**Ansible**

**installation ansible**

* Sudo su –
* Change host name (hostname ansible-control-node)
* Sudo su –
* amazon-linux-extras install java-openjdk11 –y
* java –version
* vi /etc/ssh/sshd-confg
* # authentication password yes (remove the # tag)
* service sshd reload
* Yum install python
* Yum install python-pip –y
* Pip install ansible
* Ansible –version
* ( if configure file is none at the time we have to create local host)
* ( mkdir /etc/ansible 🡪 cd /ect/ansible 🡪 vi hosts 🡪 localhost <type and save> 🡪 cat hosts)
* vi /etc/sudoers
* ansadmin ALL(ALL) NOPASSWD:ALL (paste under root and save)
* yum install docker –y
* docker –version
* service docker start
* service docker status
* useradd dockeradmin
* passwd dockeradmin
* new password : ansadmin
* conform password : ansadmin
* sudo mkdir docker ( it shoud be on /opt/)
* ls (docker will show)
* sudo chown –R ansadmin:ansadmin /opt/docker (ls -l)
* usermod -aG docker dockeradmin

**password less connection-beetween ansible and docker instance**

* sudo su – ansadmin
* ssh-keygen
* ls –al
* cd .ssh
* ssh-copy-id ansadmin@ docker IP
* yes and give password
* number of keys added : 1
* cd..
* ssh ansadmin@docker IP (docker will connected with the ansible)
* exit (again it will came to ansible server)
* ssh-copy-id localhost (yes and give password)
* added one key (that is localhost)
* ansible all –m ping (it will ping successfully )

**Docker**

**Docker-installation on ansible and docker instance**

**Install docker and start docker services**

* sudo su –
* host name docker-host
* sudo su -
* amazon-linux-extras install java-openjdk11 –y
* java –version
* vi /etc/ssh/sshd-confg
* # authentication password yes (remove the # tag)
* service sshd reload
* yum install docker –y
* docker --version
* start docker services
* service docker start
* service docker status

**Create a user called dockeradmin**

* useradd dockeradmin
* passwd dockerad min
* new password : ansadmin
* conform password : ansadmin
* vi /ete/sudoers
* ansadmin ALL=(ALL) NOPASSWD:ALL

**add a user to docker group manage docker**

* usermod -aG docker dockeradmin

**Connection between Jenkins and ansible**

* go to Jenkins
* manage Jenkins 🡪configure system 🡪 publish over SSH
* ADD
* SSH server name : ansible server (name of ansible)
* host name : ansible IP (paste)
* user name : ansadmin 🡪 use password authentication
* remote directory :
* pass phase/password:

click on test configuration 🡪 success (now successfully connected Jenkins to ansible server)

🡪 apply and save

Docker is a target system

Go to the Github 🡪 select source code 🡪 hosts 🡪 localhost 🡪 docker IP (paste)

Update 🡪 Change commit

* login to dockerhub and credentials on ansible instance using “**root user**”
* sudo su – (it will connecting as a root user)
* docker login
* user name and password (dockerhub credentials)
* login successful

**Jenkins-job (Maven Project)**

* **project name:** deploy on container using ansible
* go to Jenkins 🡪 new item 🡪 project name 🡪 maven project 🡪 ok
* git 🡪 git url 🡪 branch : \*/main
* Build 🡪 pom.xml 🡪 Goals : clean install package
* Post build action 🡪 send build artifacts over SSH
* **Source file :** dockerfile,hosts,create-simple-devops-project.yml,create-simple-devops-image.yml
* **Remote directory :** /opt/docker (files will deploy here)
* **Applay and save**
* **Build – console output successful**

* After successful build

4 files deploying to ansible server 🡪 opt/docker 🡪 ls –l

If you want to multiple deployments give multiple IP address in HOSTAS file

* On docker host 🡪 opt 🡪 docker image
* On ansible srver 🡪 docker 🡪 docker image (or) docker ps

