**DevOps Project**

**INSTALLING JAVA**

java -version

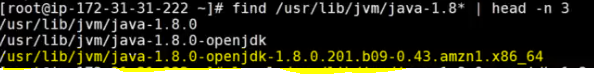
to see the version of java

yum remove java-1.7.0\*

to remove the default java and provide the version

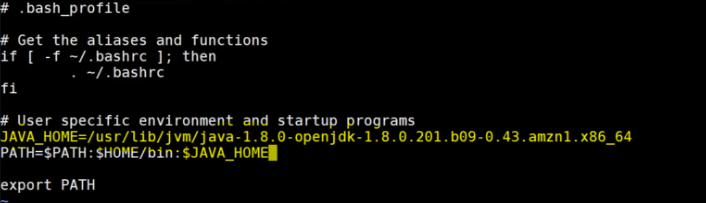
yum install java-1.8\*

find /user/lib/jvm/java-1.8\* | head -n 3



copy the specific path from over here

vi ~/.bash\_profile



paste the above copied path

echo $JAVA\_HOME

**INSTALLATION OF GIT**

yum install git -y

**INSTALLATION OF MAVEN**

cd /opt

here we can download the maven from website

https://maven.apache.org/download.cgi

in that we need to download bin.tar.gz

right click and copy the link address

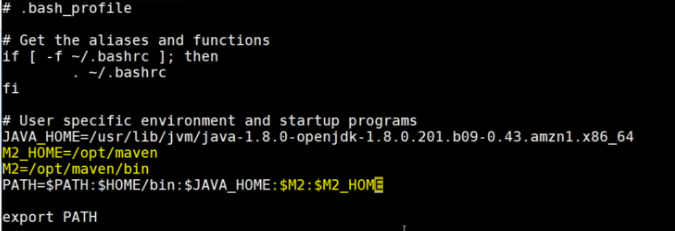
wget <link>

it will download the file

tar xzvf apache-maven-3.8.1-bin.tar.gz

it will unzip the file

vi ~/.bash\_profile



let’s add the path for getting maven default

echo $M2

we should see the path

mvn –version

**INSTALLATION OF TOMCAT**

cd /opt

let’s follow the steps in the website for installing

select tomcat 8

right click on tar.gz and copy the link address

wget <link>

it will download the file

tar xzvf <tomcatfilename>

it will unzip the file

cd tomcat/bin

./startup.sh

it is to start the tomcat server

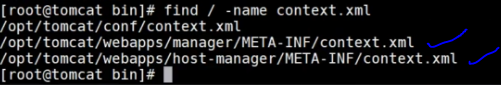
./shutdown.sh

it is to start the tomcat server

here it will not allow us to login

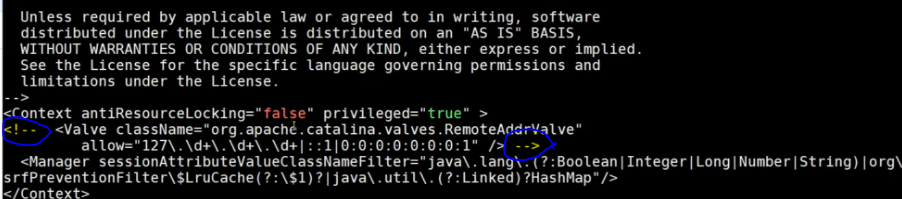
find / -name context.xml

it will search the file in the system



in that we should edit the two-given path

vi /opt/tomcat/webapps/manager/META-INF/context.xml



here we should comment it out but given symbol, because here it is mention to access only from the local host system…

vi /opt/tomcat/webapps/host-manager/META-INF/context.xml

here also same comment should be added like before

then restart the tomcat services ( need to go with bin path)

