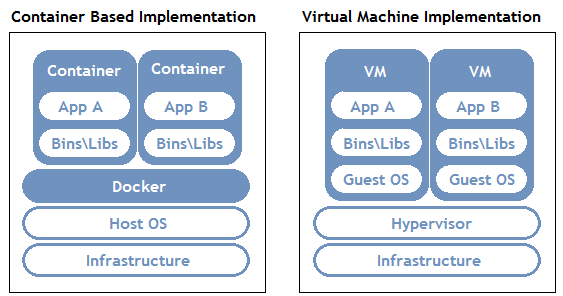
# What is docker?

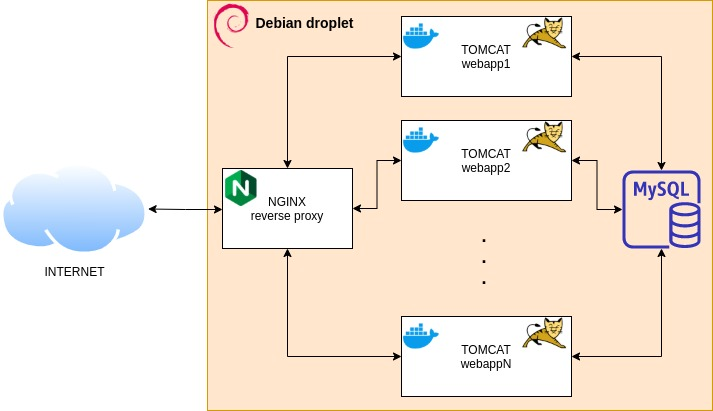
1. Docker and Virtual boxes work almost the same way
2. Virtual boxes need a separate OS for each box where all docker containers use the single host OS.
3. Unlike with virtual machines, each Docker container does not use a complete operating system kernel
4. Containers are isolated from one another and bundle their own software, libraries, and configuration files; they can communicate with each other through well-defined channels.

# Understanding containers

Container technology can be thought of as three different categories:

1. **Builder**: a tool or series of tools used to build a container, such as a distro builder for LXC, or a Dockerfile for Docker.
2. **Engine**: an application used to run a container. For Docker, this refers to the docker command and the docker daemon. For others, this can refer to the container daemon and relevant commands (such as podman.)
3. **Orchestration**: technology used to manage many containers, including Kubernetes and OKD.





# Difference between Docker vs VMware

1. Each VMware needs separate OS
2. VMware consume the extra OS resources like RAM, CPU, and disk space
3. All docker container works on a single OS
4. Docker is easy and less time consuming to deploy
5. Lighter and faster than VM
6. It’s open-sourced
7. Same docker container (virtual machine) runs at developer PC, SQL environment, and LIVE system, there is no hazard of compatibility issue at any environment. No dependency issue at all.
8. The developer did 90% work, we only have to just manage, maintain and scale these containers.

# How does docker work

1. First, install a ubuntu os within a docker container. By default docker pull image from the docker hub repository
2. If developers build a docker image with our applications and push that image to a docker repository. Then, we have to pull the dev image and run it within a container
3. If further works are needed, login to that container and start working

# Install Docker

Docker has a broken package issue with CentOS 8. If you plan to use docker then use CentOS 7 or ubuntu.

**@CentOS 7**

[root@ansible-client ~]# yum install -y yum-utils

[root@ansible-client ~]# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

[root@ansible-client ~]# yum install docker-ce docker-ce-cli containerd.io

**@Ubuntu 20.0.4**

mahidul@mahidul-pc:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common

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

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

mahidul@mahidul-pc:~$ sudo apt-get update

mahidul@mahidul-pc:~$ sudo apt install docker-ce

mahidul@mahidul-pc:~$ sudo systemctl status docker

Command related to Docker requires you to be a “**root**” user or a “**docker**” group member. If you are not working as a root user, you can use the “**usermod**” command to add the current user to the “**docker**” group:

mahidul@mahidul-pc:~$ sudo usermod -aG docker ${USER}

Here, the “**-aG**” option will append the current user “**${User}**” to the “**docker**” group.

mahidul@mahidul-pc:~$ sudo docker info

Server Version: 20.10.11

mahidul@mahidul:~$ sudo docker -v

Docker version 20.10.14, build a224086

# Configure and run a docker container

***Search a docker OS image i.e. ubuntu, centos distro***

mahidul@mahidul-pc:~$ sudo docker search ubuntu

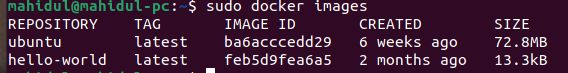
***Download ubuntu images for docker***

mahidul@mahidul-pc:~$ sudo docker pull ubuntu

mahidul@mahidul-pc:~$ sudo docker pull centos

***Check downloaded docker images***

mahidul@mahidul-pc:~$ sudo docker images



***Run Ubuntu image and create a new docker container***

mahidul@mahidul-pc:~$ sudo docker run -it ubuntu



The output is also displaying our container ID, which is “**b88bb44b456e**“. You should note your container ID because you will later need it to manage this container on your system.

