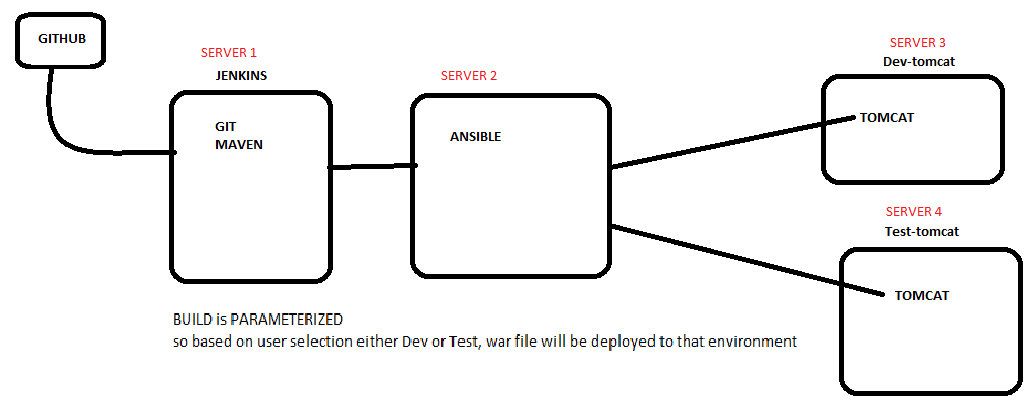
# ANSIBLE DEPLOYMENT PIPELINE



STEPS:-

========

1.install jenkins,maven,git on server1

2.create a sample jenkins job to find the workspace of jenkins on linux

3.configure epel repo and install ansible,check python version on server 2

4.ansible communicates with nodes via SSH, so create a user and add him on sudoers file and also enable passwordless authentication from server2 to server3 with same user or nodes

5.communicate from server2 to server3 via ping

6.install ssh plugin on jenkins and create a job to copy binaries from jenkins to ansible server i.e server3

7.install java and tomcat on server3,server4

8.write a playbook to copy binaries from server2 to server3 or server4 by triggering a Parameterized jenkins job based on user selection

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

STEP 1

+++++++

In SERVER 1:-

==========

installing java,maven,jenkins on ubuntu(20.04):-

JAVA-8:-

—---------

sudo apt install openjdk-8-jdk -y

whereis jvm

update-java-alternatives --list

vi .bashrc

JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64

export JAVA\_HOME

PATH=$PATH:$JAVA\_HOME/bin

export PATH

source .bashrc

. /etc/environment

echo $JAVA\_HOME

/usr/lib/jvm/java-1.8.0-openjdk-amd64

MAVEN:-

—---------

cd /opt

mkdir maven

cd maven

wget https://downloads.apache.org/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz

tar xvfz apache-maven-3.6.3-bin.tar.gz

vi /etc/profile.d/maven.sh

export M2\_HOME=/opt/maven/apache-maven-3.6.3

export PATH=${M2\_HOME}/bin:${PATH}

source /etc/profile.d/maven.sh

mvn --version

JENKINS:-

—------------

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

sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

sudo apt update

sudo apt install jenkins

sudo systemctl start jenkins

sudo systemctl status jenkins

sudo systemctl enable jenkins

sudo ufw allow 8080

—--------------------------------------------------------------------------------------------------------

STEP 2

+++++++

In SERVER 3 and SERVER 4:-(ubuntu)(Dev-tomcat, Test-tomcat)

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

goto server3 and server4 NEXT

sudo apt update -y

sudo useradd -m vijay -s /bin/bash

sudo passwd vijay

- add password

visudo

# User privilege specification

root ALL=(ALL:ALL) ALL

vijay ALL=(ALL) NOPASSWD: ALL

sudo vim /etc/ssh/sshd\_config

- change "PasswordAuthentication yes" in the file

sudo systemctl restart sshd

STEP 3

+++++++

In SERVER 2:-(ubuntu)(Ansible server)

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

goto server2

sudo apt update -y

sudo useradd -m vijay -s /bin/bash

sudo passwd vijay

- add password

visudo

# User privilege specification

root ALL=(ALL:ALL) ALL

vijay ALL=(ALL) NOPASSWD: ALL

sudo vim /etc/ssh/sshd\_config

- change "PasswordAuthentication yes" in the file

sudo systemctl restart sshd

sudo apt install ansible

su - vijay

ssh-keygen ---this command generates both public and private key, we need to copy the public key to target server

ssh-copy-id vijay@44.203.164.200 ----server3 ip address,public key will be copied to (/home/vijay/.ssh/authorized\_keys file) of server3

ssh-copy-id vijay@52.91.197.166 server4 ip address,public key will be copied to (/home/vijay/.ssh/authorized\_keys file) of server4

- provide vijay password --- for the first time it asks password

- ssh -i vijay@44.203.164.200 -- can able to login to server3 from server2 without password

- exit

cd /etc/ansible --- ansible communicates with nodes by using this hosts file

sudo vi hosts

[webserver1]

44.203.164.200 ---server 3

[webserver2]

