12/14jul

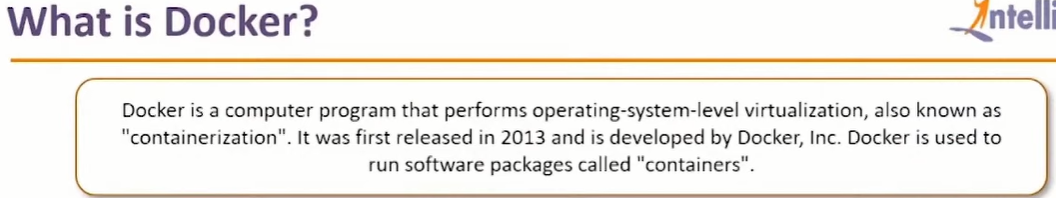
What Is docker ?

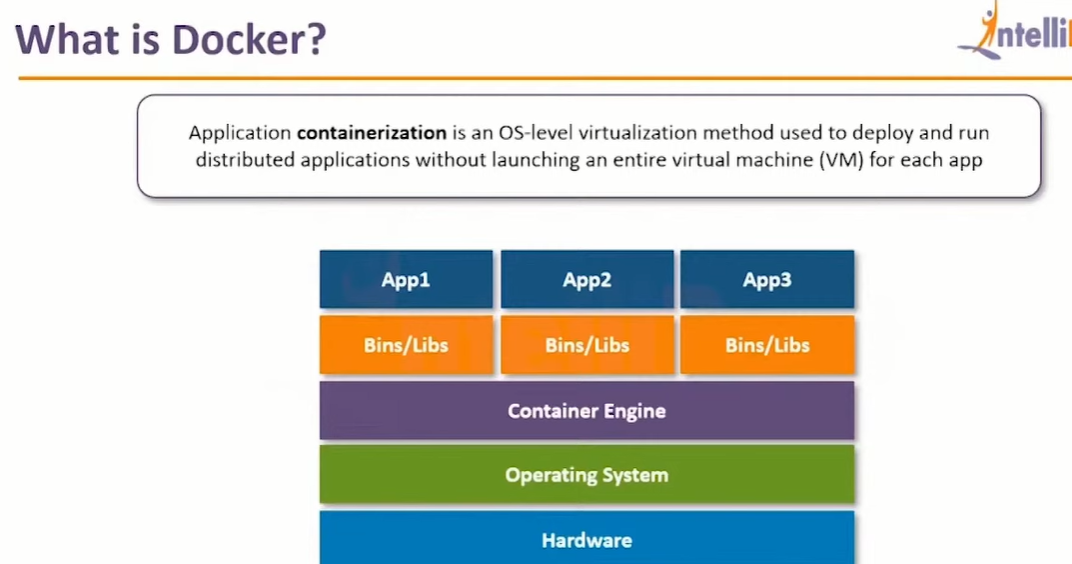
Docker is a containerization tool.

Virtualization -- Fixed hardware allocation.

Containerization - No Fixed Hardware

Process isolation ( Dependency in os is removed )





+++++++++++++++++++++++++

In comparison to the traditional virtualization functionalities of hypervisors,

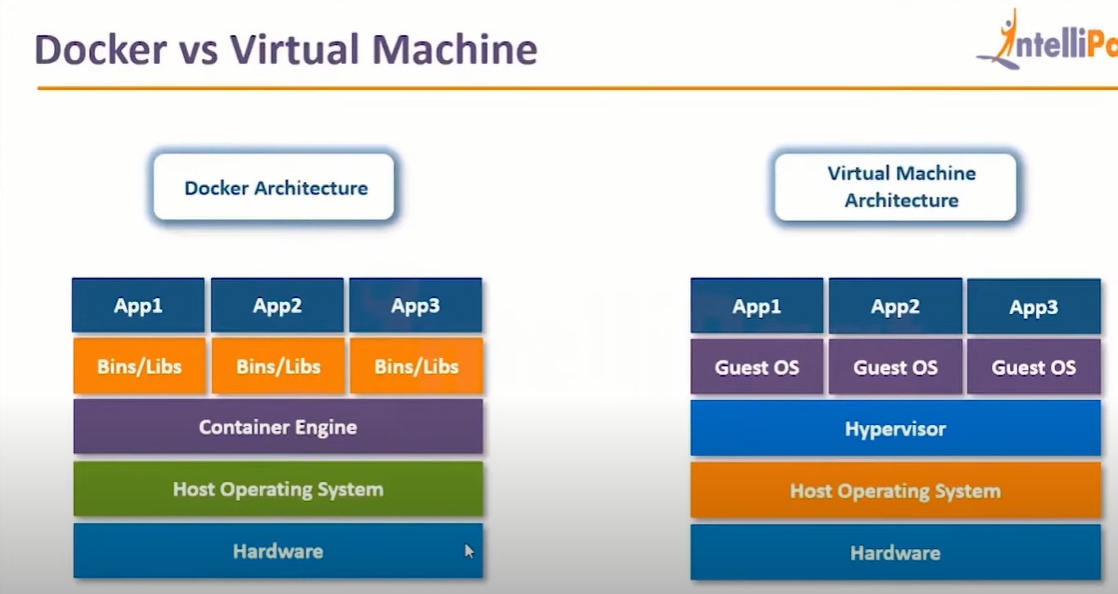
Docker containers eliminate the need for a separate guest operating system for every new virtual machine.