./shutdown.sh

./startup.sh

now after restart we need to open tomcat in website but here, we need a password

cd /opt/tomcat/conf

vi tomcat\_users.xml

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

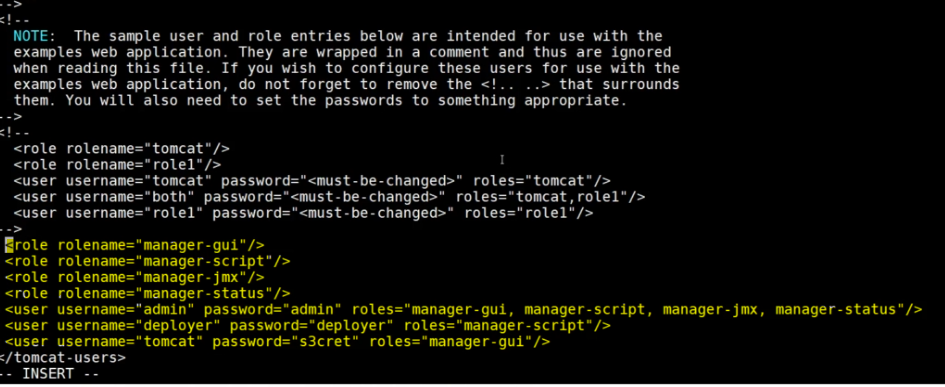
<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>

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

here we need to add the user’s passwords and roles



now restart the server and run the website

now we can use above username and password in website

**INSTALLATION OF DOCKER**

yum install docker

service docker start

here we need to create user

useradd dockeradmin

passwd dockeradmin

cat /etc/group

here we can see all the groups

usermod -aG docker dockeradmin

here we are adding user dockeradmin to the group docker

**INSTALLATION OF ANSIBLE**

yum install python

yum install pip

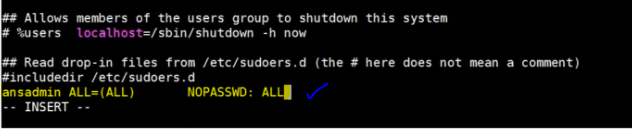
pip install ansible

mkdir /etc/ansible

user add ansadmin

passwd ansadmin

vi sudo



we should add in last line, by adding user here it will add user to sudo

yum install docker

service docker start

usermod -aG docker ansadmin

password authentication should be changed to yes like we have done in docker with Jenkins in below

su – ansadmin

switch to ansadmin user

**INSTALLATION OF KUBERNETES**

Here for kubernetes we need 2 virtual CPU system. In AWS need to launch new instance with type t2.medium

get into the instance and

apt-get update

sudo apt-get install -y conntrack

-First, we should install conntrack because it is related to network-

<https://www.radishlogic.com/kubernetes/running-minikube-in-aws-ec2-ubuntu/>

-here in this link, we have all instructions to install minikube-

curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s [https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl](https://storage.googleapis.com/kubernetes-release/release/stable.txt%60/bin/linux/amd64/kubectl)

-it will download & install kubectl which it talks to API server and needed to copy both lines at once and paste in terminal-

