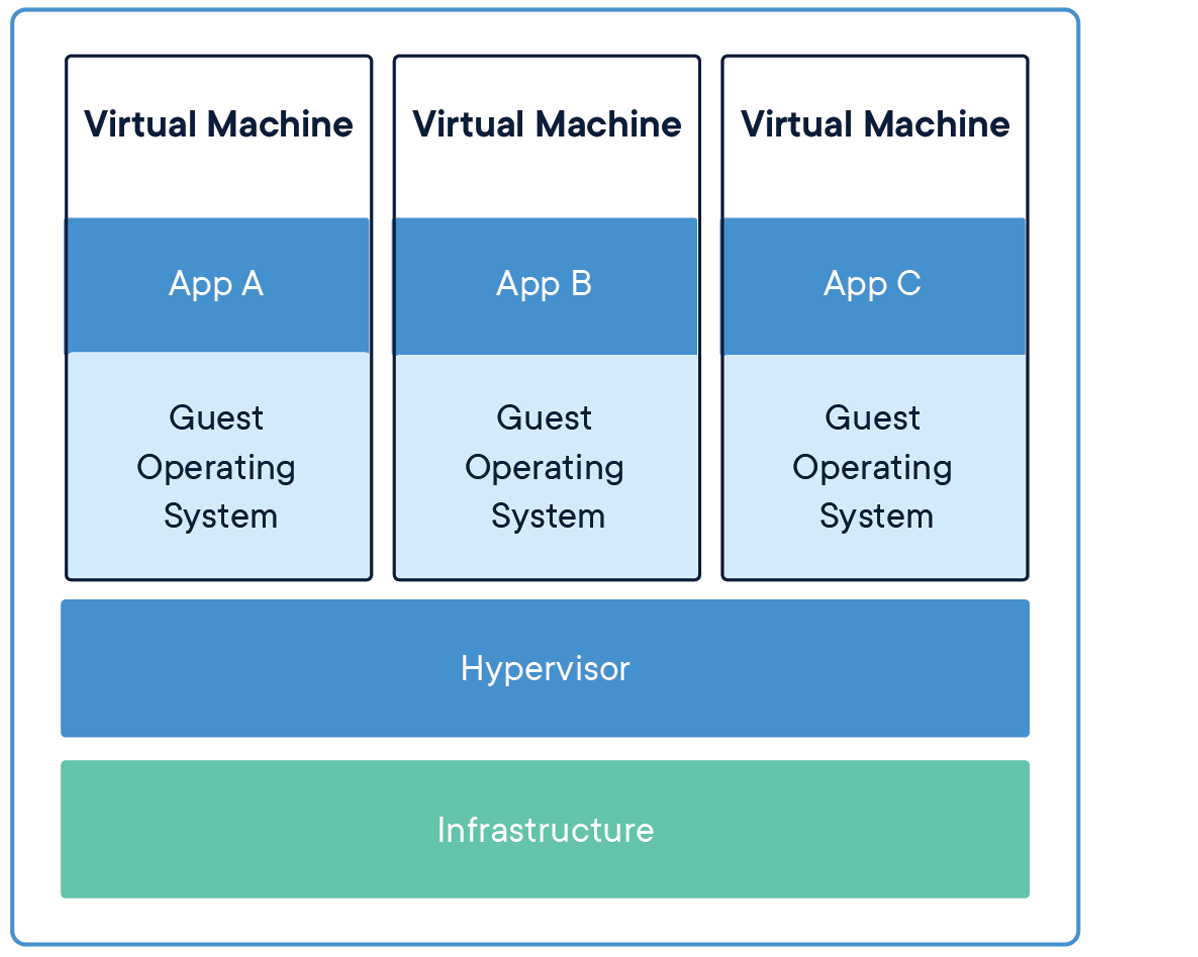


CONTAINERS

Containers are an abstraction at the app layer that packages code and dependencies together. Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space. Containers take up less space than VMs (container images are typically tens of MBs in size), can handle more applications and require fewer VMs and Operating systems.



Container

A **container** is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. ... Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure.

Container softwares

Docker

Rocket

Init

Cri – o

Docker

**Docker** is a set of [platform as a service](https://en.wikipedia.org/wiki/Platform_as_a_service) (PaaS) products that use [OS-level virtualization](https://en.wikipedia.org/wiki/OS-level_virtualization) to deliver software in packages called containers.[[6]](https://en.wikipedia.org/wiki/Docker_(software)#cite_note-SYS-CON_Media-7) Containers are isolated from one another and bundle their own software, [libraries](https://en.wikipedia.org/wiki/Library_(computing)) and configuration files; they can communicate with each other through well-defined channels.[[7]](https://en.wikipedia.org/wiki/Docker_(software)#cite_note-8) Because all of the containers share the services of a single [operating system kernel](https://en.wikipedia.org/wiki/Kernel_(operating_system)), they use fewer resources than [virtual machines](https://en.wikipedia.org/wiki/Virtual_machine).[[8]](https://en.wikipedia.org/wiki/Docker_(software)#cite_note-what-is-a-container-9)

