DevOps Project2

Github, Jenkins, Ansible

Developer will upload his java code along with testing code in Github

Github will be integrated with Jenkins-server

Jenkins will compile and build the code and deploy it to QA server (in which Ansible will also be installed)

Then Jenkins will trigger ansible playbook in same QA server via ssh command

Playbook will upload that code on webserver.

1. launch 3 ubuntu ec2--> jenkins-server, QA-server(Ansible), web-server
2. connect to jenkins-server--> install java, Jenkins, git, maven

**Setups of jenkins as a direct installation**

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

1 Update the apt repository

sudo apt-get update

2 Install java

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

3 Add the jenkins repository keys to the apt repository

wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

4 Add the debain package address to 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

7 apt-get install git maven -y

1. connect to QA-server--> tomcat9, tomcat9-admin, ansible

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 -------- it contains all the dependencies that require for tomcat

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 add the below data

<tomcat-users>

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

</tomcat-users>

These credentials we need to pass in Jenkins so that it can deploy artifact in it.

* Here there is a file servers.xml in which you can change your port number.

6 Restart tomcat9

sudo service tomcat9 restart

7 To access tomcat from the level of browser

public\_ip\_qaserver:8080

8 # apt-get install ansible

1. connect to web-server--> install tomcat9, tomcat9-admin

1 Connect to web-Server 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 -------- it contains all the dependencies that require for tomcat

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 add the below data

<tomcat-users>

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

</tomcat-users>

These credentials we need to pass in Jenkins so that it can deploy artifact in it.

* Here there is a file servers.xml in which you can change your port number.

6 Restart tomcat9

sudo service tomcat9 restart

1. connect to ansible server-->

vim /etc/ansible/hosts

add -->

[web]

private\_ip\_of\_webserver

1. create pw-less authentication between ansible-server and jenkins-server

Get connected with ansible-server

passwd root

vim /etc/ssh/sshd\_config

PermitRootLogin yes

passwordAuthentication yes

:wq

systemctl restart sshd

Get connected with jenkins-server

# ssh-keygen

# ssh-copy-id -i root@private\_ip\_of\_ansible-server

Finally make sure whether passwordless authentication is successful or not

# ssh root@ private\_ip\_of\_ansible-server

1. create pw-less authentication between ansible-server and web-server

Get connected with web-server

passwd root

vim /etc/ssh/sshd\_config

PermitRootLogin yes

passwordAuthentication yes

:wq

systemctl restart sshd

Get connected with ansible-server

# ssh-keygen

# ssh-copy-id -i root@private\_ip\_of\_httpd-server

Finally make sure whether passwordless authentication is successful or not

# ssh root@ private\_ip\_of\_web-server

1. upload your java-code repo in github
2. integrate github with jenkins

go to jenkins

copy jenkins url

go to gihub page

settings --> webhooks--> add webhook

Payload url -->http://jenkins\_ip:8080/github-webhook/

content type--> application/json

go to Jenkins page

cloud--> configure --> add new token--> generate --> copy token keep it somewhere--> apply and save

go to github page

secret--> paste token

add webhook

1. install 'publish over ssh' plugin in Jenkins

we need this plugin to run any command on specific server managed by Jenkins.



Get connected with jenkins-server

passwd root

vim /etc/ssh/sshd\_config

PermitRootLogin yes

passwordAuthentication yes

:wq

systemctl restart sshd

2. manage jenkins--> configure system --> scroll down--> SSH servers--> Add
3. fill details of jenkins server, give root id and pw,

name: Jenkins

Hostname: private-ip-of-jenkins-server

1. --> Test Configuration--> make sure whether successful or not

do same for Ansible server



create new job--> freestyle --> give name ‘deployment’

source code management --> git

give url of java-code repo

change ‘master’ to ‘main’

build now

Build Trigger

Github hook trigger got GITScm polling



Add build step🡪select ‘invoke top level maven targets’ 🡪 add goals🡪‘package’

1. Apply – save🡪 build

War file will get generated which we will deploy to QA-server

1. Create new job🡪 testing 🡪 freestyle project
2. Add github url of testing repo from github 🡪 build
3. Install plugin 🡪 deploy over ssh
4. Go to deployment job 🡪 configure🡪 post-build actions🡪send build artifacts over ssh 🡪 select Jenkins 🡪

Exec command🡪 rsync -zavh /var/lib/jenkins/workspace/deployment/webapp/target/\*.war root@QA-server-ip:/var/lib/tomcat9/webapps/testapp.war

Build now so that war file will get deployed to QA server where we will do testing

1. Go to testing job 🡪 configure

1 Go to Build section

2 Click on Add Build step

3 Click on Execute shell

java -jar path/testing.jar

to check path go to console o/p of first job in testing job. Copy that path and go to Jenkins server and cd to that path. Check whether testing.jar is available.

Path🡪 /var/lib/jenkins/workspace/testing

4Apply--->Save

5 Go to the dashboard of jenkins

6 Go to the testing job--->Click on Build icon

This job will download the selenium test scripts and execute them.

1. Now if testing passes next step is to deploy this war file to web-server
2. Create playbook in QA/ansible server

# mkdir /sourcecode

# vim playbook1.yml

---

- name: deploying project to prod server

hosts: all

tasks:

- name: copy .war file to web-server

copy:

src: /var/lib/tomcat9/webapps/testapp.war

dest: /var/lib/tomcat9/webapps/shubhya.war

…

1. Go to testing job 🡪 configure🡪 postbuild action🡪 add postbuild action 🡪 send build artifacts over ssh🡪 name- ansible 🡪 exec command: ansible-playbook /sourcecode/playbook1.yml -b  
   apply🡪 save 🡪 build
2. Check whether its deployed successfully in web-server

Go to browser and check

Public-ip-of-webserver:8080/testapp

1. Lets link both jobs

Go to deployment job🡪 configure🡪 postbuild actions 🡪 add postbuild actions 🡪 build other projects 🡪 projects to build: testing 🡪 save apply