Steps to create a container from existing container via images

Create a container with the name container1

1. Creating container in docker

docker run -it --name <container name> <base image> /bin/bash

docker run -it - -name container1 ubuntu /bin/bash

1. You will enter inside the container1 and there

touch file1 file2 file3 file4 file5

(touch creates a blank files with the name file1, file2…)

1. exit

TO see all containers

docker ps -a

TO see only running containers

docker ps

1. Now we have to create image of this container

docker commit <container1> <Image name>

docker commit container1 newimage

1. docker images

you will find the newimage also available in the set of existing images in docker engineclear

1. docker run -it - -name container2 newimage /bin/bash

(this will create a container 2 with custom image – newimage)

1. you will enter inside the container (after sno 2)

ls

1. you will see the files which you have created in container 1
2. Get into the container 🡪 docker attach “docker name”
3. For get into the container need to start first 🡪 docker start container1
4. For get into the container need to start first 🡪 docker stop container1
5. Docker ps – it will show only running containers.
6. Docker ps -a – it will show all the containers.
7. Get into the container 🡪 docker attach “docker name”
8. Run any comments like – apt update -y 🡪 it’ll update all the packages. And install any applications – apt install apache -y

Creating replicate, the container

A container can only be created from image

1. Docker commit container1 container1IMG 🡪 it’ll create the container1 replica with name as container1IMG
2. Docker images 🡪 It’ll show all the images in docker registry with container1IMG
3. Docker rmi <image ID> 🡪 to remove docker image
4. Docker run -it --name co2 co1img /bin/bash
5. Docker ps 🡪 it’ll show the co2 is created
6. Docker run -it --name webapp -p 8090:80 ubuntu /bin/bash 🡪 exposing the port 8090 of ec2 with port 80 of container

Commends of docker

* + - 1. From 🡪 which would the base image of the container, always on top of the docker file.
      2. Run 🡪 To execute the command inside the container, create a layer of image for us
      3. Maintainer 🡪 author of the docker file
      4. Copy 🡪 copy the files from the local system to the destination
      5. Add 🡪 it is similar to copy command, but it provides the feature to download the file from internet.
      6. Expose 🡪 if we want to expose the specific port

Docker composes.

Docker-compose build

Docker-compose up -d

To writing the docker file

Vi Dockerfile (Case sensitive)

FROM ubuntu

WORKDIR /tmp

RUN echo “Hello World”> /tmp/testfile

COPY file1.txt /tmp

Docker build -t newimage . (Make sure buildx is installed or not) 🡪 newimage will be created

Docker images 🡪 it will show the newimage

Docker run -it --name co1 newimage /bin/bash 🡪 creating a container with name co1 and custom image of newimage and execute the bash terminal in it.

Docker login 🡪 login to docker hub (username: \*\*\*\*, password: \*\*\*\*)

Docker push newimage 🡪 pushing the custom image into docker hub (It will access denied)

Docker tag <image> <username>/<filename> == docker tag newimage subbaramt99/firstfile 🡪 creating tag with username and image

That tag can able to push into docker hub

Docker push subbaramt99/firstfile 🡪 pushing the custom image with username tag then it wont show access denied it will pushed into docker hub

Overview of docker

1. Create a EC2 machine
2. Install a docker
3. Create a container (CO1)
4. Install application, create a file, anything we want to made done it here
5. create a custom image of container (CO1 img)
6. log in to docker hub ()
7. create a tag of username with custom image
8. push the image to docker hub