The service has both free and premium tiers. The software that hosts the containers is called **Docker Engine**.[[8]](https://en.wikipedia.org/wiki/Docker_(software)#cite_note-what-is-a-container-9) It was first started in 2013 and is developed by [Docker, Inc.](https://en.wikipedia.org/wiki/Docker,_Inc.)[[9]](https://en.wikipedia.org/wiki/Docker_(software)#cite_note-os4u-10)

Image

It’s a template for desired state of the application

Container

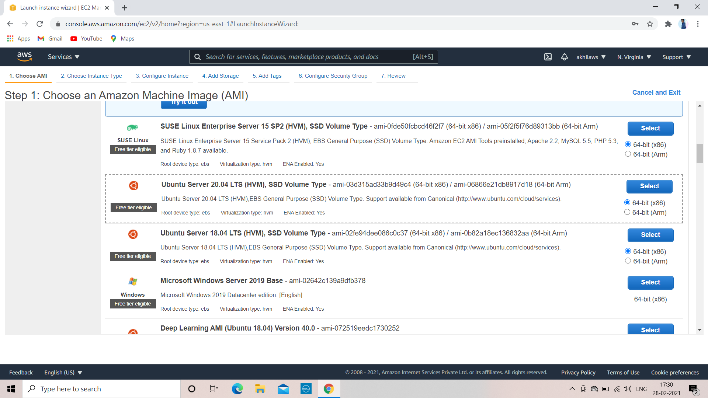
The running state of image

Registry

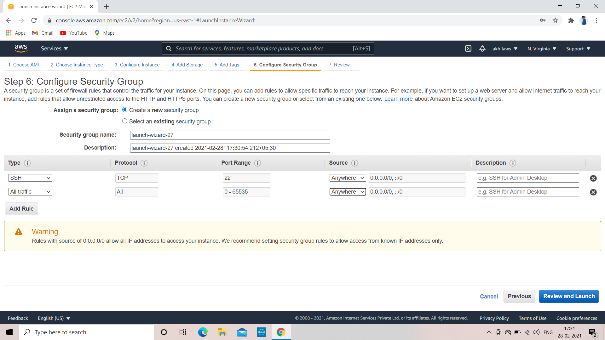
The place where we can keep the images for others availability

Step 1

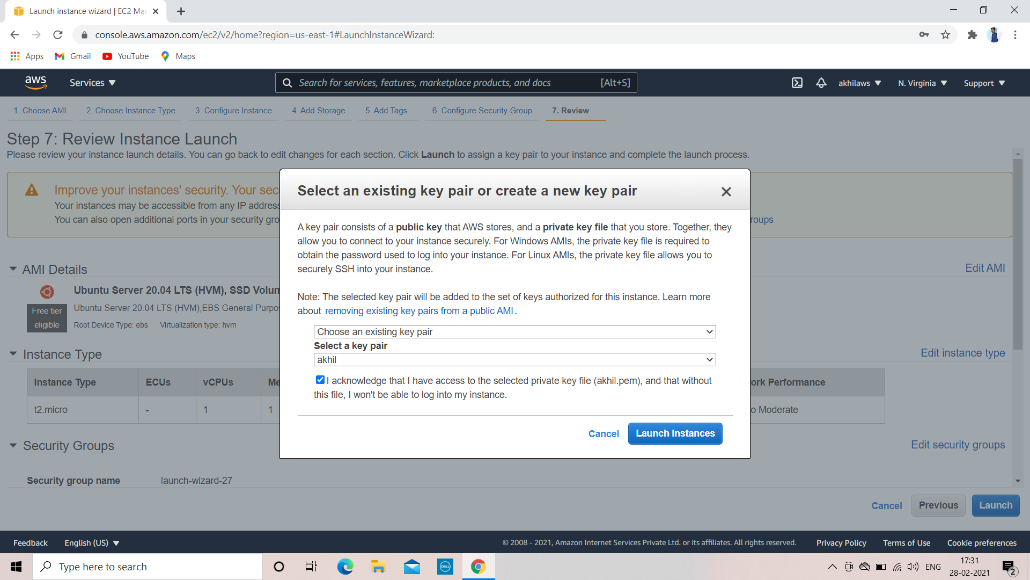
Open AWS and select ec2 services and choose one AMI