Docker implements a high-level API to provide lightweight containers that run processes in isolation.

What is container ?

A Docker container enables rapid deployment with minimum run-time requirements. It also ensures better management and simplified portability.

This helps developers and operations team in rapid deployment of an application.



How to install Docker on linux :

Create Ubuntu Machine on AWS

All Traffic - anywhere

Connect using git bash

https://get.docker.com/

Go to Root Account

$ sudo su -

# curl -fsSL https://get.docker.com -o get-docker.sh ( this will download shell script in the machine)

# sh get-docker.sh ( This will execute the shell script, which will install docker )



How to check the docker is installed or not

# docker --version

We should be comformatable with four terms

1) Docker Images

Combinations of binaries / libraries which are necessary for one software application.

2) Docker Containers

When image is executed comes into running condition, it is called container.

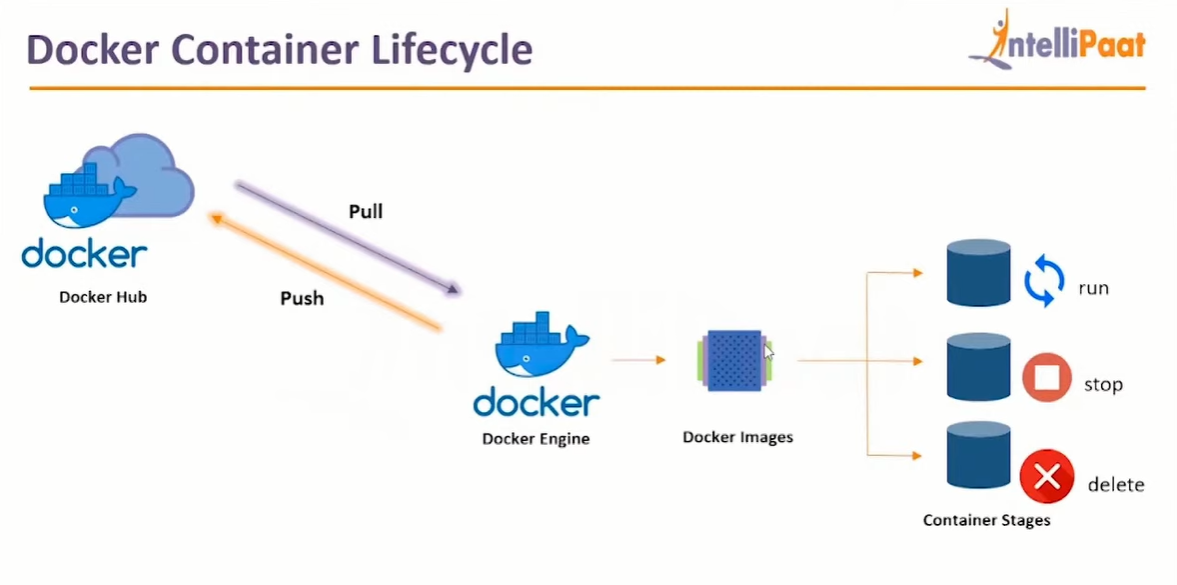
3) Docker Host

Machine on which docker is installed, is called as Docker host.

4) Docker Client

Terminal used to run docker run commands ( Git bash )

On linux machine, git bash will work like docker client.



