pipeline {

agent any

// tools {

// maven 'maven3.6'

// }

// environment {

// M2\_INSTALL = "/usr/bin/mvn"

// }

stages {

stage('Clone-Repo') {

steps {

checkout scm

}

}

stage('Build') {

steps {

sh 'mvn install -Dmaven.test.skip=true'

}

}

stage('Unit Tests') {

steps {

sh 'mvn compiler:testCompile'

sh 'mvn surefire:test'

junit 'target/\*\*/\*.xml'

}

}

stage('Deployment') {

steps {

sh 'sshpass -p "gamut" scp target/gamutgurus.war gamut@172.17.0.4:/home/gamut/Distros/apache-tomcat-8.5.61/webapps'

sh 'sshpass -p "gamut" ssh gamut@172.17.0.4 "/home/gamut/Distros/apache-tomcat-8.5.61/bin/startup.sh"'

}

}

}

}

What is difference between Ansible Playbook and Ansible role?

Have you heard of Ansible Galaxy. What is it used for?

How will you store password, servername in a secure manner in Ansible playbook. How will you secure them.

Have you used Ginger with Ansible

Have you have experience in Ansible tower. How have you used it?

1)maintain all the playbooks in Github/gitlab/bitbucker

2)Establish connectivity between tower and git hub

3)Run the playbooks from the tower(launch)

4)we can see playbooks , inventories, hosts, failed hosts

Command to check list of modules

ansible-doc yum

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

1)playbook for java installation

[]vi java.yml

---

- hosts: webserver

remote\_user: root

become: yes

tasks:

- name: installalling java jdk 8 and it's dependencies

yum: name=java-1.8.0-openjdk state=installed

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

2)Playbook for tomcat

[]vi tomcat.yml

---

- hosts: webserver

remote\_user: root

become: yes

tasks:

- name: removing older version of tomcat

file: path=/usr/local/tomcat state=obsent

- name: download tomcat

get\_url: url=http://mirrr........

- name: create tomcat folder

command: mkdir /usr/local/tomcat

- name: extract archive

unarchive: src=/tmp/apache-tomcat-8.0.9.tar.gz dest=/usr/local/tomcat

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

[]wget url

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

3) role for mysql

---

- hosts: webserver

remote\_user: root

gather\_facts: false

roles:

- mysql

[]ansible-galaxy init mysql

initializing the role

[]cd /etc/ansible/rloes

[]ansible-galaxy init mysql

[]cd roles

[]mysql

[]ls -la

[]cd defaults

[]vi main.yml

mysql\_user: mysql

mysql\_port: 3306

mysql\_datadir: /var/lib/mysql

mysql\_bind: 0.0.0.0

mysql\_pkg: mysql\_server

mysql\_pid: /var/run/mysql/mysqld.pid

mysql\_socket: /var/lib/mysqld/mysql.sock

mysql\_confpath: /etc/mycnf

mysql\_service: mysql

[]cd handlers.yml

[]vi main.yml

---

- name: restart mysql service

service: name="{{mysql\_service}}" state=restart

[]cd tasks/

[]vi main.yml

---

- name: install mysql servername

yum: name="{{mysql-pkg}}" state=present

- name: create mysql config

template: src="{{mycnf.j2}}" dest="{{mysql\_cnfpath}}"

notify:

- restart: mysql service

[]cd templates

[]vi mycnf.j2

user="{{mysql\_user}}"

pid\_file="{{mysql\_pid}}"

socket="{{mysql\_socket}"

port="{{mysql=port}}"

datadir="{{mysql\_datadir}}"

bind\_address="{{mysql\_bind}}"

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

4)Playbook for Jenkins:

---

- hosts:

remote\_user: root

become: yes

gather\_facts: false

tasks:

- name: Ensure Jenkins Repository is Installed

yum\_repository:

name: jenkins

state: present

description: Official Jenkins Yum Repo

baseurl: http://pkg.jenkins.io/redhat

- name: Ensure Jenkins is Installed

yum :

name: jenkins

update\_cache: yes

state: present

- name: Enable and Start the Jenkins Service

service:

name: jenkins

enabled: yes

state: started

- name: Open Firewall Port

firewalld:

zone: public

port: 8080/tcp

permanent: true

state: enabled

immediate: true

types or states:

stop

started

enabel

restart

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

./startup.sh

[]./stratup.sh

[]./shutdown.sh

Note:

update\_cache=yes tells Ansible's +module to refresh the caches before applying whatever change is necessary (if any)

When immediate is set to true on an Command component, the action is invoked in the Apply Request Values phase

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

5)Playbook for copyimg keys

---

- hosts: webserver

remote\_user: root

become: yes

gather\_facts: false

tasks:

- name: make a directory

file:

path: "/home/{{ansible\_user/.ssh}}"

state: directroty

- name: create empty file

file:

path: "/home/{{ansible\_user}}/.ssh/authorized\_keys"

state: touch

- name:append the key

lineinfile:

path: "/home/{{ansible\_user}}/.ssh/authorized\_keys

line: "{{pubkey}}"

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

7)Playbook for ping test

---

- hosts: webserver

remote\_user: root

become: yes

tasks:

- name: ping machine

ping:

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

8)Playbook to install docker

---

- hosts: webserver

remote\_user: root

become: yes

tasks:

tasks:

- name: Install yum utils

yum:

name: yum-utils

state: latest

- name: Install device-mapper-persistent-data

yum:

name: device-mapper-persistent-data

state: latest

- name: Install lvm2

yum:

name: lvm2

state: latest

- name: Add Docker repo

get\_url:

url: https://download.docker.com/linux/centos/docker-ce.repo

dest: /etc/yum.repos.d/docer-ce.repo

become: yes

- name: Install Docker

package:

name: docker-ce

state: latest

become: yes

- name: Start Docker service

service:

name: docker

state: started

enabled: yes

become: yes

Note:

yum-utils is a collection of tools and programs for managing yum repositories, installing debug packages, source packages, extended information from repositories and administration.

devicemapper requires the lvm2 and device-mapper-persistent-data packages to be installed. Changing the storage driver makes any containers you have already created inaccessible on the local system.

docker-lvm-plugin is a volume plugin, which creates logical volumes using LVM2. LVM2 is a userspace toolset that provides logical volume management in Linux.

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

9)Installing Nexus through playbook

---

- hosts: webserver

remote\_user: root

become: yes

tasks:

- name: Install wget

yum: name=wget state=installed

- name: Create Nexus installation directory

file:

path={{ nexus\_installation\_dir }}

state=directory

become: true

- name: Download Nexus package

get\_url: url="http://download.sonatype.com/nexus/oss/{{ nexus\_package }}" dest={{ nexus\_download\_dir }}/{{ nexus\_package }}

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

10)installing python

---

- hosts: webserver

remote\_user: root

become: yes

- name: Install Python

raw: yum -y install python

command: yum -y install python

when: check\_python.rc != 0

On Ubuntu systems we use the apt command (and we have to update its cache):

- name: Install Python

raw: apt -y update && apt install -y python-minimal

when: check\_python.rc != 0

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

11)installing Mongo DB

---

- hosts: webserver

remote\_user: root

become: yes

- name: Add MongoDB repository

apt\_repository: repo='deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' state=present

- name: install mongodb

apt: pkg=mongodb-org state=latest update\_cache=yes

notify:

- start mongodb

handlers:

- name: start mongodb

service: name=mongod state=restarted

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

12)Playbook to install nginx

---

- hosts: all

tasks:

- name: ensure nginx is at the latest version

yum: name=nginx state=installed

- name: start nginx

service: name=nginx state=started

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

13)Playbook for selinum

---

- hosts: webserver

remote\_user: root

become: yes

---

- name: Download the gecko driver

get\_url: url="https://github.com/mozilla/geckodriver/releases/download/v{{ gecko\_version }}/geckodriver-v{{ gecko\_version }}-macos.tar.gz" dest=/tmp mode=0777

- name: Install the gecko driver

unarchive: src="/tmp/geckodriver-v{{ gecko\_version }}-macos.tar.gz" dest=/usr/local/bin/ mode=0700 copy=no