52.91.197.166 ----server 4

ansible all -m ping --- test whether we get response from server3

sudo vim /etc/ssh/sshd\_config

- change "PasswordAuthentication yes" in the file

sudo systemctl restart sshd

cd /opt

sudo mkdir artifact

sudo chown -R vijay:vijay /opt/artifact

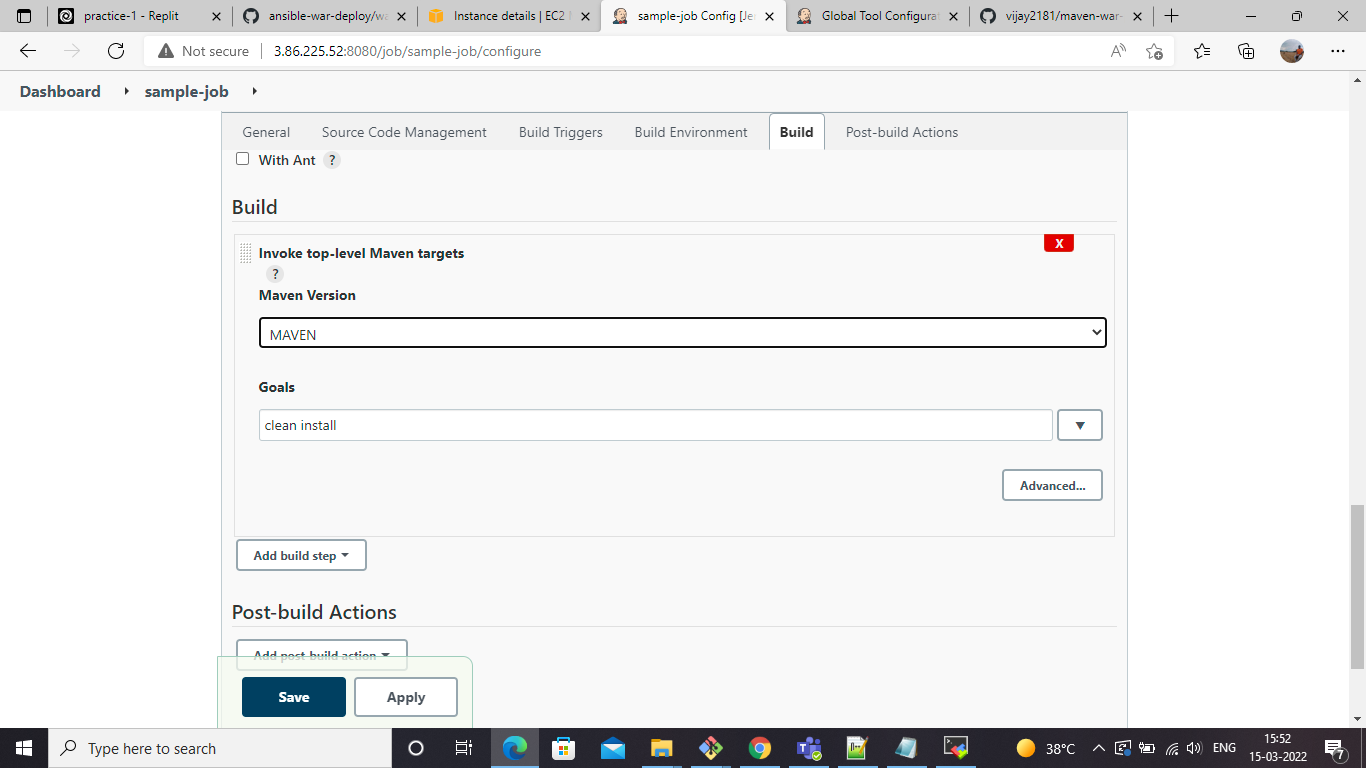
STEP 4

+++++++

In SERVER 1:-

===========

once build a sample job to know the location of jenkins



workspace(/var/lib/jenkins/workspace/sample-job/target/project-3-1.0-SNAPSHOT.war)

sudo apt-get update -y

sudo apt-get install -y sshpass

passwd jenkins –jenkins through ‘jenkins’ user runs the jobs

su - jenkins

ssh-keygen ---this command generates both public and private key, we need to copy the public key to ansible server

ssh-copy-id vijay@44.203.164.200 ----ansible ip address,public key will be copied to - provide vijay password --- for the first time it asks password

- ssh -i vijay@44.203.164.200 -- can able to login to ansible server without password

- exit

sshpass -p 'vijay' ssh 'vijay@18.212.36.144' "ansible all -m ping"

* The above command will give the results from ansible server(remote) which is to be executed on jenkins server
* -p option requires password

STEP 5

+++++++

SERVER 3 and SERVER 4:-(ubuntu)(Dev-tomcat, Test-tomcat)