chmod +x ./kubectl

-it will give the execute permission-

sudo mv ./kubectl /usr/local/bin/kubectl

-it will move kubectl to bin folder-

sudo apt-get update

sudo apt-get install docker.io

-It will install docker-

curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin/

-it will download the minikube-

minikube start --vm-driver=none

-it will start mini cube-

minikube status

**INSTALLATION OF JENKINS**

let’s follow the steps in the website for installing

after installing

service Jenkins status

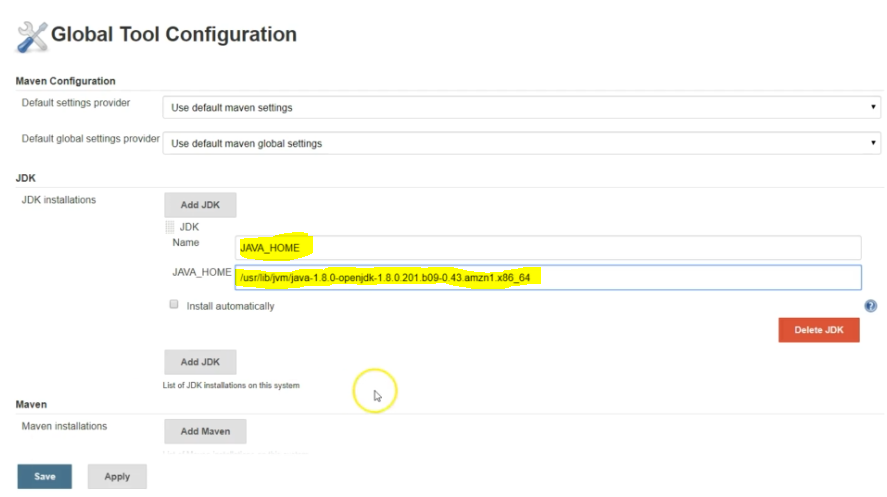
service Jenkins start

service Jenkins stop

<ip address>:8080

**Java with Jenkins**

In Jenkins – manage Jenkins – global tool configuration



add the path of java in it

next we need to add git to Jenkins, if plugin is not available need to add

**plugin - GitHub**

In Jenkins – manage Jenkins – global tool configuration

here we can see git after installation of git in jenkins

path will be automatically taken default…. no need to give path in it

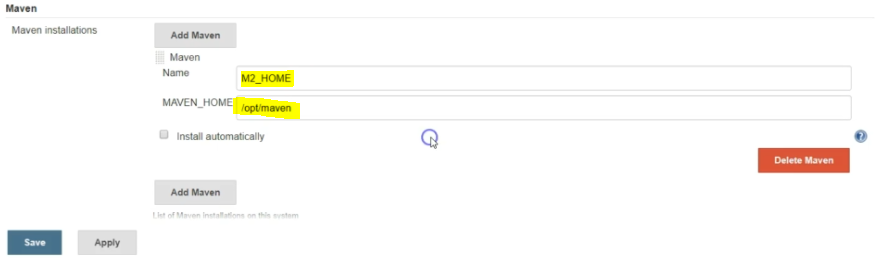
**Maven with Jenkins**

here we need to add maven plugin

**plugin - Maven Integration**

**plugin – Maven Invoker**

In Jenkins – manage Jenkins – global tool configuration



here we need to provide path

from here while doing job we should provide git repository link

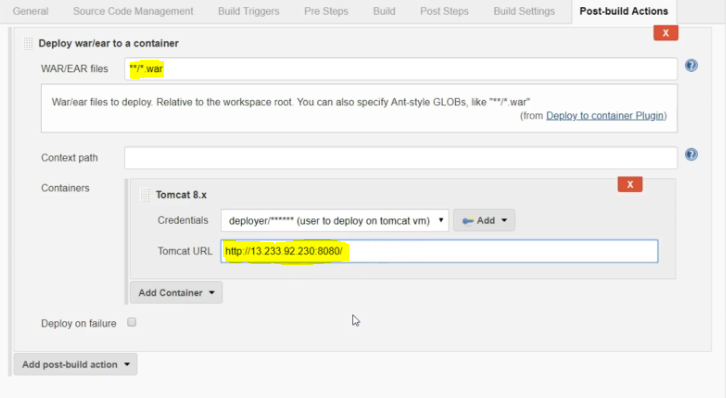
and build tab we should mention pom.xml and in goals we should give clean install package

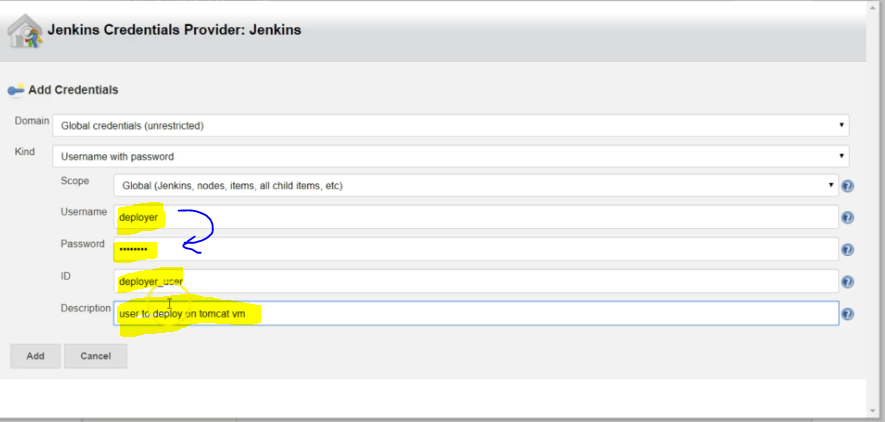
cd /var/lib/jenkins/workspace/

it is the workspace for our jobs in jenkins

**plugin – Deploy to container**

here after adding the plugin in post-build action here we can see **Deploy war/ear to container**





there are the credentials should be added

now we should build the job where the war file and application will be deployed in tomcat server

