Docker Installation - <https://docs.docker.com/engine/install/>

Docker Installation on ubuntu

sudo apt update -y

sudo apt install docker.io -y

sudo systemctl start docker (It will start the service on runtime)

sudo systemctl status docker

(q or ctrl+c)

sudo systemctl enable docker (It will enable the service on boot, when the server will reboot or will start from stopped state. The service will be running)

**To pull docker image from dockerhub**

sudo docker pull image

**To list docker images in docker host**

sudo docker image ls

**To search for a image on dockerhub**

sudo docker search ubuntu

**To deploy container and connect to it**

sudo docker run -it --name container\_name image\_name command\_to\_execute\_at\_the\_time\_of\_container\_deployment

sudo docker run -it --name mycontainer ubuntu bash

**To logout from container**

Ctrl + P and Ctrl + Q

**To List running container**

sudo docker ps

**To List all container (stopped & running)**

sudo docker ps -a

**To stop a container**

sudo docker stop container\_name/container\_id

sudo docker stop mycontainer

**To start the container**

sudo docker start container\_name/container\_id

**To Login/Connect to a running container**

sudo docker exec -it container\_name/container\_id command

sudo docker exec -it mycontainer bash

**To check container resource usage (all running container)**

sudo docker stats

**To check container resource usage of a specific container**

sudo docker stats container\_name/container\_id

**To remove the container from docker host**

sudo docker stop container\_name/container\_id

sudo docker rm container\_name/container\_id

**To remove the container forceful**

sudo docker rm -f container\_name/container\_id

**To remove a image which is currently not in use by a container**

sudo docker rmi image\_name/image\_id

sudo docker image rm image\_name/image\_id

**To rename a container**

sudo docker rename current\_container\_name new\_container\_name

**To map container port with host port to expose application over host network**

docker run -d -p 80:80 --name=web\_container nginx

docker run -it --name=webcontainer -p 80:80 ubuntu bash

**Deploying Static HTML Application in container**

docker run -it --name=webcontainer -p 80:80 ubuntu bash

apt update -y

apt install apache2 git -y

cd /var/www/html

git clone <https://github.com/devopstraining99/demo-app>

mv demo-app/\* /var/www/html

service apache2 start

**To create a image from container.**

docker commit container\_name image\_name:image\_tag

docker commit webapp myapp

**Create a docker hub account:**

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

**To Login to Dockerhub account using cli**

docker login

Username - docker hub account username

Password - dockerhub account password

Login Success

**To tag and push the image**

docker tag image\_name dockerhub\_username/repo\_name

docker tag myfirstimage gauravdemo06/myfirstimage

docker push gauravdemo06/myfirstimage

**To remove all stopped container and images from docker host**

docker system prune -fa

—-------------------------------

docker pull gauravdemo06/myappimage001

docker run -it --name mywebapp -p 80:80 gauravdemo06/myappimage001 bash

service apache2 start

**Docker file reference -** <https://docs.docker.com/engine/reference/builder/>

**Creating Image with Dockerfile**

touch Dockerfile

vim Dockerfile

FROM ubuntu

RUN apt update -y && apt install apache2 git -y

RUN cd /var/www/html && git clone <https://github.com/devopstraining99/demo-app>

&& mv demo-app/\* /var/www/html

ENTRYPOINT apache2ctl -DFOREGROUND

EXPOSE 80

docker build -t image\_name .

# Docker file

FROM ubuntu

RUN apt update -y && apt install apache2 -y

COPY data.zip /var/www/html # data.zip

ADD data.zip /var/www/html # It will extract data.zip file into /var/www/html

ENTRYPOINT apache2ctl -DFOREGROUND

CMD apache2ctl -DFOREGROUND (The command can be override by the user at the time of container deployment)

#ENV env\_variable\_name=env\_variable\_value

ENV os=ubuntu

EXPOSE 80