**Basics Docker Commands**

Login into instance & open terminal- $ sudo su

To update our machine-

yum update –y

To install Docker –

yum install docker –y

To check docker installation –

which docker (o/p - /usr/bin/docker)

To check version of docker-

docker –version OR docker –v

To check service status-

Docker info OR service docker status

To start docker service-

Service docker start (o/p- **active and running)**

To See all images present in your local machine

# docker images

To see only running container

# docker ps

(go to hub.docker.com & search for required services in search bar)

To see all containers

# docker ps -a (P- Process & S- Status)

To fine out images in docker-hub

# docker search jenkis/ubuntu/centos

To Download image from dockerhub to local machine

# docker pull jenkis/ubuntu/centos

To run and create service from dockerhub directly- Eg.1

docker run –it ubuntu /bin/bash

To check operating system of container – ( u must be in container)

Cat /etc/os-release (o/p- version, name)

To run and create service from dockerhub directly- Eg.2

docker run –it centos /bin/bash

Cat /etc/os-release (o/p- version, name)

To give name to container

# docker run –it --name irfan ubuntu bin/bash

To start container

# docker start irfan (Container name)

To go inside container

# docker attach irfan

To stop container

# docker stop irfan (Container name)

To delete container

# docker rm irfan

**Dockerfile components & diff command**

Login into AWS account and start ur EC2 instance Access it from putty

Now we have to create container from our own image.

Therefore create one container First

# docker run –it –name irfancontainer ubuntu /bin/bash

# cd tmp/

Now create one file inside tmp directory

# touch myfile

Now if you want to see the difference between the base image & changes on it then

# docker diff irfancontainer

O/P- C /root

A /root/.bash\_history

C /tmp

A /tmp/myfile

Now create image of this container

# docker commit irfancontainer updateimage

# docker images

Now create container from this image

# docker run –it –name newcontainer updateimage /bin/bash

Root @id # ls

# cd tmp

Tmp # ls

o/p – myfile (you will get all files back)

**Image creation from dockerfile :**

Create a file named Dockerfile

Add instructions in dockerfile

Build dockerfile to create imag

Run image to create container

# vi Dockerfile (Add below contain in vi)

FROM ubuntu

RUN echo “I am Dimond” > /tmp/testfile

To create image out of dockerfile

# docker build –t test . (after this command all command will execute which given in vi Dockerfile)

# docker ps –a

# docker images

Now create container from the above image

# docker run –it -- name testcontainer test /bin/bash

# ls

# cd tmp/

# ls (o/p – testfile)

# cat testfile ( o/p - I am Dimond)

# exit

Now more command in Dockerfile :

# vi Dockerfile

FROM ubuntu

RUN echo “I am devops Engineer” > /tmp/testfile

ENV myname irfan

COPY testfile1 /tmp

ADD test.gz /tmp

Esc- :wq

Make some file as per given in dockerfile

# touch testfile1

# ls

# touch test

# ls

# tar –cvf test.tar test

# ls  
# gzip test.tar

# ls

# rm –rf test (test file will delete)

# ls (o/p – test.tar.gz)

# docker build –t newimage .

(o/p- all 6 command in dockerfile will execute successfully)

To make container from this image :

# docker –it –name newcontainer newimage /bin/bash

(o/p we directly run in tmp file: root @ 63fg56g880:/tmp)

# ls

# cat testfile (o/p- I am devops engineer)

# echo $myname – irfan

# exit from container

(To practice more take new code from google and add it in vi dockerfile)

**VOLUME**:

**Creating Volume from Dockerfile**

If new machine taken then – sudo su

# yum update –y

# yum install docker –y

# service docker start

# service docker status ( o/p- **Active and running**)

# touch file1 file2

# ls (o/p- file1 file2)

# vi Dockerfile

FROM ubuntu

VOLUME [“/myvolume”]

Escape- :wq

Make image from this dockerfile –

# docker build –t myimage . ( Make sure dot (**.**) should be given after myimage)

o/p – all command run successfully from image.

# docker images (o/p- ubuntu & myimage)

Now make new container from this image –

# docker run –it –name container1 myimage /bin/bash

(Now we come in new container)

# ls (o/p – shown myvolume )

# cd myvolume

# touch filex filey filez

# ls (o/p - filex filey filez)

Exit from container

Now we will share myvolume to new container so we will get same file filex filey filez in new container.