***Work inside a docker container***

root@b88bb44b456e:/# apt update

root@b88bb44b456e:/# apt install nodejs

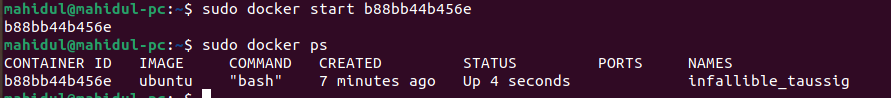
root@b88bb44b456e:/# node -v

v10.19.0

root@b88bb44b456e:/# exit

***Check the active container status***

mahidul@mahidul-pc:~$ sudo docker ps



***Check how much resources each docker is consuming***

mahidul@mahidul-pc:~$ sudo docker stats

***Start a docker container with docker id/name***

*mahidul@mahidul:~$ sudo docker run -it nginx*

mahidul@mahidul-pc:~$ sudo docker start infallible\_taussig

or

mahidul@mahidul-pc:~$ sudo docker start b88bb44b456e

***Rename a container***

mahidul@mahidul:~$ sudo docker rename bold\_swanson tc

***Check active containers***

mahidul@mahidul:~$ sudo docker *container* ps

or

mahidul@mahidul:~$ sudo docker container ls

***Check all (active+inactive) containers***

mahidul@mahidul:~$ sudo docker *container* ps -a

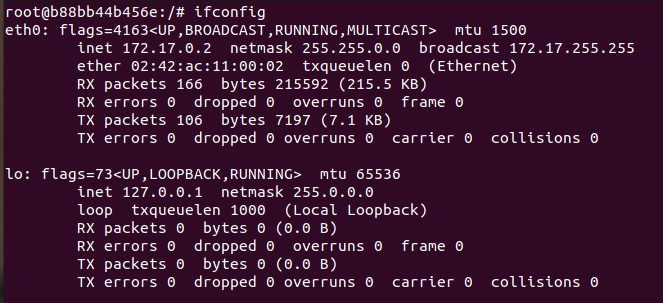
***Log in to a docker container***

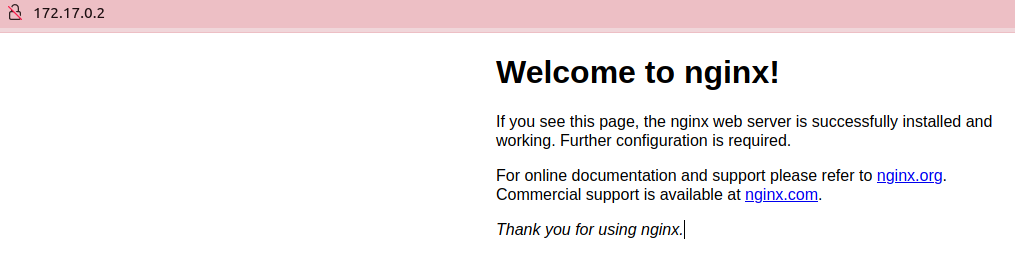
mahidul@mahidul-pc:~$ sudo docker *container* exec -dit infallible\_taussig /bin/bash