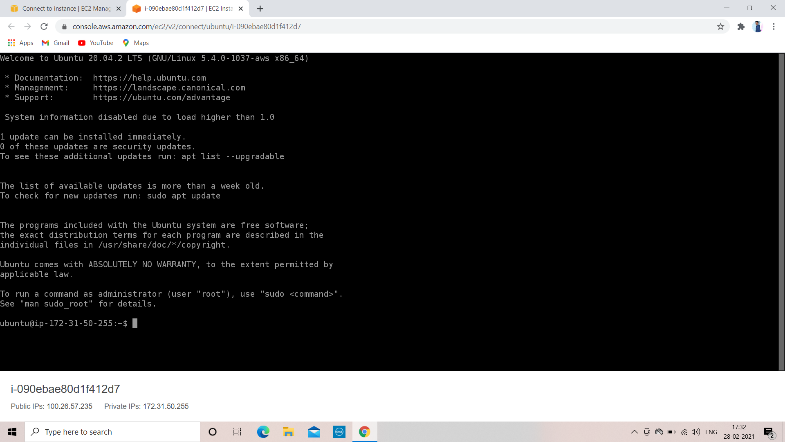
Allow all traffic



Finally review and launch the instance

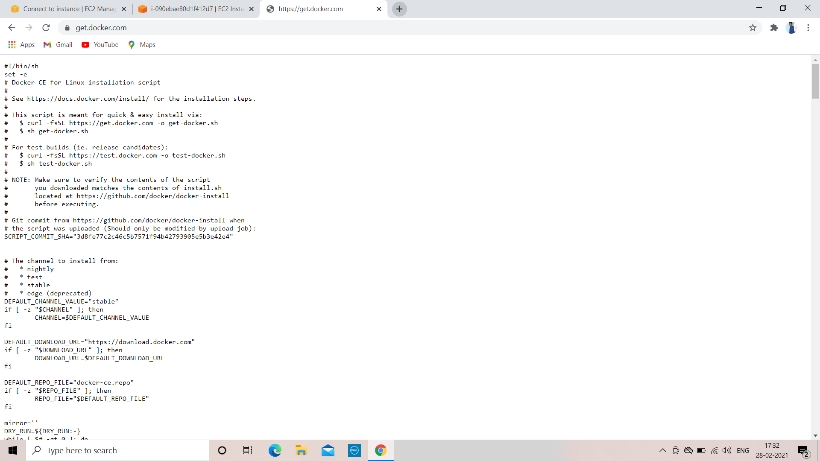


Connect the instance to the Linux



Copy the URL from the get.docker.com

Using URL download the docker



After downloading the docker run the command

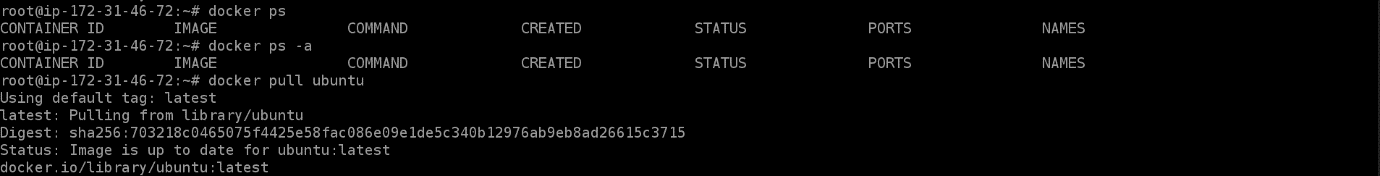
Docker image

A **Docker image** is a read-only template that contains a set of instructions for creating a **container** that can run on the **Docker** platform. It provides a convenient way to package up applications and preconfigured server environments, which you can use for your own private use or share publicly with other **Docker** users

docker images [to check the images]

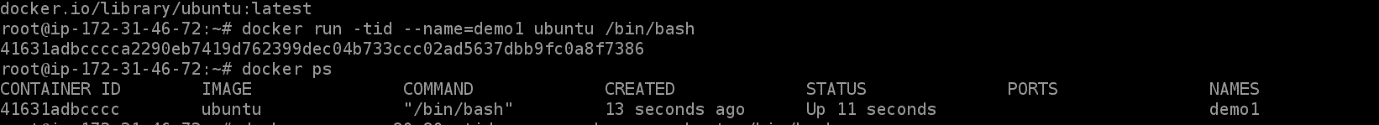
docker ps [to see the running container]

docker ps -a [to see the stopped container]



To run the container, use the below command

docker run -tid –name=container name image name or id /bin/bash

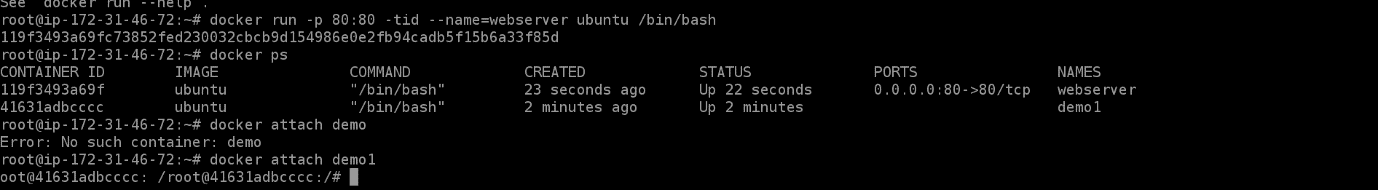


To run the webserver, use below command

docker run -p 80:80 -tid –name=webserver [image name or id of the image] /bin/bash

