Docker setup:

Create new Ec2 instance for Docker server

>Sudo su -

>hostname docker-host

>sudo su -

>yum install docker

>docker --version

>docker ps

(this command using to list the running docker containers)

>service docker start

>service docker status

>docker ps

(no containers because it is fresh setup)

so we need to create our own containers.

docker containers created out of docker images.

Docker images created from docker file.

Instailly we need to write docker file. From docker file we need to create docker image.

From docker image we need to create docker containers.

Instead of … we can use predefined images are available in docker hub.

Docker hub is registery which contains docker images which are created by diff users.

Note: if u don’t have docker images and docker file then we can pull from docker hub. These images created by officialy image owners.

HUB.DOCKER.COM

Here we can create public docker images and private images also.

Public registery

Private registery

We can pull some docker images from docker hub.

In docker hub search tomcat image:

First pull image then we can create container.

go to Docker server(command)

>docker image ls

>docker pull tomcat:latest

>docker images

now we can create container from image:

>docker run –-name tomcat-container –p 8080:8080 tomcat:latest

then open browser enter url :

ipv4 of docker:8080

ctl-c

>docker ps

>docker ps -a

to remove a docker container

so

>docker rm containerid

create new container

> docker -d run -–name tomcat-container –p 8080:8080 tomcat:latest

here d stands detachment.so it wont display in frontend.

>docker ps

then open browser enter url :

docker ipv4:8080

INTEGRATING DOCKER WITH JENKINS:

Open Jenkins browswer

Click manage Jenkins

Click manage plugins

Select available

Search publish over ssh

Select install without restart

Note: we need to add docker server credentials so we wil create docker user

Open docker server(command)

>useradd dockeradmin

>passwd dockeradmin

Note: user should be part of docker group

>cat /etc/group

>usermod -aG docker dockeradmin

(adding user to the docker admin group)

>id dockeradmin

so we need to give docker credentials to Jenkins server

open Jenkins browswer

select manage Jenkins

select configure system

select publish over ssh

click on ADD

name:docker-host

hostname:ipaddress

(go to docker command run command: ip adr then copy ip address and past in host name)

username:dockeradmin

remote directory:

click on advanced

select use password authentication

password: dockeradmin

click on test connection.

If connection fails then

Enable password authentication

In docker server(command)

>vi /etc/ssh/sshd\_config

/password

passwordauthentication yes

:wq

>service sshd reload

then back to Jenkins browser

then click on TEST CONNECTION

success:

JENKINS jobs to copy the ARTIFACTS to Docker host:

Open JENKINS Browser:

Newitem

Itemname:deploy on docker host

Copy from : deploy on tomcat (previous job)

Click ok

Description: Deploy on Docker

Git same

Poll scm no need

Build is same

Post build actions : no need

Post steps: select send build artifactover ssh.

NOTE: So it is transfer to target server through SSH

Name: docker-host

Source files: webapp/target/\*.war

Open jenkin server(command)

Note: Here get path of war file from jenkins which u want to deploy

**Apply & Save**

Before building this job

Open docker server(command) check what is there docker admin home:

>ls

>pwd

>whoami

then open Jenkins browser then click on buildnow

once build success open docker server(command)

>ls

>cd webapp/target/

>ls

(YOU can see WAR file)

Now we need to copy this war file to docker container so we need to create first

Docker file.

Open docker server(command):

>ls

>vi Dockerfile

FROM tomcat:latest

MAINTAINER BV Satya

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

:wq

>docker ps

>ls

Dockerfile

>docker build –t devops-project **.**

(build image)

( . means check the dockerfile present location and build the docker image)

Check now docker images list

>docker images

Note: now in docker image has War file . now we can create container from image.

>docker run --name devops-container -p 8080:8080 devops-project

Now open browser copy the ipv4 of aws docker server

Ipv4:8080/webapp

Then go docker command ctl-c

>docker ps -a

>docker rm containerid

So now removed container and now I want

To create container again.

>docker run -d --name devops-container -p 8080:8080 devops-project

>docker ps

>docker ps -a

(it will show all containers which stopped or running)

Now open browser copy the ipv4 of aws docker server

Ipv4:8080/webapp

Note: previously we seen how to copy artifact on docker host

**Now how to deploy war file into docker container using JENKINS:**

Open Jenkins browser

Click on newitem

Itemname: deploy on container

Copy from: deploy on docker

Click on OK.

Description : deploy on container

Git: same

Build :same

Post-build actions:

Name: same

Source files: same

Remove prefix:same

Exec command: cd /home/dockeradmin; docker build –t devops-image .;docker run

–d --name devops-container –p 8080:8080 devops-image;

(above syntax is using to go dockeradmin home then build the image and create container from image.)

Then apply and save.

Before run build first check any docker images and containers are there.

First open docker server(command)

>pwd

>ls

>rm - rf webapp.war

>docker ps -a

>docker images

>docker rmi docker-project

Now open Jenkins browser

Then click build now

Note: once run build then it move war file

Into docker container from image

>docker ps

>docker ps –a

>docker images

>docker ps -a

Then open browser

**Docker server ipv4:8080/webapp**

**Note:Assume there is code changes and**

**Build it then new war file create and**

**Deploy into container.**

**Now click buildnow**

**Check in console output execution**

**Note: We cant deploy to new container with same name.**

**So Jenkins cant manage deployments properly.**

**So to address this issues we will use Ansible is deployment tool.**

**ANSIBLE:**