or

mahidul@mahidul-pc:~$ sudo docker attached infallible\_taussig

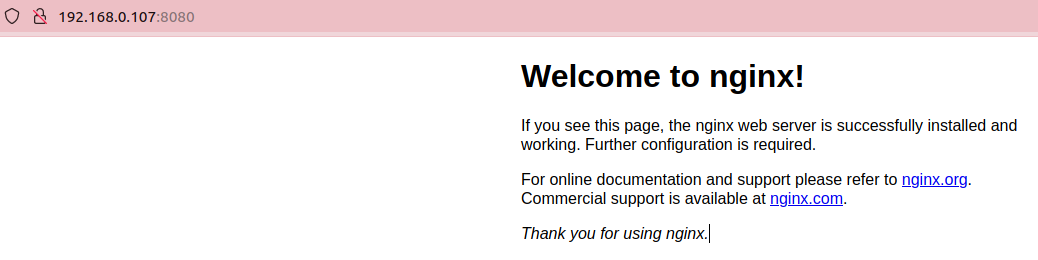
***Check the IP address***





**Create a new container with public IP mapping**

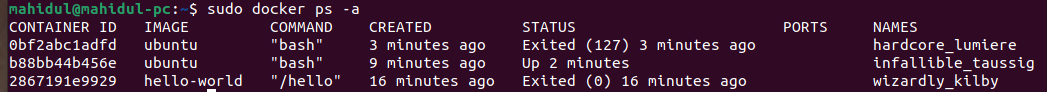
mahidul@mahidul-pc:~$ sudo docker run -t -d -p 8080:80 --name ubuntu\_host\_2 ubuntu



******

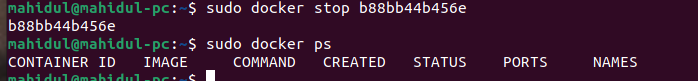
***Check all container status***

mahidul@mahidul-pc:~$ sudo docker ps -a



***Stop a docker container***

mahidul@mahidul-pc:~$ sudo docker stop b88bb44b456e



**How to remove a Docker container on Ubuntu**

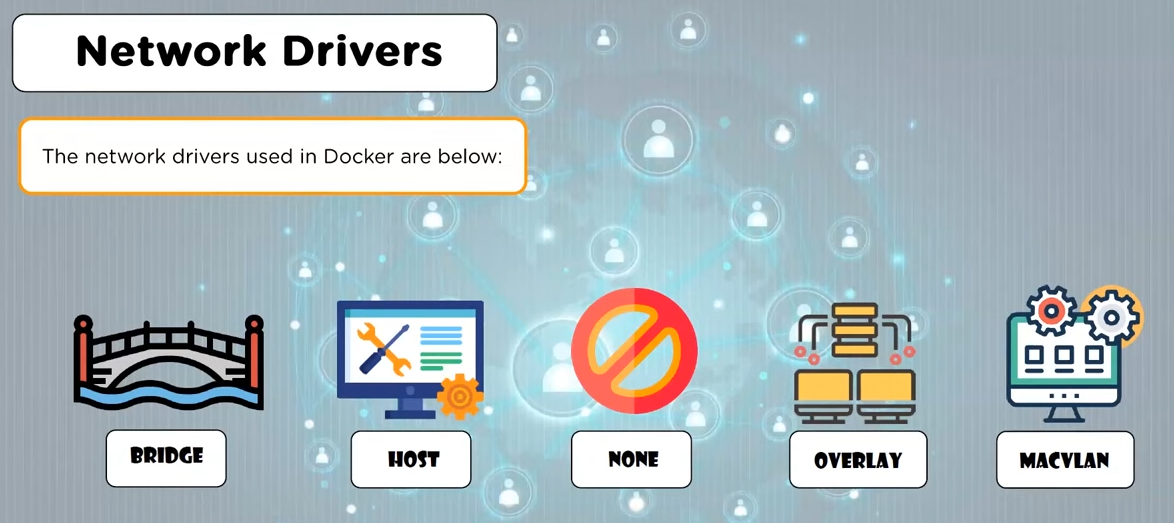
mahidul@mahidul-pc:~$ sudo docker stop brave\_brown