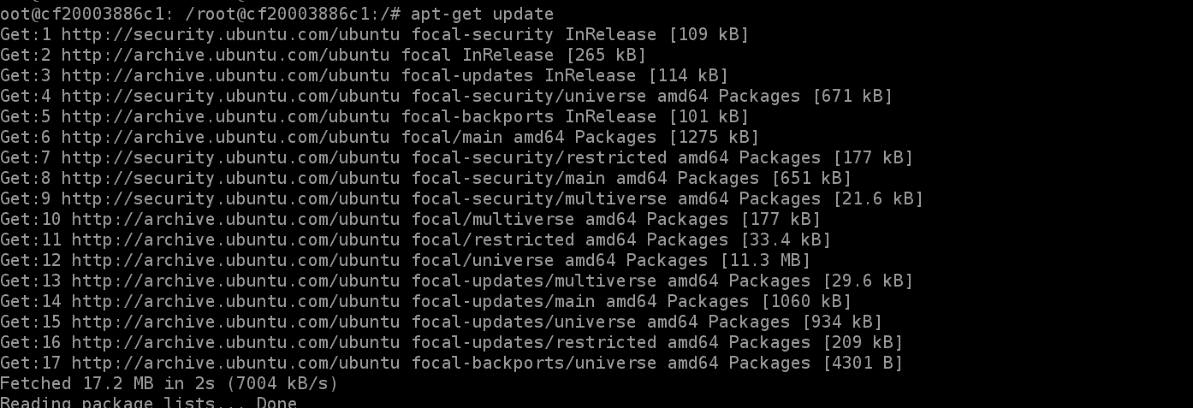
login to the container, use the command

docker attach [container name]



