Aws Instance ip:- 3.13.79.163

ssh -i "ubuntuaws.pem" [ubuntu@ec2-3-13-79-163.us-east-2.compute.amazonaws.com](mailto:ubuntu@ec2-3-13-79-163.us-east-2.compute.amazonaws.com)

**Docker Role in Devops**

i. Build the Application

ii. Run the Application

iii. Ship the Application

Container is a system / A running state of a image is a container

Container using union file system

Task we do:

1- Manage Images

2- Manage Container

3- Manage Network

4- Docker Compose

5- Docker Volume

6- Docker image Creation

* Docker build
* Docker pull
* Docker run

Home Directory of Docker: ( /var/lib/docker )

**Install Docker In Instance**

Refer the link :-

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>

<https://hub.docker.com/>

First, update your existing list of packages:

$ sudo apt-get update

Next, install a few prerequisite packages which let apt use packages over HTTPS:

$ sudo apt install apt-transport-https ca-certificates curl software-properties-common

Then add the GPG key for the official Docker repository to your system:

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

Add the Docker repository to APT sources:

$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"

Next, update the package database with the Docker packages from the newly added repo:

$ sudo apt-get update

Make sure you are about to install from the Docker repo instead of the default Ubuntu repo:

$ apt-cache policy docker-ce

Finally, install Docker:

$ sudo apt-get install docker-ce

Start the services of Docker

$ sudo service docker start

Check that it's running:

$ sudo systemctl status docker

Give user to docker permission

$ sudo usermod -aG docker <username>

**Install Docker Image from hub.docker.com**

$ docker pull ubuntu ( by default latest version of image will pull )

$ docker pull ubuntu:18.04 (for particular version pull )

To view the running container

$ docker ps

To view the all container

$ docker ps -a

Create container with login in same container

$ docker run -ti --name=(name of container) (image id) /bin/bash

Ex:- $ docker run -ti --name=test 7565dg3546 /bin/bash

Create container with detachmode

$ docker run -tid --name=(name of container) (image id) /bin/bash

Ex:- $ docker run -tid --name=test 7565dg3546 /bin/bash

To inspect / View the container

$ docker inspect <container name>

Ex:- $ docker inspect test1

To inspect / View the Image

$ docker inspect <image name / ID>

Ex:- $ docker inspect ubuntu

Login to particular container

$ docker attach <container name>

Ex:- $ docker attach test1

Login with execute

$ docker exec -ti <container name> /bin/bash

Ex:- $ docker exec -ti test1 /bin/bash

To View the container logs

$ docker logs <container name>

Ex:- $ docker logs test1

To stop the container

$ docker stop <container name>

Ex:- $ docker stop test1

To come out from running container

Hold :- CTRL+P+Q

Remove the container

$ docker rm <container name>

Ex:- $ docker rm test

Remove forcefully container

$ docker rm -f <container name>

Ex:- $ docker rm -f test

Create container with port forwarding command

$ docker run -ti --name=(container name) -p 80:80 ( image id ) /bin/bash

Ex:- $ docker run -ti --name=test2 -p 80:80 75623cd1 /bin/bash

Make an Apache2 web-server in container

First make a container with port forwarding

$ docker run -ti --name=(container name) -p 80:80 ( image id ) /bin/bash

$ apt-get update

$ apt-get install apache2 -y

$ dpkg -l apache2

$ service apache2 status

$ service apache2 start

$ apt-get update

$ apt-get install vim ( install VI editor )

$ apt-get install curl ( its will view of localhost page )

Bridge Network is default network of container

To view the Docker network

$ docker network ls