Docker installation

```
sudo apt-get remove docker docker-engine docker.io containerd runc -y
sudo apt-get update
sudo apt-get install -y\
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
 sudo add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
   $(lsb_release -cs) \
   stable"
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io -y
```

Add the existing user to docker group to run docker commands without sudo

```
sudo usermod -aG docker ${USER}
```

Docker commands

```
docker pull <image>    # pulls an image from docker hub to your local machine

#example
docker pull ubuntu   # This pulls the  latest Ubunt image from Docker hub

# To start a container
docker run -it <image> # if the specified image doesn't exist, docker pulls the image from

# example
docker run -it ubuntu # This will start a container from ubuntu image

# To start a container and map a port
docker run -it -p 8080:80 ubuntu  # This will start a container from ubuntu image and also

# To start a container and ssh into it
docker run -it -p 8080:80 ubuntu /bin/bash # This will start the container with port mapping

# To map volume on to docker contaienr
docker run -it -p 8080:80 -v ${PWD}:/root ubuntu /bin/bash # This will map the presetn work

#To start a container in detached mode (running in the background)
docker run -d -p 8080:80 -v ${PWD}:/root ubuntu [command]
```

```
#example
docker run -d -p 8080:80 -v ${PWD}:/root ubuntu sleep 4800 # THis will start a container an


# To come out of container without killing it
Ctrl+p+q

# To enter into a docker container that was started interactively using '-it'
docker attach <container_id>  #Note: the container should be in runnnig status


# To see list of running containers
docker ps

# To see list of all containers (including stopped contianers)
docker ps -a


# TO start a container

docker start <container_id>

# To stop a container

docker stop <container_id>

# To restart a container

docker restart <container_id>

# To delete a container

docker rm <container_id> # Only stopped containers can be deleted.


# to list all images on the host
docker images


# to delete an image

docker rmi <image_name>
```

Create an image from a Container

Create a container from base image - Such as ubunut ssh into the container and make changes- install packages, add users, change files, add files Come out of the container `Ctrl + P +Q`

Now create an image out of the existing container

```
docker commit <container_id> <image_name>:<tag>
```

Example

```
docker commit 76d6re78 myimage:latest
```

The above command will create a new image with the name as

To check your images

```
docker images
```

This is locally available. To make it available for everyone

1. Go to Docker hub (hub.docker.com)
2. Create a Repository with the name you want for the image. ex: myimage
3. Now on your local machine tag your image

```
docker tag <image_name> <docker_hub_username>/<image_name>:<tag>
```

Example

```
docker tag myimage kbaddi/myimage:latest
```

In the above example `kbaddi` is the Docker hub username username

4. Log into Docker Hub

```
docker login -u <docker_hub_username>
```

example

```
docker login -u kbaddi
```

This will prompt for your DockerHub password, enter the password.

5. Push the image to Dockerhub

```
docker push <image_name>:<tag> <docker_hub_username>/<image_name>:<tag>
```

example

```
docker push myimage:latest kbaddi/myimage:1.0
```

Note: In the above example myimage repository should alread by there in Dockerhub account of `kbaddi`

Dockerfile

–> Create a container from a base image (Ubuntu) –> RUN apt-get update –> RUN apt-get install apache2 –> RUN apt-get install curl –> ADDED a file

Dockerfile is a set of instrucions which creates an image