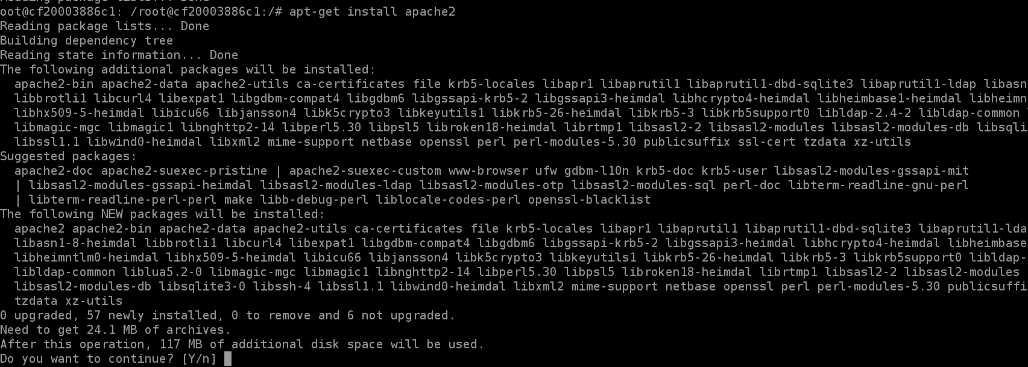
And update

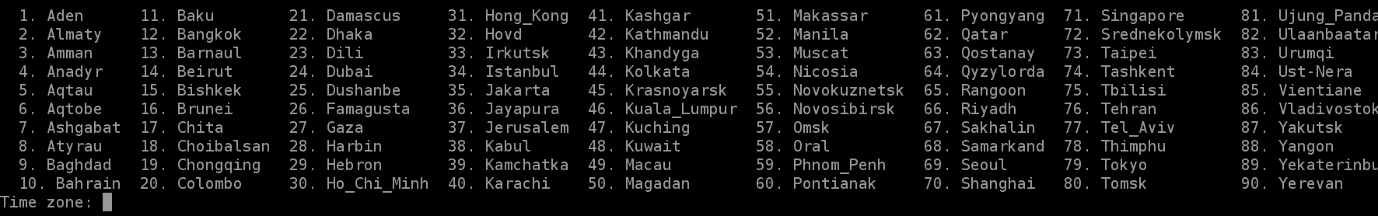
apt-get update



And install the webserver

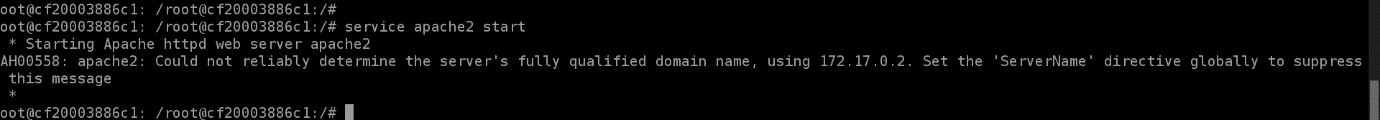
apt-get install apache2



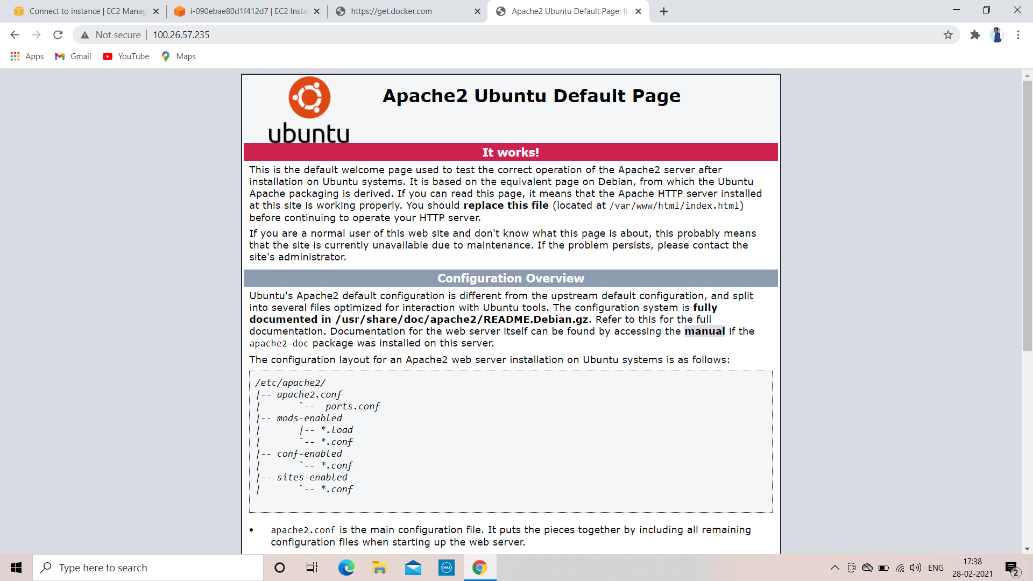


Start the service using below command

Service apache2 start



Copy the ip of the instance and search in the browser

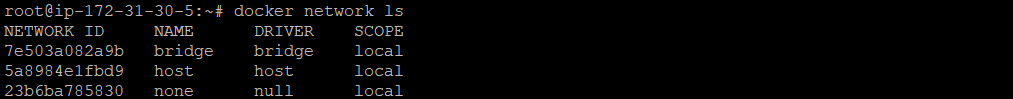


If config {to see ip address}

To see the network, use the commad

Docker network ls

Docker network ls



In docker we have 3 networks

1. Bridge network -- which allows containers connected to same bridge to communicate which providing isolation from containers which are not connected to the bridge network.
2. Host – to connect outside of the network
3. None – we don’t use any network

docker official site

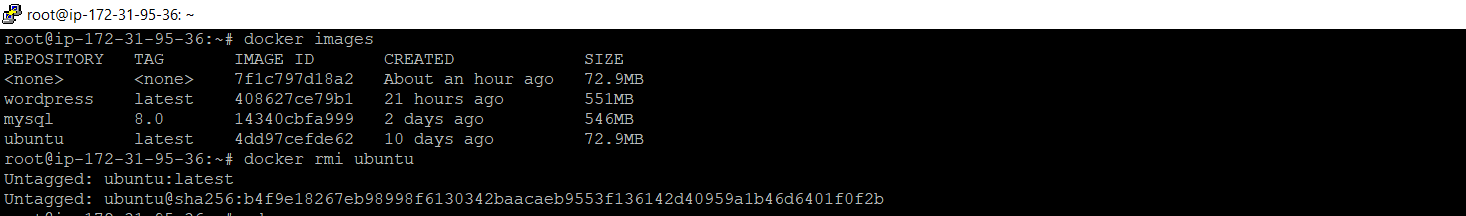
Docker.io

Docker home directory

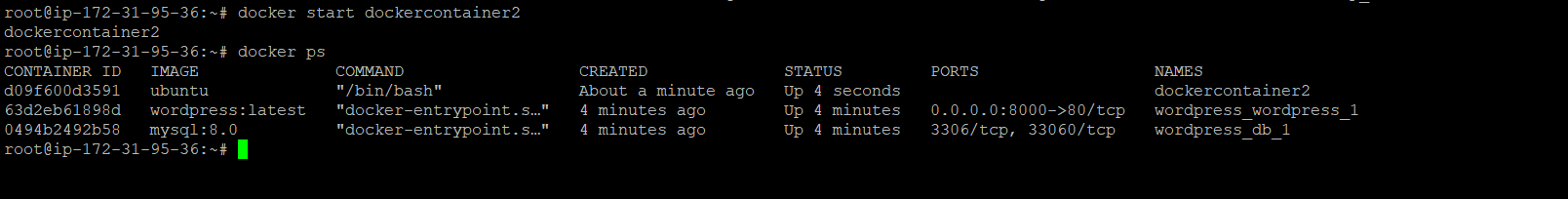
* /var/lib/docker
* Docker stop [container name] to stop the container