Docker Commands

--------------------

Working on Images

-------------------------

1 To download a docker image

docker pull image\_name

2 To see the list of docker images

docker image ls

(or)

docker images

3 To delete a docker image from docker host

docker rmi image\_name/image\_id

4) To upload a docker image into docker hub

docker push image\_name

5) To tag an image

docker tag image\_name ipaddress\_of\_local\_registry:5000/image\_name

6) To build an image from a customised container

docker commit container\_name/container\_id new\_image\_name

7) To create an image from docker file

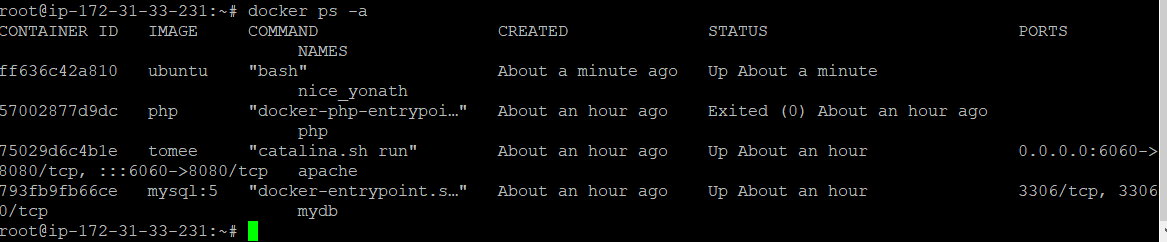
docker build -t new\_image\_name

8) To search for a docker image

docker search image\_name

9) To delete all images that are not attached to containers

docker system prune -a



++++++++++++++++++++++++++++++++++++++++++++++

To the container

Working on containers

-----------------------------

10) To see the list of all running continers

docker container ls

11) To see the list of running and stopped containers

docker ps -a

12) To start a container

docker start container\_name/container\_id

13) To stop a running container

docker stop container\_name/container\_id

14) To restart a running container

docker restart container\_name/container\_id

To restart after 10 seconds

docker restart -t 10 container\_name/container\_id

15) To delete a stopped container

docker rm container\_name/container\_id

16) To delete a running container

docker rm -f container\_name/container id

17) To stop all running containers

docker stop $(docker ps -aq)

18) To restart all containers

docker restart $(docker ps -aq)

19) To remove all stopped containers

docker rm $(docker ps -aq)

20) To remove all contianers(running and stopped)

docker rm -f $(docker ps -aq)

21) To see the logs generated by a container

docker logs container\_name/container\_id

22) To see the ports used by a container

docker port container\_name/container\_id

23) To get detailed info about a container

docker inspect container\_name/container\_id

24) To go into the shell of a running contianer which is moved into background

docker attach container\_name/container id

25) To execute anycommand in a container

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

Eg: To launch the bash shell in a contianer

docker exec -it container\_name/container\_id bash

26) To create a container from a docker image ( imp )

docker run image\_name

++++++++++++++++++++++++++++++++++++++++++++++++

-p Used for port mapping between port of container with the dockerhost port.

-P Used for automatic port mapping ie, it will map the internal port of the container

with some port on host machine.

This host port will be some number greater than 30000

-v Used for attaching a volume to the container

--volume-from Used for sharing volume between containers

--network Used to run the contianer on a specific network

--link Used for linking the container for creating a multi container architecture

--memory Used to specify the maximum amount of ram that the container can use

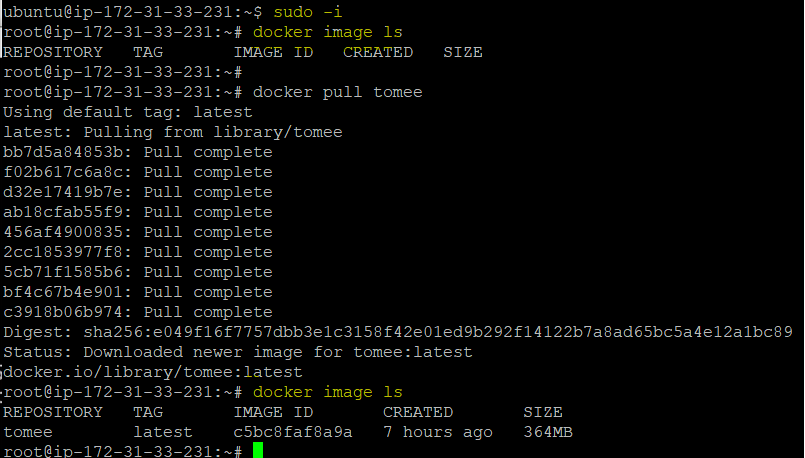
=======================================================================

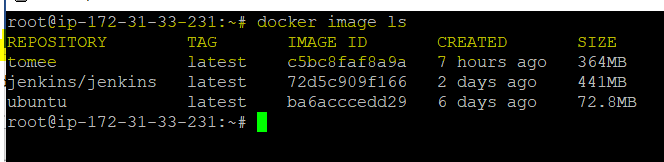
Docker hub is docker registory

To Download the images we used below command :