cd /opt/tomcat/webapps

here the war file and webapp will be deployed after building it

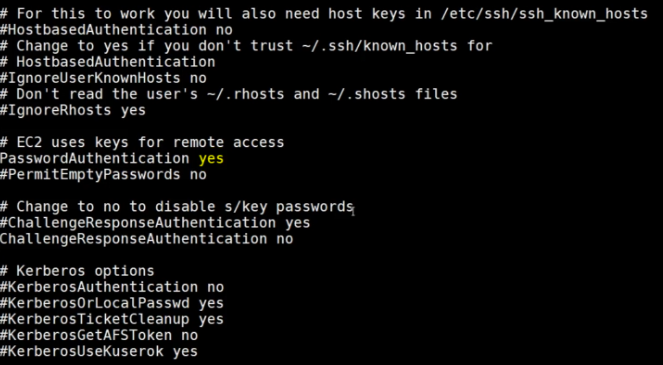
<Ip add>:8080/webapp

here in website we can check whether it is deployed or not by

**Docker with Jenkins**

**plugin – publish over ssh**

vi /etc/ssh/sshd\_config

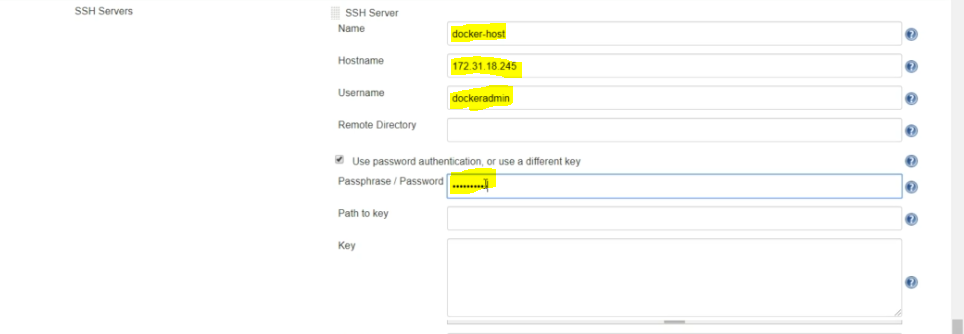


here we need to enable the password by typing yes

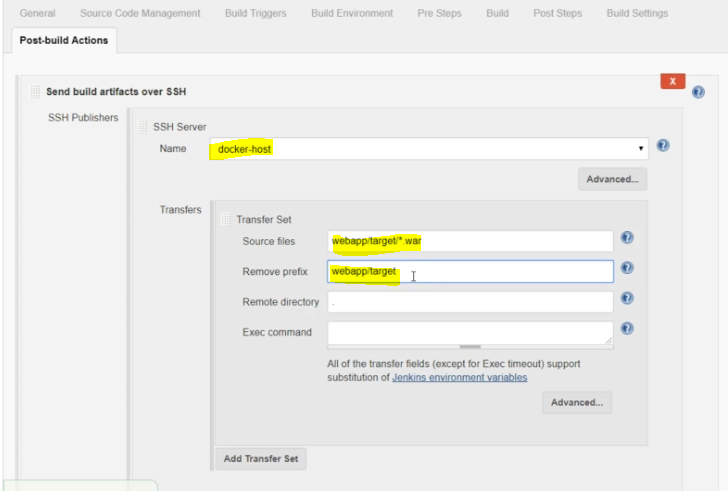
service sshd reload

In Jenkins – manage Jenkins – configure system

here we need to add ssh servers



here we need to give paasword of the dockeradmin which we have given



then it will deploy the war file to the docker host machine

cd /home/dockeradmin

in this path war file will be deployed

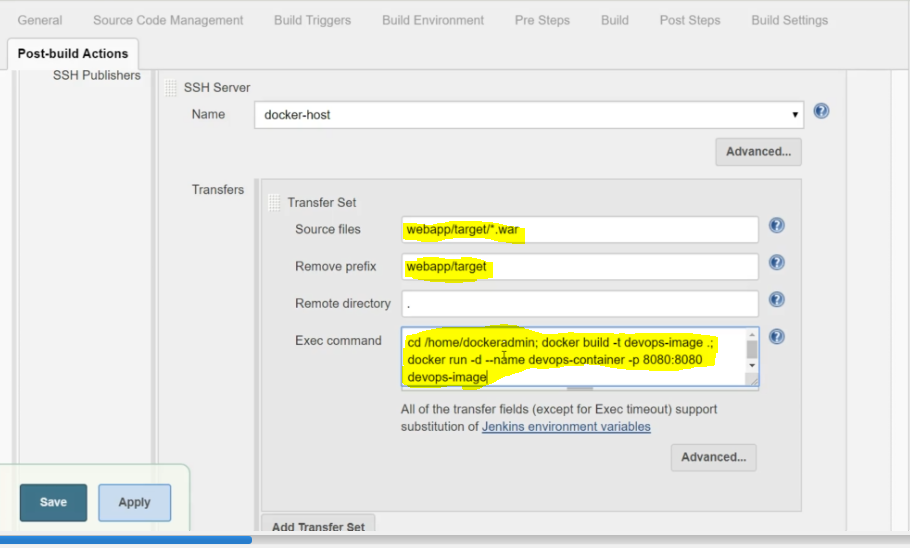
in that path we can write a Dockerfile

vi Dockerfile

FROM tomcat:latest

MAINTAINER Siva

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



now it will build the image from docker file and create a container and copy the deploy the file in the container as mentioned in the docker file

here the problem is it can’t deploy the war file if it is done in second attempt because first attempt the docker container was build and second time it can’t create a same named docker container again.

for rectifying this we will use ansible as a build tool