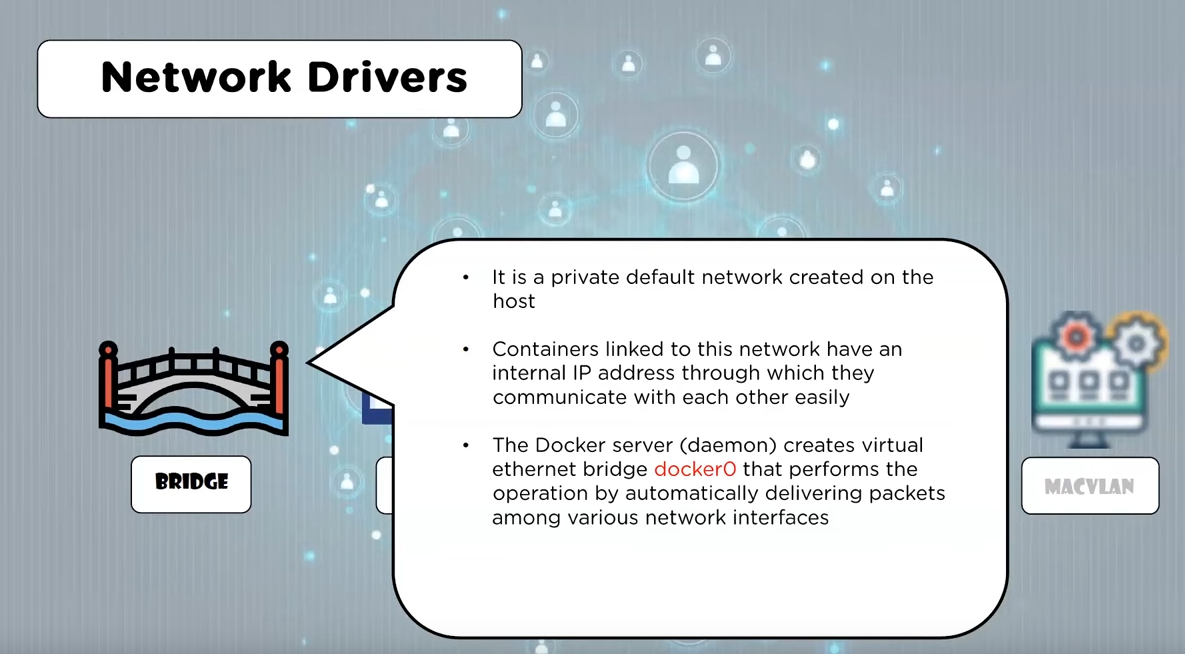
mahidul@mahidul-pc:~$ sudo docker rm brave\_brown