Docker pull tomee/Ubuntu/Jenkins/Jenkins

For listings the images : Docker image ls

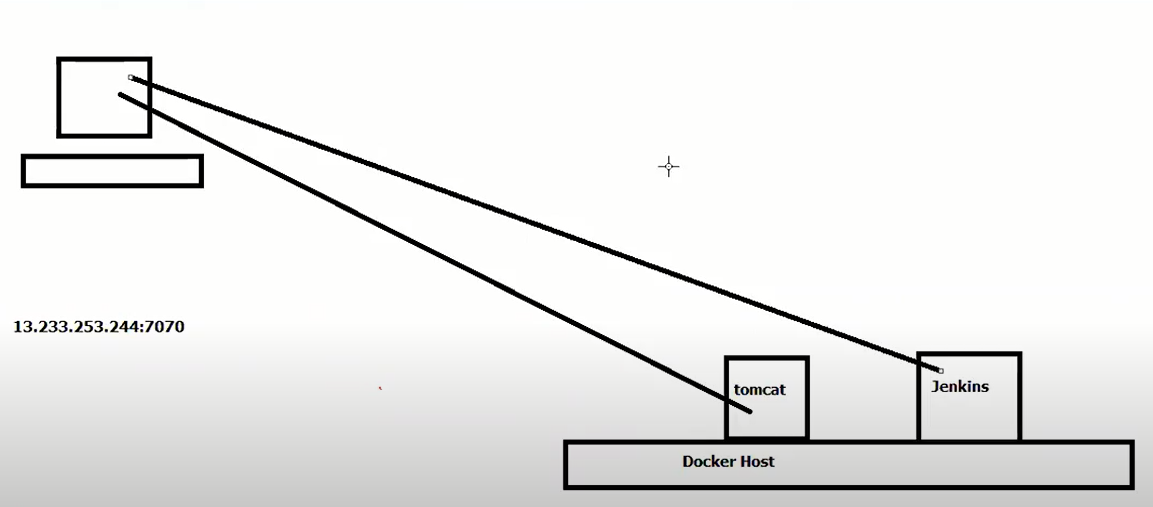


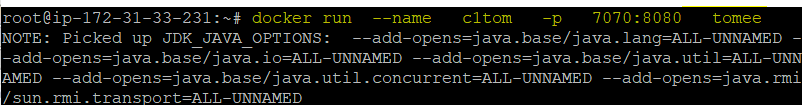


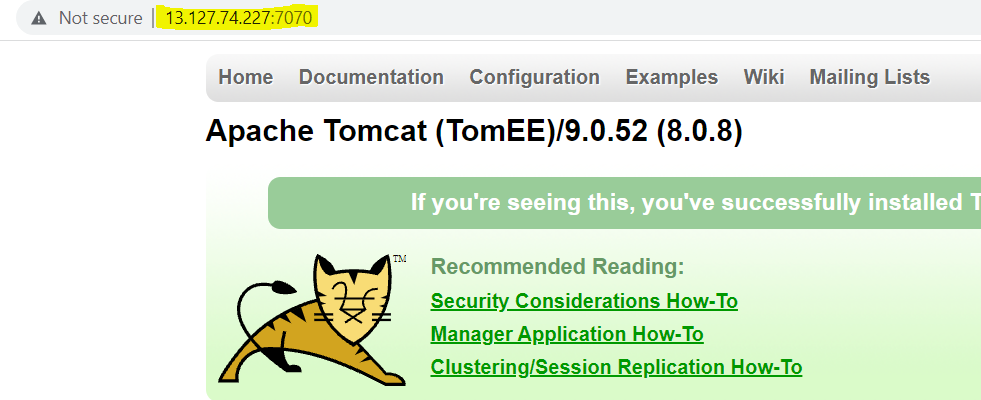
TO create a container from an image

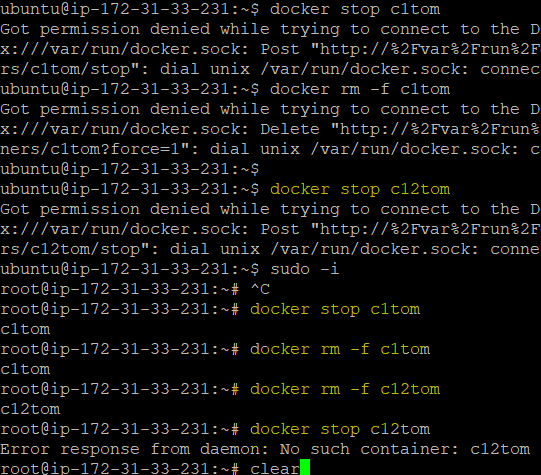
# docker run --name mytomcat -p 7070:8080 tomee