- name: Download the chrome driver

get\_url: url="http://chromedriver.storage.googleapis.com/{{ chrome\_version }}/chromedriver\_mac32.zip" dest=/tmp mode=0777

- name: Install the chrome driver

unarchive: src=/tmp/chromedriver\_mac32.zip dest=/usr/local/bin/ mode=0700 copy=no

- name: Download the safari extension

get\_url: url="http://selenium-release.storage.googleapis.com/{{ safari\_version }}/SafariDriver.safariextz" dest=/tmp mode=0777

- name: Open the safari extension

command: open /tmp/SafariDriver.safariextz

- name: Download selenium zip

get\_url: url="https://github.com/SeleniumHQ/selenium/archive/selenium-{{ selenium\_version }}.zip" dest=/tmp mode=0777

- name: Unzip selenium

unarchive: src="/tmp/selenium-selenium-{{ selenium\_version }}.zip" dest=/tmp mode=0700 copy=no

- name: Build Selenium

command: ./go clean release

args:

chdir: "/tmp/selenium-selenium-{{ selenium\_version }}/"

creates: "/tmp/selenium-selenium-{{ selenium\_version }}/build/dist/selenium-server-standalone-{{ selenium\_version }}.jar"

- name: Move Selenium build

command: mv "selenium-server-standalone-{{ selenium\_version }}.jar" /usr/local/bin/selenium-server.jar

args:

chdir: "/tmp/selenium-selenium-{{ selenium\_version }}/build/dist/"

creates: /usr/local/bin/selenium-server.jar

- name: Create script for selenium

template: src=selenium dest=/usr/local/bin/selenium mode=0700

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

Steps to perform to automate AWS infra through Ansible:

1)Launch t2.micro and install ansible-doc

2)install ansible

-->yum install wget

-->wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

-->rpm -ivh epel-release-latest7.noarch.rpm

-->sudo yum -y update

-->sudo yum install python python-pip open-ssl ansible

-->ansible --version

4)yum -y install pyhthon-pip

5)install boto3 package (pip install boto3)

6)python -m pip install awscli

Playboooks:

1)Playbook for VPC

[]vi vpc.yml

---

- hosts: localhost

connection: local

remote\_user: root

become: yes

gather\_facts: no

tasks:

- name: creating VPC

ec2\_vpc\_net:

region: us-east-1

name: myvpcyum

state: present

cidr\_block: 20.0.0.0/16

[]vi awscredentials.yml

aws\_key: \*\*\*\*\*

aws\_id: \*\*\*\*\*

aws\_region: \*\*\*\*

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

2)Playbook for IAM user

[]vi iam.yml

---

- hosts: localhost

connection: local

remote\_user: pavani

become: yes

gather\_facts: no

tasks:

- name: creating coule of IAM users

iam:

region: us-east-1

iam\_type: users

name: "{{item}}"

state: present

password: "pavani@123"

with\_items:

- test1

- test2

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

3)Creating playbook for ec2 instance

[]vi ec2.yml

---

- hosts: localhost

connection: local

remote\_user: pavani

become: yes

gather\_facts: no

tasks:

- name: creating ec2

ec2

region: us-east-1

image: ami-040903569d18d3cde

instance\_type: t2.micro

count: 1

groupid: sg-5c5b0572

key\_name: abc

vpc\_subnet\_id: subnet-cf11b8c1

assign\_public\_ip: yes

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

---

- hosts: localhost

connection: local

remote\_user: root

become: yes

gather\_facts: no

tasks:

- name: creating VPC

ec2\_vpc\_net:

region: us-east-1

name: myvpcyum

state: present

cidr\_block: 20.0.0.0/16

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

---

- hosts: localhost

connection: local

remote\_user: pavani

become: yes

gather\_facts: no

tasks:

- name: creating ec2

ec2:

region: us-east-1

image: ami-040903569d18d3cde

instance\_type: t2.micro

count: 1

group\_id: sg-5c5b0572

key\_name: abc

vpc\_subnet\_id: subnet-cf11b8c1

assign\_public\_ip: yes

[]pip install boto

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