**Ansible with Jenkins**

2.

sudo su –

Cd .ssh

Vi authorized\_keys

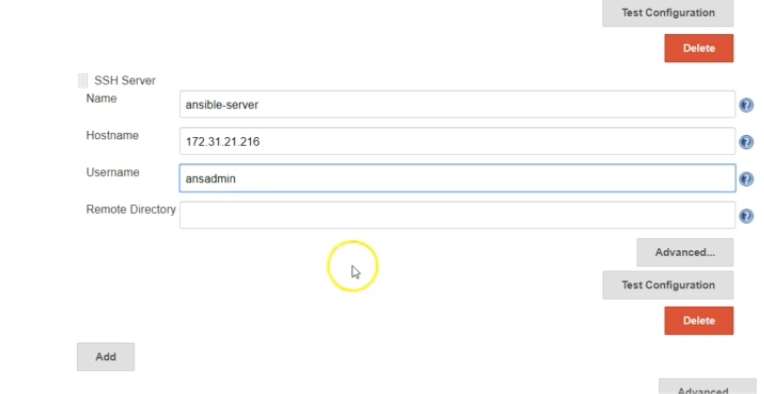
-paste the text in the starting line-

1.  
sudo su –  
ssh-keygen  
-press enter to take everything default-  
cd /root/.ssh  
cat id\_rsa.pub  
-copy the content-  
3.  
ssh <private ip address of host machine>  
-now I will be with host machine-

this will connect ansible host to docker host

exit

exit from docker server for now



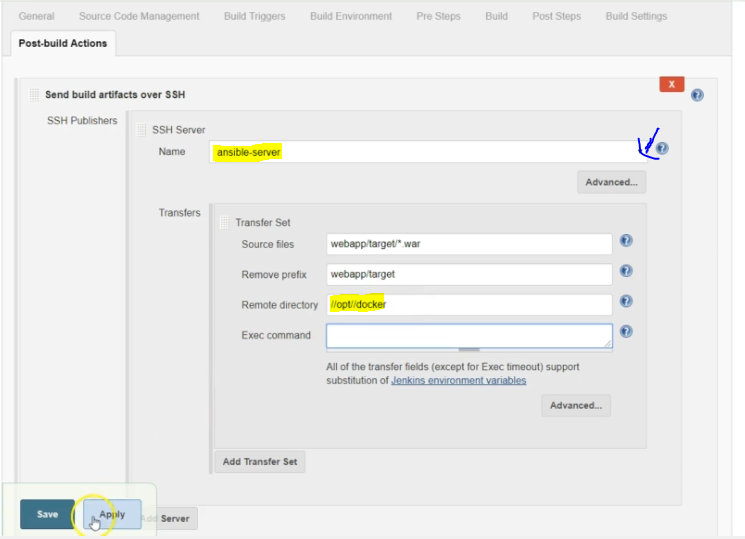
as we have done to docker we need to add the server for ansible

cd /opt

mkdir docker

sudo chown -R ansadmin:ansadmin /opt/docker

this to give full access for the given path



now build the job we can able see the war file in the above created docker directory

cd /opt/docker

vi Docker file

here write the same content written in Docker with Jenkins

vi hosts

localhost

<docker host private IP address>

docker login

we have to give username and password

let’s go to host system i.e. **docker host**

user add ansadmin

passwd ansadmin

usermod -aG docker ansadmin

su – ansadmin

here we should be with ansadmin user in docker host machine

now come to **ansible server**

vi create-simple-devops-image.yml

---

- hosts: all

  become: true

  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 yankils/simple-devops-image