# docker run –it - - name container2 - - privileged=true - -volumes-from container1 ubuntu /bin/bash

(o/p – ls – myvolume. Cd myvolume. ls - filex filey filez)

# exit

# docker start container2

# docker attach container2

# ls (o/p- myvolume)

# cd myvolume

# ls (o/p - filex filey filez)

**VOLUME create by command**

# docker run –it - - name container3 -v /volume2 ubuntu /bin/bash

# ls (o/p – volume2)

# cd volume2

# touch vol1 vol2 vol3

# ls - vol1 vol2 vol3

Now make other new container –

# docker run –it - - name container4 - -privileged=true --volumes-from container3 ubuntu /bin/bash

# we are in new container- ls- volume2

# cd volume2

# ls - vol1 vol2 vol3

# touch irfanfile ( to check this file in container3)

# exit

# docker start container3

# docker attach container3

# ls- volume2

# cd volume2 (o/p – irfanfile)

# exit

Now we share volume with HOST to container –

How to create Host and container mapping –

# pwd

# cd . .

# ls – ec2-user

# cd ec2-user

# ls – (filename exist on ec2-user)

Now create new container –

# docker run –it - - name hostcont –v /home/ec2-user:/sayyad - - privileged=true ubuntu /bin/bash

Now u r in new container hostcont- do ls here

# ls (o/p – sayyad file)

# cd sayyad

# ls (o/p – same file found as present on ec2-user)

# Exit

# ls (o/p – same file found which present in hostcont)

# docker start hostcont

# docker attach hostcont

# cd sayyad/

# touch xyz xxx rst

# exit

# ls (o/p - xyz xxx rst)

**Some Other Docker basics command**

# docker volume ls

# docker volume create (volumaname)

# docker volume rm (volumaname)

# docker volume prune (It removed all unused docker volume)

# docker volume inspect (Containername)

**Docker Port Expose :**

Login into AWS account and create one linux instance

Now go to putty & Login as ec2-user

# sudo su

# yum update –y

# yum install docker –y

# docker -- version

# service docker start

# service docker status ( **Active and Running** )

# docker run –td - - name techserver –p 80:80 ubuntu

# docker ps

# docker port techserver

o/p – 80/TCP ->> 0.0.0.0/80

# docker exec –it techserver /bin/bash

# apt-get update

# apt-get install apache2 –y

# cd /var/www/html

# echo “ I am devops engineer” >index.html

# service apache2 start

Now take public ip of instance and put in chrome (It will show massage which was put in container)

If we see our massage from container that means anyone can access through internet.

Now we take other example- here we create Jenkins container

# docker run –td - - name myjenkins –p 8080:8080 jenkins/Jenkins

( This command will create and run new container myjenkins and pulling all repo from Dockerhub for jenkns)

Now copy public ip from instance and paste in chrome & add 8080 port it will show Jenkins page as it was configure in jenkin container.

**How to push docker image in dockerhub :**

Login into AWS account and create one linux instance

Now go to putty & Login as ec2-user

# sudo su

# yum update –y

# yum install docker –y

# docker -- version

# service docker start

# docker run –it ubuntu /bin/bash

Now create some files in container.

# touch file1 file2 test1 test2 xyz

Now create image of this container.

# docker commit container1 image1

Now create new account in hub.docker.com ( Create username & password)

Now go to Ec2 instance

# docker login ( enter your username and password)

Now give tag to your image

# docker tag image1 docker id or username/newimage- any name which will display in dockerhub)

# docker push docker id/ newimage

Now you can see this image in dockerhub account

Now create one instance in Mumbai region and pull image from hub. Again run below command on new Mumbai region machine.

# sudo su

# yum update –y

# yum install docker –y

# docker -- version

# service docker start

# docker pull docker id/newimage (this will pull image from hub to Mumbai machine)

Now make container from this image and see the shared files.

# docker run –it - - name mycon docker id/newimage /bin/bash ( We will get all files from this image)

To stop all running containers :

# docker stop $ (docker ps –a –q)

To delete all stoped container :

# docker rm $( docker ps –a –q)

To delete all images :

# docker rmi –f $ (docker images –q)