Setup of linux servers using Vagrant

========================================

1 Download an install oracle virtual box

https://www.virtualbox.org/wiki/Downloads

2 Download and install vagrant

https://developer.hashicorp.com/vagrant/downloads

3 Download the vagrant file and paste it into an empty folder

4 Open command prompt and change directory to the folder where vagrant folder is copied

cd path\_of\_folder\_where\_vagrant\_file\_is\_copied

vagrant up

=======================================================================================

Jenkins

===========

This is a tool used for implementing CI-CD

Stage in CI-CD

====================

Stage 1 (Continuous Download)

------------------------------------

Whenever developers upload some code into the Git repository

Jenkins will receive a notification and it will download

all that code.This is called as Continuous Download

Stage 2 (Continuous Build)

-----------------------------------

The code downloaded in the previosu stage had to converted

into a setup file commonly known ar aritfact.To create this

artifact jenkins uses certain build tools like ANT,Maven etc

The artifact can be inthe format of a .jar,.war..ear file etc

This stage is called as Continuous Build

Stage 3 (Continuous Deployment)

-----------------------------------

The artifact created in the previous stage has to be deployed into

the QAServers where a team of testers can start accessing it.

This QA environment can be running on some application servers like

tomcat,Weblogic etc.Jenkins deploys the artifact into these application

servers and this is called Continuous Deployment

Stage 4 (Continuous Testing)

-----------------------------------

Testers create automation test scripts using tools like selenium,UFT etc

JEnkins run these automation test scripts and checks if the application

is working according to clients requitrement or not,If testing fails

JEnkins will send automated email notifications to the corresponding

team members and developers will finx the defects and upload the modifed

code into Git,Jenkins will again start from stage 1

Stage 5 (Continuous Delivery)

-----------------------------------

Once the application is found to be defect free Jenkins will deploy it

into the Prod servers where the end user or clinet can start accessing it

This is called continuous delivery

Here the first 4 stages represent CI (Continuous Integration)

the laste stage represents CD (Continuous Delivery)

=========================================================================

=========================================================================

Setup of Jenkins

==================================

1 Create 3 AWS ubuntu instances and name then JenkinsServer,QAServer,ProdServer

2 Connect to Jenkinsserver using Gitbash

3 Update the apt repository

sudo apt-get update

4 Install jdk

sudo apt-get install -y openjdk-11-jdk

5 Install git and maven

sudo apt-get install -y git maven

6 Downlaoded jenkins.war

wget https://get.jenkins.io/war-stable/2.361.3/jenkins.war

7 To start jenkins

java -jar jenkins.war

8 To access jenkins open browser

public\_ip\_of\_jenkinserver:8080

9 Unlock jenkins by entering the password

10 Click on Install suggested plugin

11 Create admin user

=========================================================================

Setup of tomcat on Qa and ProdServers

==========================================

1 Connect to QAserver using Gitbash

2 Update the apt repository

sudo apt-get update

3 Install tomcat9

sudo apt-get install -y tomcat9

4 Install tomcat9-admin

sudo apt-get install -y tomcat9-admin

5 Edit the tomcat-users.xml file

cd /etc/tomcat9

sudo vim tomcat-users.xml

Delete all the content from the file and add the below content

<tomcat-users>

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

</tomcat-users>

6 Restart tomcat

sudo service tomcat9 restart

======================================================================

Continuous Download

===============================

1 Open the dashboard of Jenkins

2 Click on New item---->Enter the item name as Development

3 Select Free style project-->OK

4 Go to Source code Management

5 Clcik on Git

6 Enter the github url where developers have uploaded the code

https://github.com/intelliqittrainings/maven.git

7 Click on Apply--->Save

=====================================================================

Continuous Build

============================

1 Open the dashboard of Jenkins

2 Go to the Development job--->Click on Configure

3 Go to Build section

4 Click on Add build step

5 Click on Top level maven targets

6 Enter the maven goal: package

7 Aplly--->Save

========================================================================

Continuous Deployment

============================

1 Open the dashboard of Jenkins

2 Go to Manage Jenkins

3 Click on Manage Plugins

4 Click on Availabl\e section

5 Search for Deploy to container plugin

6 Install it

7 Go to the dashboard of Jenkins

8 Go to the Development job--->Click on configure

9 Go to Post build actions

10 Click on Add post build action

11 Click on Deploy war/ear to container

war/ear file: \*\*/\*.war

Context path: testapp (This is the name that testers will enter in browser to access the

application)

Click on Add container

Select tomcat9

Enter tomcat9 credentials

Tomcat url: private\_ip\_qaserver:8080

12 click on Apply--->Save

===========================================================================

Continuous Testing

===========================

1 Open the dashboard of Jenkins

2 Click on New item

3 Enter some item name (Testing)

4 Select Free style project

5 Enter the github url where testers have uploaded the selenium scripts

https://github.com/intelliqittrainings/FunctionalTesting.git

6 Go to Build section

7 Click on Add build step

8 Click on Execute shell

java -jar path/testing.jar

9 Apply--->Save

================================================================================

Linking the Development job with the Testing job

=======================================================

1 Open the dashboard of Jenkins

2 Go to the dEvelopment job

3 Click on configure

4 Go to Post build actions

5 Click on Add post buuild actions

6 Click on Build another job

7 Enter the job the Testing

8 Apply--->Save

This is called as upstream/downstream configurations

==================================================================

Copying artifacts from Development job to Testing job

=========================================================

1 Open the dashboard of Jenkins

2 Click on Manage Jenkins--->Manage plugins

3 Go to Availbale section--->Search for "Copy Artifact" plugun

4 Click on Install without restart

5 Go to the dashboard of Jenkins

6 Go to the Development job--->Click on Configure

7 Go to Post build actions

8 Click on Add post build actions

9 Click on Archive the artifacts

10 Enter files to be archived as \*\*\\*.war

11 Click on Apply--->>Save

12 Go to the dashboard of jenkins

13 Go to the Testing job---->Click on configure

14 Go to Build section

15 Click on Add build step

15 Click on Copy artifacts from other project

16 Enter project name as "Development"

17 Apply---->Save

================================================================

Stage 5 (Continuous Delivery)

===================================

1 Open the dashboard of jenkins

2 Go to Testing job--->Configure

3 Go to Post build actions

4 Click on Add post build action

5 Click on Deploy war/ear to container

war/ear files: \*\*\\*.war

contextpath: prodapp

Click on Add container

Select tomcat9

Enter username and password of tomcat9

Romcat url: private\_ip\_of\_prodserver:8080

6 Apply---->Save

===========================================================================

Alternate ways of setup of Jenkins

=======================================

1 Update the apt repository

sudo apt-get update

2 Install jdk:1.8

sudo apt-get install -y openjdk-8-jdk

3 Added the jenkins keys to the apt key repository

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \

/usr/share/keyrings/jenkins-keyring.asc > /dev/null

4 Add the debain package repository to the jenkins.list file

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \

https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

5 Update the apt repository

sudo apt-get update

6 Install jenkins

sudo apt-get install -y jenkins

==========================================================================