  - name: push image on to dockerhub

    command: docker push yankils/simple-devops-image

  - name: remove docker image from ansible server

    command: docker rmi simple-devops-image:latest yankils/simple-devops-image

    ignore\_errors: yes

vi create-simple-docker-projet.yml

---

- hosts: all

  become: true

  tasks:

  - name: stop current running container

    command: docker stop simple-devops-container

    ignore\_errors: yes

  - name: remove stopped container

    command: docker rm simple-devops-container

    ignore\_errors: yes

  - name: remove docker image

    command: docker rmi yankils/simple-devops-image:latest

    ignore\_errors: yes

  - name: pull docker image from dockerhub

    command: docker pull yankils/simple-devops-images:latest

  - name: creating container using simple-devops-image

    command: docker run -d –name simple-devops-container -p 8080:8080 simple-devops-image:latest

these yml scripts are the ansible-playbooks

if we need, we can run these playbooks manually also

ansible-playbook -I hosts create-simple-devops-image.yml –limit localhost

here it will run the playbook with yml script to hosts we have given

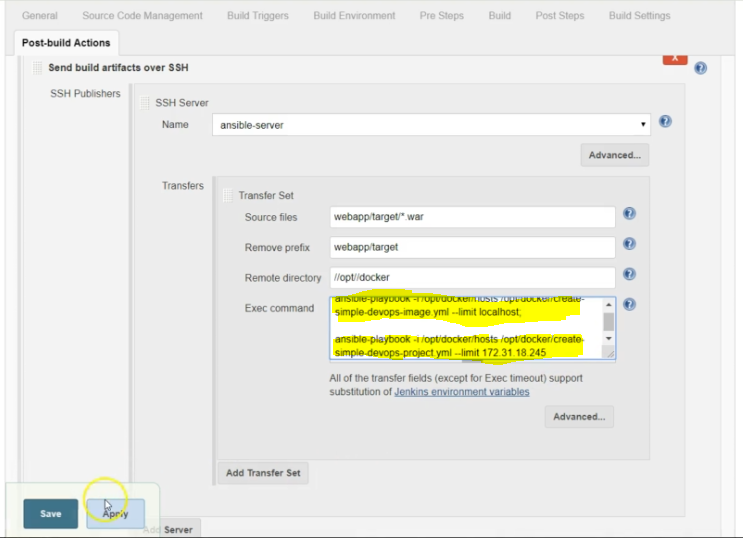
here limit is to it will run the playbook in the localhost only not the ip address we have given to host file

this create-simple-devops-image.yml should be run in localhost only so we have limited

ansible-playbook -I hosts create-simple-devops-project.yml -limit <private IP address>

this create-simple-devops-project.yml should be run in docker host machine only so we have limited to the specific ip of docker host machine

but these should be done automatic in Jenkins… so we can build CICD pipeline in jenkins



in Jenkins in post build option in exec command we need to run playbook… so it can be automatic

ansible-playbook -I /opt/docker/hosts /opt/docker/create-simple-devops-image.yml –limit localhost

ansible-playbook -I /opt/docker/hosts /opt/docker/create-simple-devops-project.yml –limit <private IP>

these two playbooks cmd should be kept in exec command in post build and we can build the job and this is automatic deploy in container

Sometime containers will not work properly, so build to containers will fail…. so kubernetes will be play main role for maintaining containers so deploying will not fails in kubernetes than ansible

**Kubernetes with Jenkins**

let’s create deploy.yml and service.yml

vi valaxy-deploy.yml

---

apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2

kind: Deployment

metadata:

  name: valaxy-deployment

spec:

  selector:

    matchLabels:

      app: valaxy-devops-project

  replicas: 2 # tells deployment to run 2 pods matching the template

  strategy:

    type: RollingUpdate

    rollingUpdate:

      maxSurge: 1

      maxUnavailable: 1

  template:

    metadata:

      labels:

        app: valaxy-devops-project

    spec:

      containers:

      - name: valaxy-devops-project

        image: yankils/simple-devops-image

        imagePullPolicy: Always

        ports:

        - containerPort: 8080

vi valaxy-service.yml

---

apiVersion: v1

kind: Service

metadata:

  name: valaxy-service

  labels:

    app: valaxy-devops-project

spec:

  selector:

    app: valaxy-devops-project

  type: LoadBalancer

  ports:

    - port: 8080

      targetPort: 8080

      nodePort: 31200

**let’s go to ansible server**

1.  
sudo su –  
ssh-keygen  
-press enter to take everything default-  
cd /root/.ssh  
cat id\_rsa.pub  
-copy the content-  
3.  
ssh <private ip address of host machine>  
-now I will be with host machine-

2.

sudo su –

Cd .ssh

Vi authorized\_keys

-paste the text in the starting line-

exit

let’s come to ansible server again

cd /opt

mkdir kubernetes

cd kubernetes

vi hosts

[ansible-server]

local host

[kubernetes]

<kubernetes public ip address>

sudo chown -R ansadmin:ansadmin /opt/kubernetes

this directory will get privilege rights to ansadmin user

vi Dockerfile

FROM tomcat:latest

MAINTAINER Siva

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

vi create-simple-devops-image.yml

---

- hosts: ansible-server

  become: true

  tasks:

  - name: create docker image using war file

    command: docker build -t simple-devops-image:latest .

    args:

chdir: /opt/kubernetes

  - name: create tag to image

    command: docker tag simple-devops-image yankils/simple-devops-image

  - name: push image on to dockerhub

    command: docker push yankils/simple-devops-image

  - name: remove docker image from ansible server

    command: docker rmi simple-devops-image:latest yankils/simple-devops-image

    ignore\_errors: yes

vi kubernetes-valaxy-deployment.yml

---

- name: Create pods using deployment

  hosts: kubernetes

  # become: true

  user: ubuntu

  tasks:

  - name: create a deployment

    command: kubectl apply -f valaxy-deploy.yml

  - name: update deployment with new pods if image updated in docker hub

    command: kubectl rollout restart deployment.v1.apps/valaxy-deployment

vi kubernetes-valaxy-deployment.yml

---

- name: create service for deployment

  hosts: kubernetes

  # become: true

  user: ubuntu

  tasks:

  - name: create a service

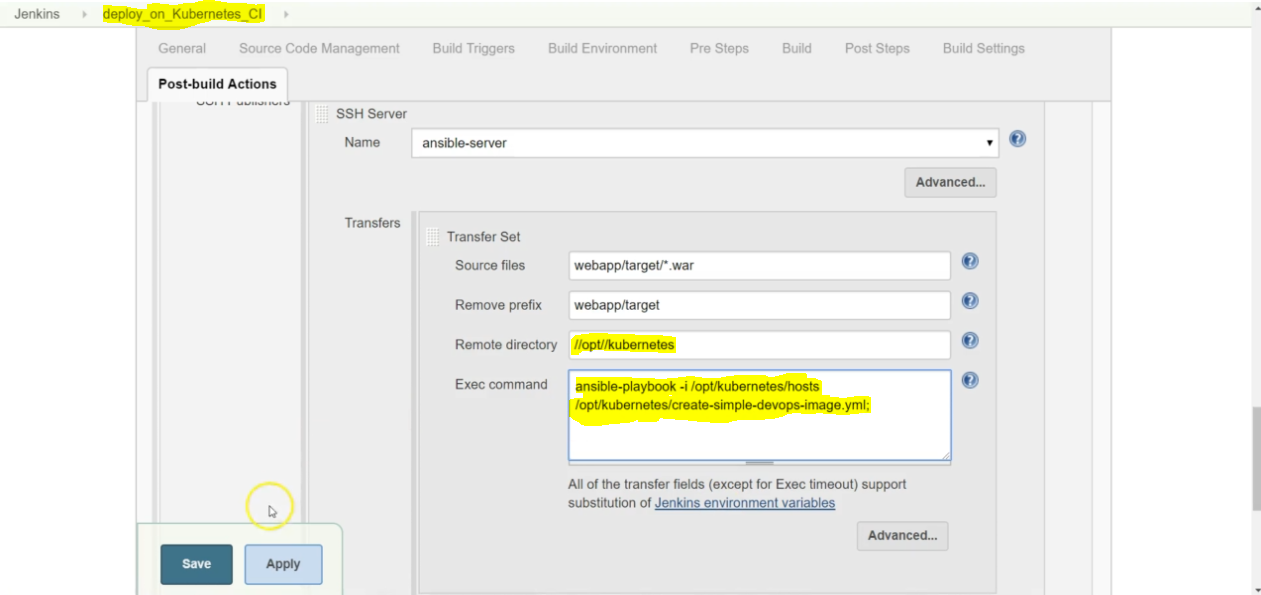
    command: kubectl apply -f valaxy-service.yml

in this directory we need total 5 files

host, Dockerfile, create-simple-devops-image.yml, kubernetes-valaxy-deployment.yml, kubernetes-valaxy-deployment.yml