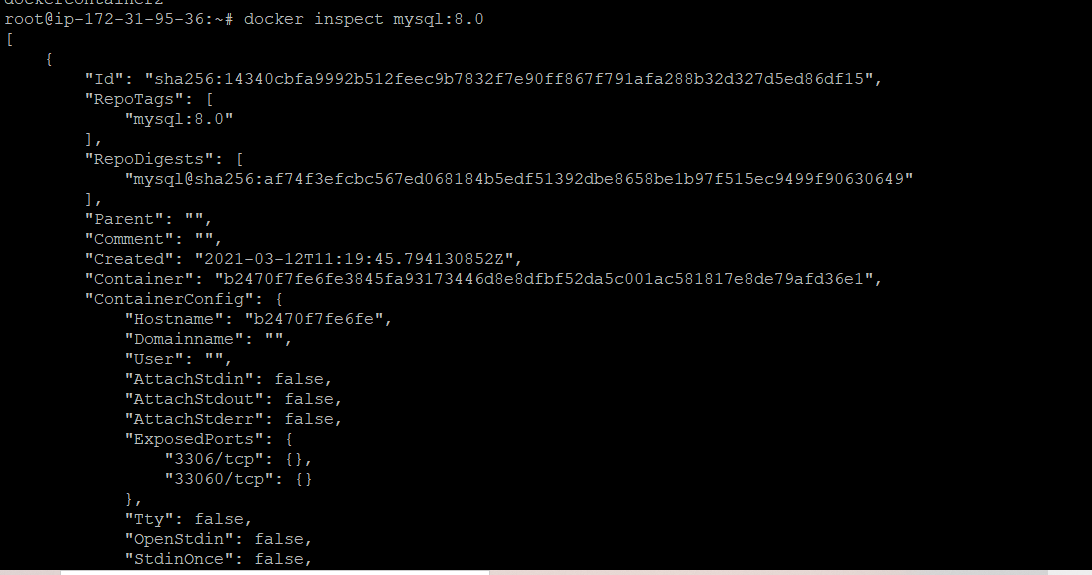
* Docker rmi [image name] to remove the container



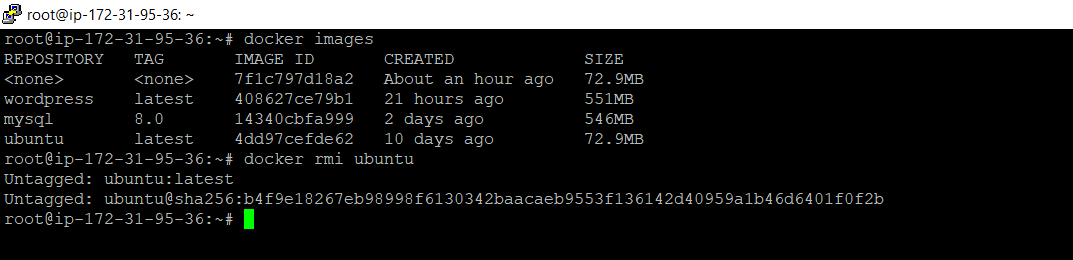
* Docker rm [container name] remove the container
* 
* Docker start [container name] to start the container

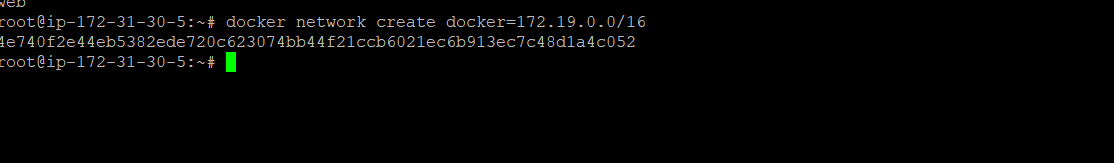


* Docker inspect [image name or image id] to see image full information



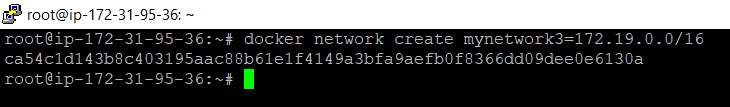
* Docker rmi [image name or id] to remove a image