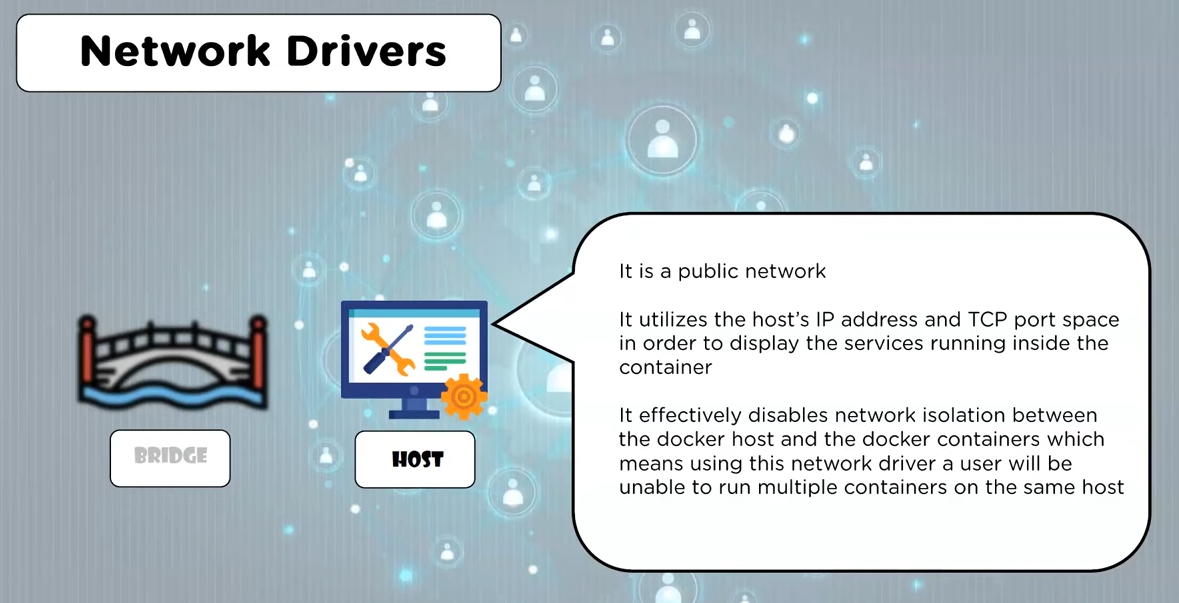
***Check docker IP address***

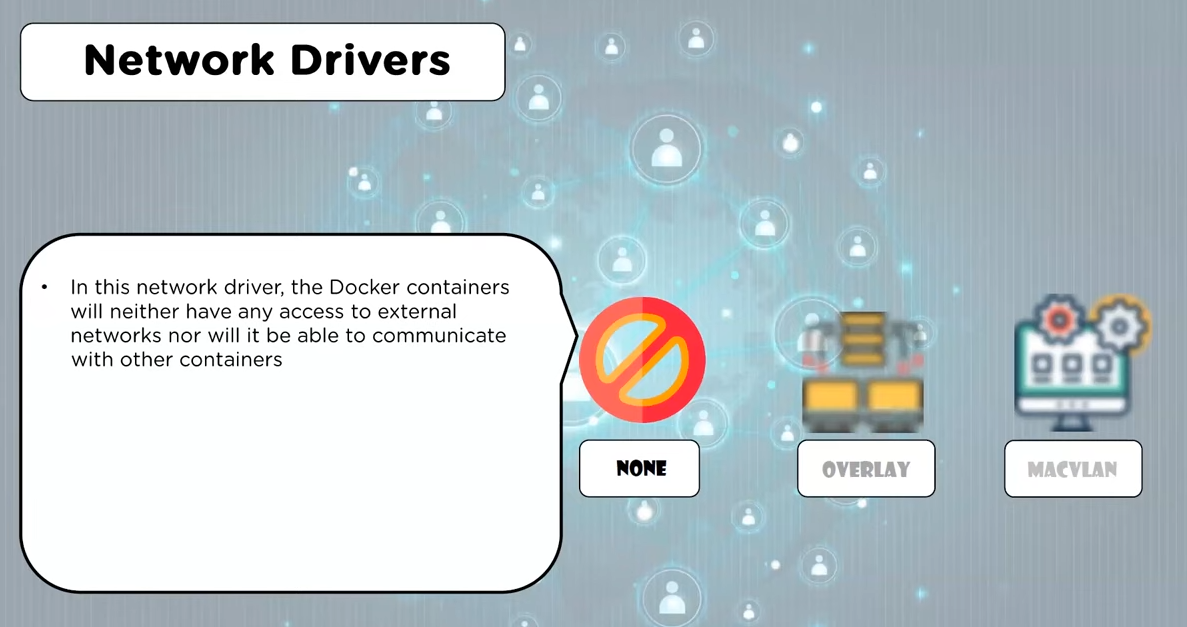
mahidul@mahidul:~$ sudo docker inspect lb | grep "IPAddress"