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

TOMCAT INSTALLATION:-

sudo apt install openjdk-8-jdk -y

cd /opt

sudo mkdir tomcat && cd tomcat

sudo wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.60/bin/apache-tomcat-9.0.60.tar.gz

sudo tar -xvf apache-tomcat-9.0.60.tar.gz

sudo chown -R vijay:vijay /opt/tomcat/apache-tomcat-9.0.60/

cd apache-tomcat-9.0.60/

cd conf

pwd

/opt/tomcat/apache-tomcat-9.0.60/conf

changing tomcat server port:-

for server 3 :-

sudo vim server.xml

<Connector port="8081" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

server 4:-

<Connector port="8082" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

we need have a user to authenticate to tomcat

cd /opt/tomcat/apache-tomcat-9.0.60/conf

sudo vim tomcat-users.xml

- add only below content in the file

<?xml version="1.0" encoding="UTF-8"?>

<tomcat-users>

<role rolename="manager-gui"/>

<role rolename="admin-gui"/>

<user username="tomcat" password="tomcat" roles="manager-gui,admin-gui"/>

</tomcat-users>

we need to allow who can access the Tomcat

sudo vim /opt/tomcat/apache-tomcat-9.0.60/webapps/manager/META-INF/context.xml

- give the below value in the file

allow=".\*" />

cd /opt/tomcat/apache-tomcat-9.0.60/bin

./startup.sh --- starts the tomcat web server

<ip address>:8081

- manager app

- asks for username and password

whatever the content placed in webapps folder will be hosted on browser

- we need to write a playbook to copy the war file from Ansible server(server 2) to server 3

STEP 6

+++++++

SERVER 2:-(ubuntu)(Ansible server)

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

cd /etc/ansible/

sudo mkdir playbooks

cd playbooks

sudo vim copy-war.yml

---

- hosts: webserver1

become: true

tasks:

- name: copying war file from ansible server to tomcat webapps folder

copy: src=/opt/artifact/\*SNAPSHOT.war dest=/opt/tomcat/apache-tomcat-9.0.60/webapps

- hosts: webserver2

become: true

tasks:

- name: copying war file from ansible server to tomcat webapps folder

copy: src=/opt/artifact/\*SNAPSHOT.war dest=/opt/tomcat/apache-tomcat-9.0.60/webapps

goto jenkins sample-jenkins job

* Select pipeline job

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

FINAL SCRIPT FOR COMPONENT DEPLOYMENT:-

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

#!groovy

pipeline {

agent any

tools {

maven 'MAVEN'

}

parameters {

choice(name: 'BUILD', choices: ['Yes', 'No'], description: 'Pick Yes or No')

string(name: "WAR\_FILE\_PATH", defaultValue: "/var/lib/jenkins/workspace/test-1/target/\*SNAPSHOT.war", trim: false, description: "Provide war file job path in jenkins")

choice(name: "ENVIRONMENT", choices: ["", "Dev", "Test", "PROD"], description: 'Choose an environment to deploy')

string(name: "ANSIBLE\_IP", defaultValue: "18.212.36.144", trim: false, description: "Provide Ansible Server IP-ADDRESS")

password(name: "ANSIBLE\_USER\_PASSWORD", defaultValue: "vijay", description: "Provide Ansible server user password")

}

stages {

stage('environment-setting') {

steps {

script {

env.BUILD = "Yes" // Setting env variable for Build

env.USER = 'Vijay'

//Generally it is possible to use Groovy’s conditionals in a declarative syntax, when we use a script step.

}

}

}

stage ('Checkout SCM'){

steps {

checkout([$class: 'GitSCM', branches: [[name: '\*/main']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/vijay2181/java-maven-SampleWarApp.git']]])

}

}

stage ('Build') {

when {

// Only build if "BUILD" value is 'Yes'

expression { params.BUILD == 'Yes' }

}

steps {

echo "Hello, Vijay!"

sh "mvn clean install"

sh "scp ${params.WAR\_FILE\_PATH} vijay@${params.ANSIBLE\_IP}:/opt/artifact"

}

}

stage("Deploy") {

parallel { //parallel is a switch case

stage("Dev") {

when {

allOf {

expression { params.BUILD == 'Yes' }

expression { params.ENVIRONMENT == 'Dev' } //two conditions must satisfy

}

}

steps {

echo "Executing Ansible playbook on remote Ansible Host from jenkins Server"

sh "sshpass -p '${params.ANSIBLE\_USER\_PASSWORD}' ssh 'vijay@${params.ANSIBLE\_IP}' 'ansible-playbook /etc/ansible/playbooks/copy-war.yml --limit webserver1' "

}

}

stage("Test") {

when {

allOf {

expression { params.BUILD == 'Yes' }

expression { params.ENVIRONMENT == 'Test' }

}

}

steps {

echo "Executing Ansible playbook on remote Ansible Host from jenkins Server"

sh "sshpass -p '${params.ANSIBLE\_USER\_PASSWORD}' ssh 'vijay@${params.ANSIBLE\_IP}' 'ansible-playbook /etc/ansible/playbooks/copy-war.yml --limit webserver2' "

}

}

stage("PROD") {

when { expression { params.BUILD == 'Yes' } }

steps {

sh "echo 'vijay'"

}

}

}

}

}

}

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