Docker network create [name of the network] –subnet=172.19.0.0/16

Docker run -tid --network=mynet --name=web ubuntu /bin/bash

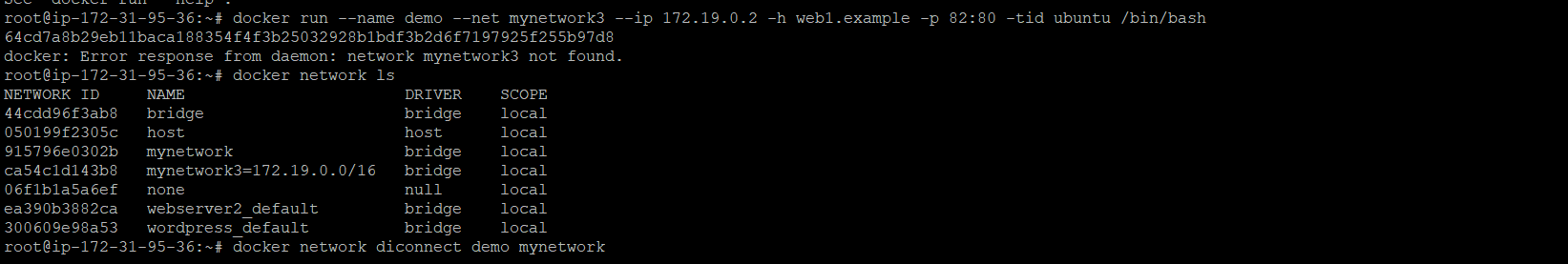
te



* Connect webserver to your network which you have created

Using below command

Docker run -tid –name [your webserver name] - -net [you network name] - -ip 172.19.0.2 -h web1.example.com -p 80:80 ubuntu /bin/bash



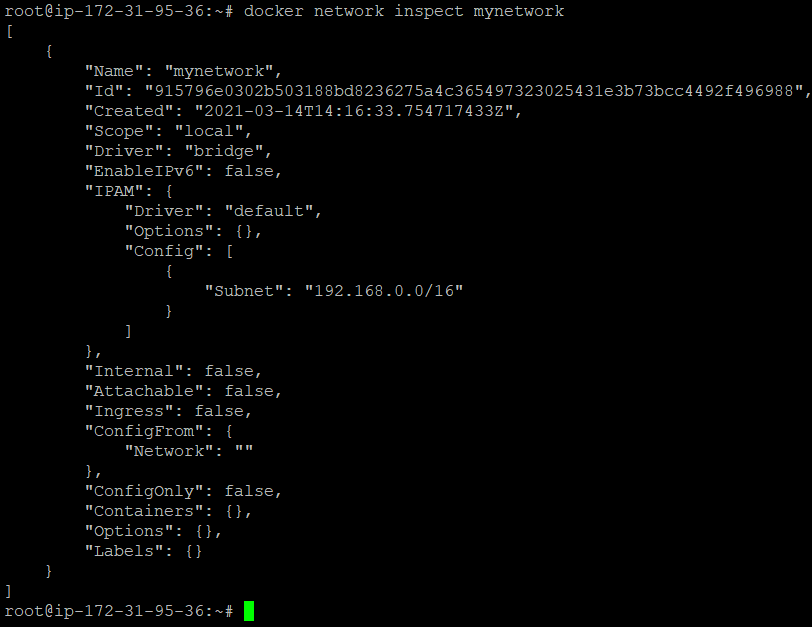
Create a index page

To write

* Cat>/var/www/html/index.html

To see information about your network use below command

* Docker network inspect [name of your network]

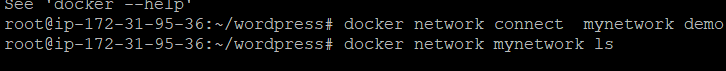


Remove network to webserver use the command

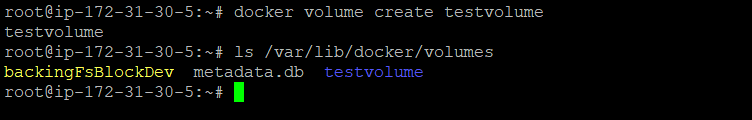
* Docker network diconnect [network name] [server name]



* Docker network connect [network name] [server name]