**Jenkins CI Job**: here it will create image using war file

* It will copy war file to the directory
* from docker file in the path it will build an image and deploy war file in it
* it will tag a git repository to the image
* it will push the image to GitHub repository
* it will remove docker image from ansible server



Remote directory - //opt//kubernetes

exec command – ansible-playbook -I /opt/kubernetes/hosts /opt/kubernetes/create-simple-devops-image.yml

here we should give

* git source
* poll SCM
* Build with pom.xml and goals as clean install package
* post build action as shown in the pic
* After creating Jenkins CD job again we need to come to CI Jobs, need to go post build and need to opt build another project and select CD job over here to get pipeline

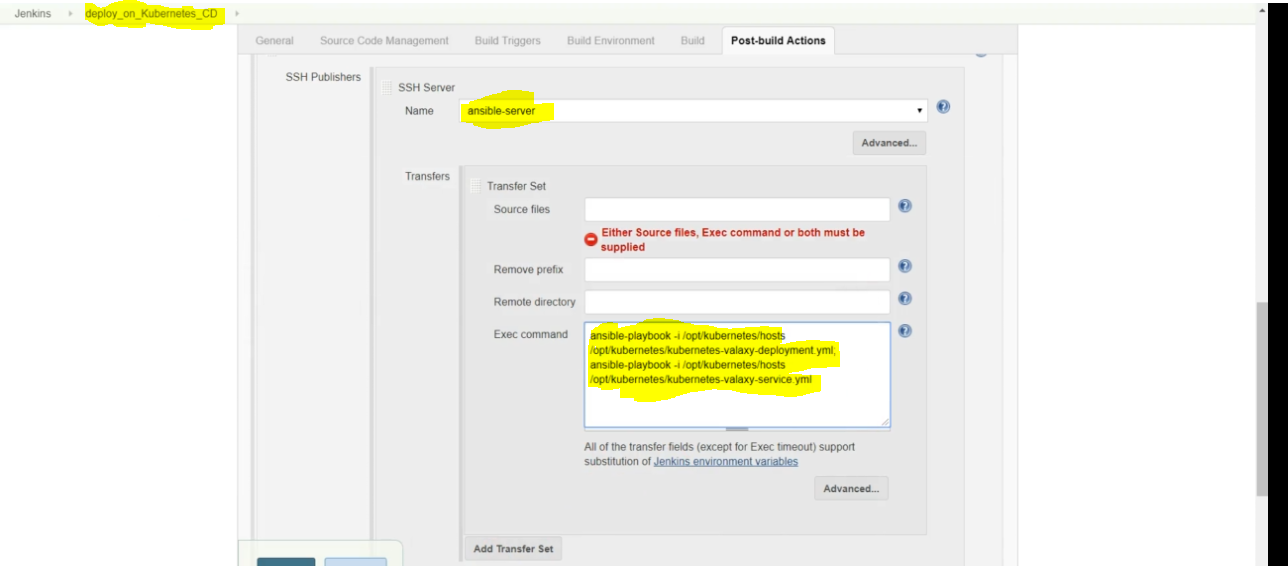
**Jenkins CD job**: it will deploy in the container of kubernetes cluster

here it will run the playbook of all yml which we mentioned before and deploy the application in kubernetes

this job will run automatically after CI job

here no need to give any GitHub, pollscm and any build triggers

only we will run playbooks over here



Exec command:

ansible-playbook -i /opt/kubernetes/hosts /opt/kubernetes/kubernetes-valaxy-deployment.yml;

ansible-playbook -i /opt/kubernetes/hosts /opt/kubernetes/kubernetes-valaxy-service.yml

now we can check the application deployed or not by going to website

< Private IPv4 DNS of kubernetes>:<port number of container>