mvn --version

6) Download & install Jenkins

Open Jenkins website (https://jenkins.io/download/)

Go to Long Term Support

Select Generic Java Package (.war)

We are selecting generic java package file because jenkins will install on those machine where java is already install. If we have java install in windows machine jenkins will work. Only pre requirement is java needs to be install.

For Windows we just need to click on the file and it will download automatically.

For Linux machine enter command wget and paste the url to download the file.

To get the URL right click on generic java package and click on copy link address.

(wget http://mirrors.jenkins.io/war-stable/latest/jenkins.war)

wget <https://get.jenkins.io/war-stable/2.277.2/jenkins.war>

https://get.jenkins.io/war-stable/2.346.2/jenkins.war

11) Start the Jenkins.war file

java -jar jenkins.war

d8eb1aedbdc64d0cba10c3f61d67c4be

Every day if we want to run the jenkins we need to run this command.

12) Access Jenkins Home Page

Select DEV Instance & Press Connect.

Copy the Domain Name On 4th point.

Paste the Domain name in the browser and in the end enter :8080 with the default port number.

We can access the jenkins with dev server Public IP.

Copy the public ip of the dev server and paste the ip address in the browser and in the end enter :8080 with the default port number.

Public

13) Unclock Jenkins

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

When we are installing jenkins it will automatically give you the password in the github terminal.

Copy the password and paste the browser.

You will get the password on the step 11

14) Press Install Suggested Plugins

15) Create First admin user

The first user which we create here is the admin user of the jenkins.

Click on save and continue.

Click on save and finish

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

Create Sample Job

Build tab

Click on execute Shell

In Command Box Enter echo " Hello Jenkins"

Click on Console Output

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

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

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 tomcat8

sudo apt-get install -y tomcat8

After this we need to install one more package

sudo apt-get install -y tomcat8-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

Graphical user interface, text, application, email

Description automatically generated

qa\_server\_public\_ip:8080

Setting the path of tomcat in jenkins

7) enter linux command in QA Server - cd /etc/tomcat8/

8) enter linux command in QA Server - ls

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

Text

Description automatically generated

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

Graphical user interface, text

Description automatically generated

11) In the end we need to add one statement

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

save and quit

press esc

type :wq

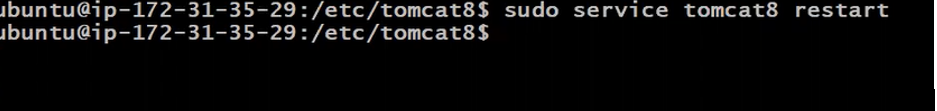
press enter

Text

Description automatically generated

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="sunilsunil" 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/sunildevops77/maven.git

5) Click on apply and save

6) Run the Job

Graphical user interface, application

Description automatically generated

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

Graphical user interface, application

Description automatically generated

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

**Graphical user interface, text, application, email

Description automatically generated**

Graphical user interface, text, application, email

Description automatically generated

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

Graphical user interface, text, application, email

Description automatically generated

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

Graphical user interface, application, Word

Description automatically generated