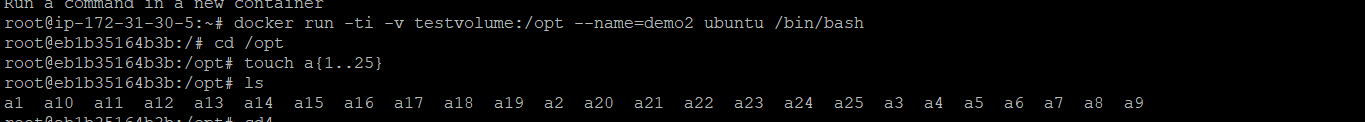


Create volume



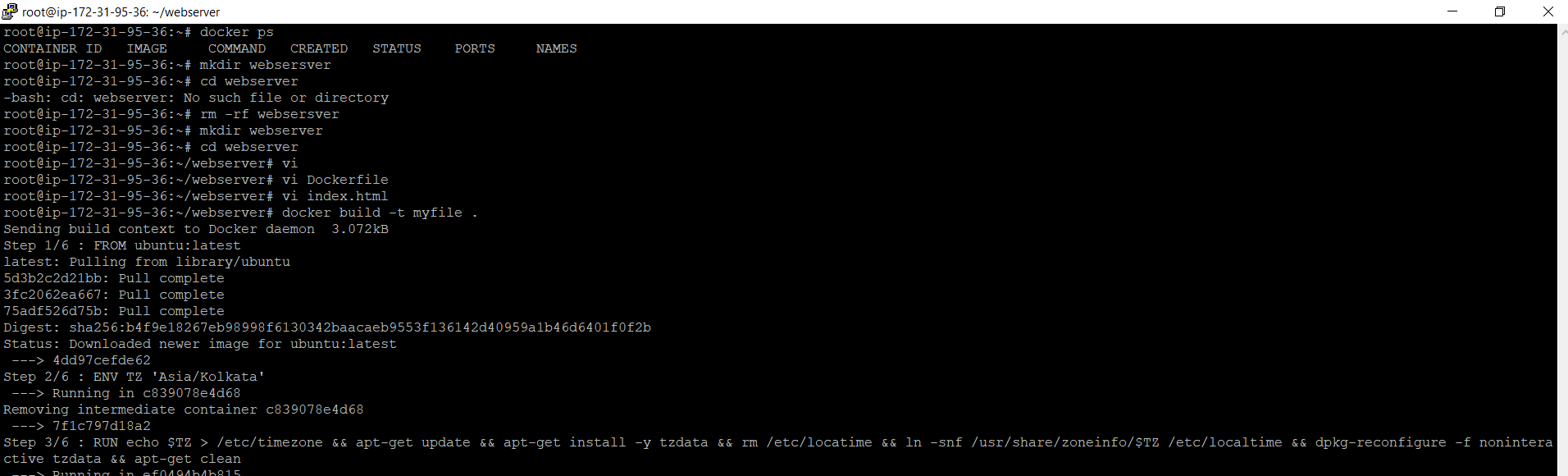
Docker run -ti -v testvolume:/opt –name =demo2 ubuntu /bin/bash

Move to opt



Dockerfile :- is a text document contain all the commands user could call on the command line to assemble image

Using build comand we can build image automatically

w

docker-compose.yaml

compose is tool defining and running multi-container docker application

with compose you use YAML file to configure your application services

define the services that make your application service in docker-compose.yaml they can run isolated environment

Main aim is to maintain multiple-container

wordpress: - blocks –to run our blocks/containers in internet

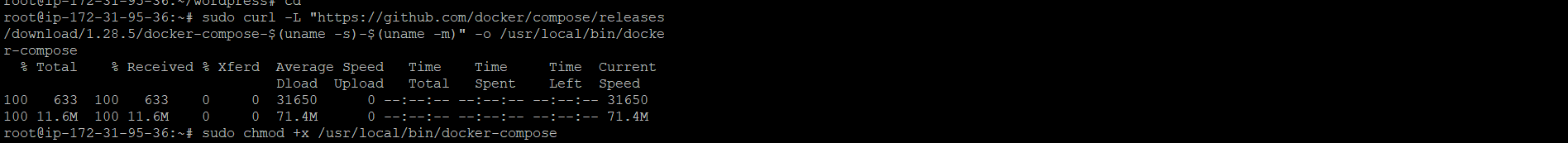
Used to create beautiful webisode and blogs

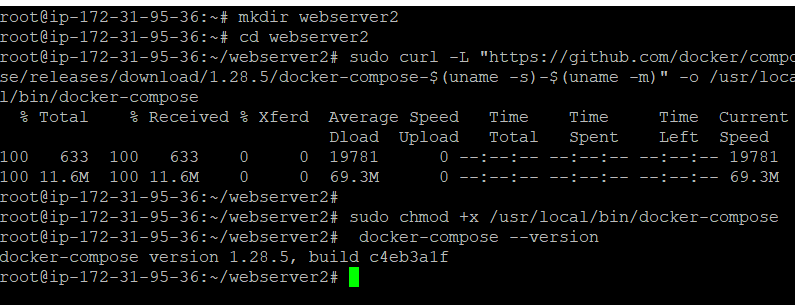
db: - database

step 1

* Download docker compose

sudo curl -L "https://github.com/docker/compose/releases/download/1.28.5/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

* sudo chmod +x /usr/local/bin/docker-compose
* docker-compose –version
* 



create a directory

mkdir [name]

move into the directory

First create docker-compose.yaml file

Edit the file using editor for example

volumes:

- db\_data:/var/lib/mysql

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: somewordpress

MYSQL\_DATABASE: wordpress

MYSQL\_USER: wordpress

MYSQL\_PASSWORD: wordpress

wordpress:

depends\_on:

- db

image: Wordpress:latest

ports:

- "8000:80"

restart: always

environment:

WORDPRESS\_DB\_HOST: db:3306

WORDPRESS\_DB\_USER: wordpress

WORDPRESS\_DB\_PASSWORD: wordpress

WORDPRESS\_DB\_NAME: wordpress

volumes:

db\_data: {}

build the wordpress application using the command