**Deploy maven project to dockerhub**

* Go to the Jenkins 🡪 select project 🡪 configure 🡪 add server
* **Source file** : webapp/target/\*.war
* **Remove prefix** : webapp/target
* **Remote directory** : /opt/docker
* **Exec command** : dockerfile,hosts,create-simple-devops-project.yml,create-simple-devops-project.yml
* Apply and save
* Build success
* Go to ansible server 🡪 docker image 🡪 webapp.war (it will add new file)
* Go to docker server 🡪 docker image 🡪 it will show the list

After creating the images automatically deleted images from the docker server but it will be there on ansible server

Go to the dockerhub and give refresh the dockerhub

* Copy the docker public IP 🡪 paste on google 🡪 docker public IP:8080/webapp 🡪 we can the output of source code

**Webhook configuration**

* Go to Jenkins 🡪 select job 🡪 configure
* Build trigger 🡪 github hook trigger for GITScm polling 🡪 save
* Go to the github 🡪 select the code 🡪 settings 🡪 webhooks 🡪 add or edit
* <http://jenkins> IP:8080/github-webhook/
* Update webhook
* Go to the Jenkins server 🡪 cd /opt/
* Git clone paste git URL
* Ls (here we can get project)
* Cd /opt/project name/webapp/scr/main/webapp
* Ls 🡪 index.jsp
* Vim index.jsp
* Change code 🡪 save
* Cat index.jsp
* Git commit –a –m “update”
* Configure email and name
* Git commit –a –m “update”

Now update successfully done

* Git status
* Git push
* Give username and password of git

Then automatically new job will start on Jenkins

* And you can check on ansible server
* Docker image 🡪 on ansible server
* Docker ps 🡪 on docker server
* Successfully changed source code
* http://<docker-public-IP>:/8080/webapp

## ****dockerfile****

# Pull tomcat latest image from dockerhub

From tomcat:latest

# Maintainer

MAINTAINER "Amit Kumar Gupta"

# copy war file on to container

COPY ./webapp.war /usr/local/tomcat/webapps

## ****create-simple-devops-image.yml****

|  |
| --- |
| --- |
|  | - hosts: all |
|  | become: yes |
|  |  |
|  | tasks: |
|  |  |
|  | - name: create docker image using war file |
|  | command: docker build -t simple-devops-image:latest . |
|  | args: |
|  | chdir: /opt/docker |
|  |  |
|  | - name: create tag to image |
|  | command: docker tag simple-devops-image ameintu/simple-devops-image |
|  |  |
|  | - name: push image on to dockerhub |
|  | command: docker push ameintu/simple-devops-image |
|  |  |
|  | - name: remove docker images from ansible server |
|  | command: docker rmi tomcat simple-devops-image:latest ameintu/simple-devops-image |
|  | ignore\_errors: yes |

## ****Hosts****

localhost

10.128.0.8

## ****create-simple-devops-project.yml****

|  |
| --- |
| --- |
|  | - hosts: all |
|  | become: yes |
|  |  |
|  | tasks: |
|  |  |
|  | - name: stop current running container |
|  | command: docker stop simple-devops-container |
|  | ignore\_errors: yes |
|  |  |
|  | - name: remove stopped docker container |
|  | command: docker rm simple-devops-container |
|  | ignore\_errors: yes |
|  |  |
|  | - name: remove current docker image |
|  | command: docker rmi ameintu/simple-devops-image:latest |
|  | ignore\_errors: yes |
|  |  |
|  | - name: pull docker image from dockerhub |
|  | command: docker pull ameintu/simple-devops-image:latest |
|  |  |
|  | - name: creating docker container using simple-devops-image |
|  | command: docker run -d --name simple-devops-container -p 8080:8080 ameintu/simple-devops-image:latest |

ansible-playbook -i /opt/doker/hosts/opt/docker/create-simple-devops-image.yml--limit localhost;

ansible-playbook -i /opt/doker/hosts/opt/docker/create-simple-devops-image.yml--limit 3.95.211.207;

Source files: \*\*/\*

Remove prefix: empty

Remote directory: empty

Exec command: pwd

Post-build Actions

Under transfer set

Source files: \*\*/\*

Remove prefix: empty

Remote directory: /var/www/html

Exec command: pwd