JENKINS DOCUMENT

Jenkins is continuous integration tool it has five stages

* **Con**tinuous Download
* Continuous Build
* Continuous Deployment
* Continuous Testing
* Continuous Delivery

Installing Jenkins

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

1 Connect to jenkins server using Git bash

2 Update the apt repoistory

sudo apt-get update

3 Install jdk

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

4 Install git and maven

sudo apt-get install -y git maven

5 Download jenkins.war

wget https://get.jenkins.io/war-stable/2.263.4/jenkins.war

6 To start jenkins

java -jar jenkins.war

7 To access jenkins from browser

public\_ip\_of\_jenkinsserver:8080

8 Unlock jenkins by entering the first admin password

9 Click on Install suggested plugins

10 Create first admin user

11 Click on Continue--->Finish

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

wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key

| sudo apt-key add -

4 Add the debain package repository to the Jenkins. List file

sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/

> \ /etc/apt/sources.list.d/jenkins.list'

5 Update the apt repository

sudo apt-get update

6 Install jenkins

sudo apt-get install -y jenkins

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

Setting up tomcat on QA and ProdServer

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

1 Connect to QAServer AWS instance using git bash

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

sudo vim /etc/tomcat9/tomcat-users.xml

Delete the entire content of the file and copy paste the below content

<tomcat-users>

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

</tomcat-users> (create your own user name and not modify the roles)

6 Restart tomcat

sudo service tomcat9 restart

Repeat the above 6 steps on the Prodserver AWS instance(prod server is for delivery the app

To the client or the end user can work on after testing)

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

Stage 1 (Continuous Download)

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

1 Open the dashboard of Jenkins

2 Click on New item---->enter item name as "Development"

3 Select Free style project---->OK

4 Go to Source code management

5 Select Git

6Enter github url where developers have uploaded the code

<https://github.com/intelliqittrainings/maven.git>( use this one or the your own github url)

7 Click on Apply--->Save

8 Go to the dashboard of Jenkins

9 Click on Build icon

This job will download all the code from github repository

into the Jenkins server AWS instance

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

Stage 2 (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 "Invoke top level maven target"

6 Enter Goal as: package

7 Apply--->Save

8 Go to the dashboard of Jenkins

9 Go to the Development job and click on Build icon

This job will convert the code into an artifact and this artifact

comes in the format of a war file

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

Stage 3 (Continuous Deployment)

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

1 Open the dashboard of Jenkins

2 Click on Manage Jenkins

3 Click on Manage Plugins

4 Go to Available section

5 Search for "Deploy to container" plugin

6 Click on Install without restart

7 Go to the dashboard of jenkins

8 Go to the Development job

9 Click on Configure

10 Go to Posy build actions

11 Click on Add post build action

12 Click on Deploy war/ear to containers

war/ear file: \*\*\\*.war Context path: qaapp

Click on Add container

Select tomcat9

Enter tomcat credentials

Tomcat url: private\_ip\_qaserver:8080

13 Apply--->Save

14 Go to the dashboard of jenkins

15 Go to the development job---->Click on build

This job will deploy the artifact into the QAServers

and to access the application from the level of browser

public\_ip\_qaserver:8080/qaapp

Stage 4 (Continuous Testing)

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

1 Open the dashbord of Jenkins

2 Click on New item--->Enter item name as "Testing"

3 Select Free style project--->OK

4 Go to Source Code Management

5 Select Git

6 Enter the github url where testers have uploaded the

selenium test scripts

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

7 Go to Build section

8 Click on Add build step

9 Click onExecute shell

java -jar path\_where\_testing.jar\_is\_downloaded

10 Apply---->Save

11 Go to the dashboard of Jenkins

11 Go to the Testing job---->Click on Build icon

This job will download the selenium test scripts and execute them

Linking the Development job with the Testing job

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

1 Open the dashboard of Jenkins

2 Go to Development job---->Configure

3 Go to Post build actions

4 Click on Add post build action

5 Click on Build other projects

6 Enter project name as "Testing"

7 Click on Save

Now if we run the Development job it will run the first three stages

of CI-CD and then it will call the Testing job which will run the

fourth stage

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

In lot of cases the Jenkinsfile is uploaded by the

developers into the remote git repository and from there

it will trigger all stages of CI-CD

Developers activity

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

1 Cone the remote git repository

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

2 Move into the maven folder and delete .git folder

cd maven

rm -rf .git

3 Initlise a new git repository

git init

4 Create a file called as Jenkinsfile and paste the Pipeline code into it

vim Jenkinsfile

Copy Paste the Groovy pipeline code here

5 Send the files to stagging and local repository

git add .

git commit -m "a"

6 Open github.com--->Create a new remote repository

and push the code into it

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

JenkinsAdmin Activity

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

1 Open the dashboard of Jenkins

2 Click on New item--->Enter some item name--->Select Pipeline Project

3 Go to Pipeline section

4 In Definition select Pipeline script from SCM

5 Select SCM as Git

6 Enter the GitHub url where developer has uploaded the code

7 Go to Build triggers

8 Click on POLL SCM and in Schedule section: \* \* \* \* \*

9 Click on Apply--->Save

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

Multi Branch Pipeline

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

Generally developers create multiple branches and upload

code related to various functionalities on these branches

We have to configure Jenkins in such a way that it triggers

CI-CD process for all these branches parallelly.

To do this we need to have a copy of Jenkins file on each branch

and then based on the instructions in the Jenkinsfile all

the stages have to be triggered

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

Developers Activity

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

1 Clone the maven repository

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

2 Move into this cloned repository and delete. git folder

cd maven

rm -rf. git

3 Initilise a new git repository

git init

4 Send the files into stagging area and local repository

git add.

git commit -m "a"

5 Create a Jenkins file and put the stages of CI that should happen

on master branch

vim Jenkinsfile

6 Send it to stagging and local repository

git add .

git commit -m "b"

7 Create a new branch called loans and create a new Jenkinsfile

git checkout -b loans

vim Jenkinsfile

Use the CI instructions that should be done on Loans branch

8 Send this to stagging and local repository

git add .

git commit -m "c"

9 Open github.com---->Create a new repository

10 Push all the branches from local machine to remote github

git push origin --all

Jenkins Admin Activity

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

1 Open the dashboard of Jenkins

2 Click on New item---->Enter item name as Multibranch Pipeline

3 Select Multibranch Pipeline--->OK

4 Go to Branch Sources---->Select Git-->enter github url where developers

uploaded the code

5 Go to Scan Multi branch pipeline triggers---->Select 1 minute

6 Apply--->Save

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

Webhooks

===========

This is used to send notifications from github to Jenkins

Whenever any code changes are done and that is checking into

github, webhook will send an immediate notification to Jenkins

and Jenkins will trigger the job

1 Open github.com---->Click on the repository that we are working on

2 On the right corner click on Setting

3 Click on Webhooks in the left panel

4 Click on Add Webhook

5 In Payload URL: http://public\_ip\_jenkinsserver:8080/github-webhook/

6 In Content type select :application/json

7 Click on Add Webhook

8 Open the dashboard of Jenkins

9 Go to the job that we want to configure

10 Go to Build triggers

11 Check GitHub hook trigger for GitScm polling

12 Click on Apply--->Save

Now if we make any changes to the code in github then github

will send an notification to Jenkins and Jenkins will run that job