**Docker**

Docker has two components:

1. Docker client
2. Docker daemon

Docker image is as an object that contains an OS files system and an application.

1. Container is a stop version of images.
2. If you are a developer you can think of an image as a class.

**Find Version.**

1. docker version

**Find How Many Images.**

1. docker images or docker image ls

**Find How Many Containers Are Run or Show List of Docker Container**

1. docker container ls
2. docker ps
3. docker ps -a (-a used for additional information )
4. docker container ls

**Stop the Containers.**

1. docker stop (id)

i.e docker stop 93434a03cd2

**Start the Containers.**

1. docker start(id)

i.e docker start 93434a03cd2

**Go Back to Running Container Shell**

1. docker exec -it 93434a03cd2 (may be name or id)

**Removing the Container So It Must Be In Stop Mode If Container Is Not Stop So First Stop the Container and Then Remove.**

1. docker rm (id) i.e docker rm 93434a03cd2
2. docker rm alpine:latest
3. docker container rm 93434a03cd2

**Pull the Image from the Docker Hub**

1. docker pull ameenalam/static-web
2. docker image pull alpine:latest

**Run the Web Container in Browser**

1. docker run --name=web1 -p 3000:80 -d ameenalam/static-web
2. docker run -it aamirpinger/helloworld sh

(Iterative mode pulls the image if not find on local machine and creates a container and run it)

Examples:

* docker run --name=web2 -p 4000:80 -d nginx:latest
* docker run --name=web3 -p 4000:80 -d mystaticapp:v1

(-d stand for detach mean run the container and it will automatically run on background)

**Build the Images from the Dockerfile**

1. docker build -t mystaticapp:v1 .

* (-t this is tag line mean when you create a images you need to assign a name and it is called tag)
* (dot represent a docker file location)

**Note:**

If you want to run a container in background so press (ctrl PQ)

**Images pushing on DockerHub**

1. docker push infectiousdiseases

(If you run this command you will get an error access denied because it is official image)

**Images**

A container images is a lightweight, standalone, executable package of software that includes everything needed to run application.

* code
* Runtime
* System Tool
* System libraries
* Settings

1. Image becomes container when they run on Docker Engine.

* An image is a build time.
* Container is a run time.

1. Container is all about being fast and lightweight.
2. Docker images are stored in image registries. i.e. https://www.dockerhub

**Image Naming And Tagging:**

1. docker image pull <repository>:<tag>
2. docker image pull nginx:latest

**Containerizing an App Form Scratch.**

The process of containerizing an app looks like this:

1. Start with your application code.
2. Create a Dockerfile that describe your app, its dependencies, and how to run it.
3. Feed this Dockerfile into the docker image build command to create an image.
4. Create and run a container from the image.