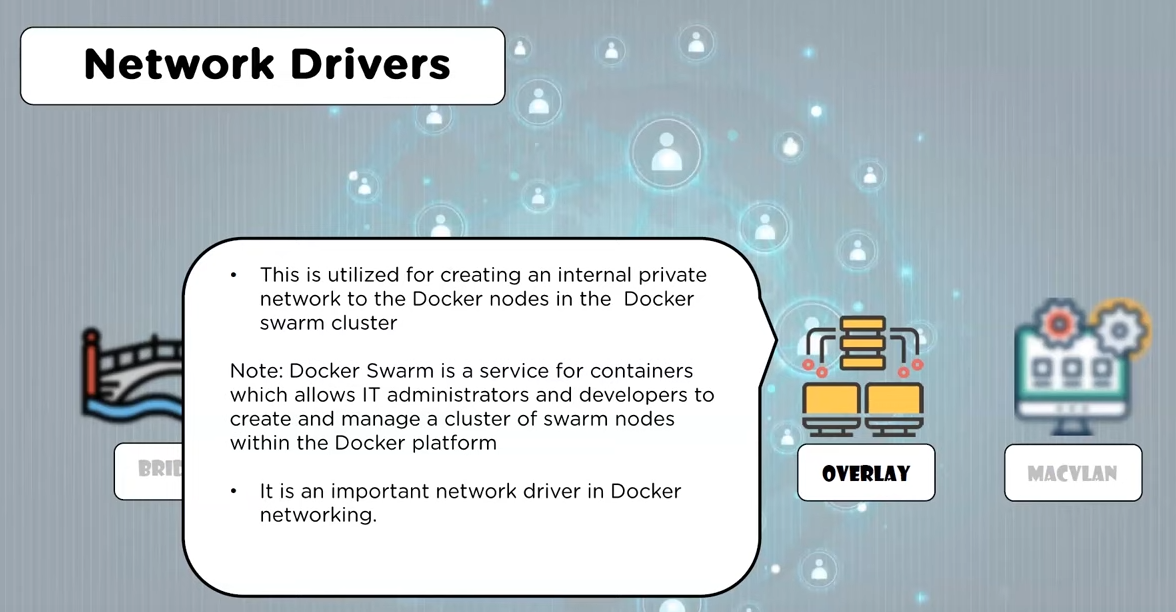
**Configure nginx:**

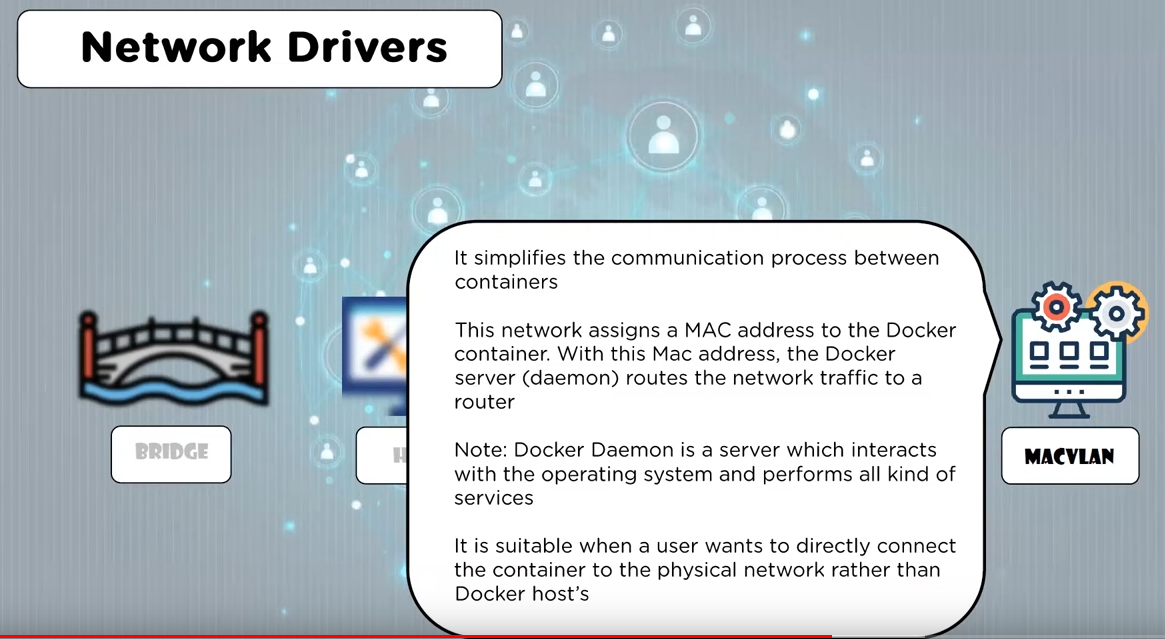
****

****

****

****

****

****

mahidul@mahidul:~$ sudo docker network ls

mahidul@mahidul:~$ sudo docker network inspect bridge

**Access the container’s services**

1. We have to use iptable(NAT) from hostOS for accessing containers
2. We can also host LB at hostOS for accessing containers

**Reference**

<https://www.youtube.com/watch?v=c6Ord0GAOp8>

<https://www.youtube.com/watch?v=Yr6-2ddhLVo>

**How to commit changes to a new Docker image and push to the Docker repository**

docker commit -m “description of changes” -a “Author\_Name” [Container\_ID] [Repository]/[Docker\_Image\_Name]

**Here:**

1. “-m” option is added to commit the message.
2. “-a” is utilized for specifying the name of the author.
3. “Author\_Name” will be your username.
4. [Container\_ID] is the ID of a specific Docker Container for which you want to commit the changes.
5. If you have created any repository on the Docker Hub, then add its name in the [Repository]; otherwise, you will write your Docker username in its place.
6. Lastly, specify the name of the new Docker image in the [Docker\_Image\_Name].

For instance, for the user “linuxhint” having the container ID “3bdee2efafad” the “docker command” will be:

sudo docker commit -m "added Node.js" -a "linuxhint" 3bdee2efafad linuxhint/ubuntu-nodejs

**How to upload our docker image to the docker hub repository**

1. Create an account at docker hub <https://hub.docker.com/signup>
2. Create a private repo
3. Log in to Docker from the command line

$ sudo docker login docker.io

1. Create a tag and push our image

$ sudo docker tag linuxhint/ubuntu-nodejs sharqa/ubuntu-nodejs

$ sudo docker push sharqa/ubuntu-node.js

**How to control hardware resource allocation in Docker**

# Reference

<https://www.youtube.com/watch?v=pTFZFxd4hOI>

<https://www.youtube.com/watch?v=eGz9DS-aIeY>

<https://linuxhint.com/install_configure_docker_ubuntu/>