PORT MAPPING:

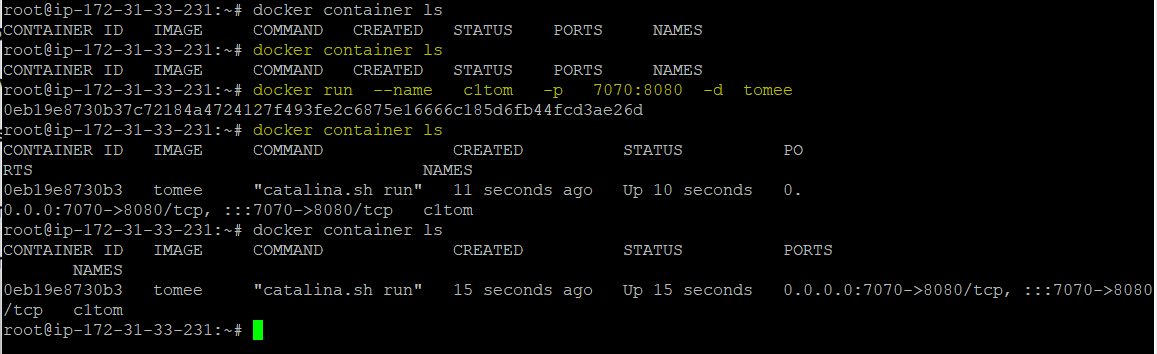








-d is for detach

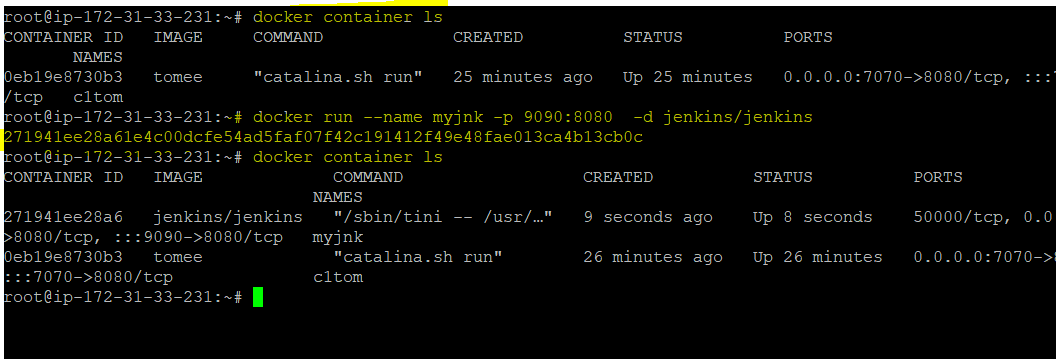


To find the public ip :

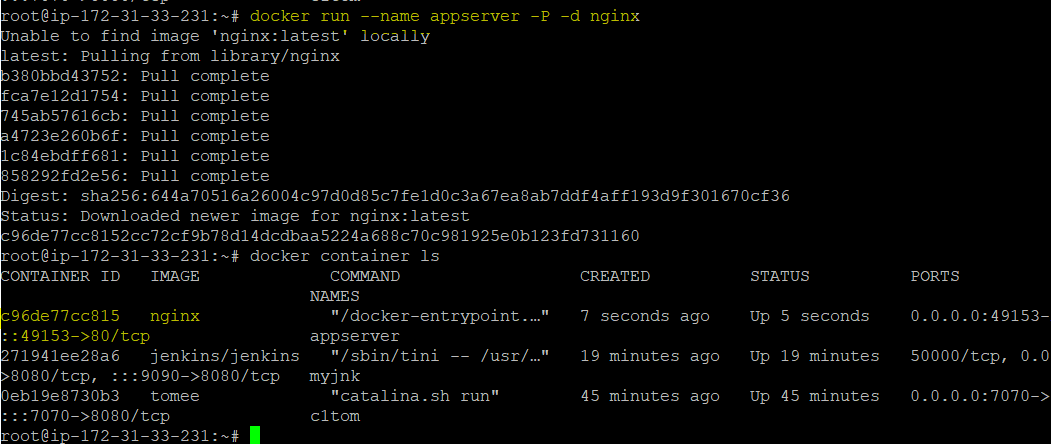
$ curl -s ifconfig.me 🡪162.202.17.123

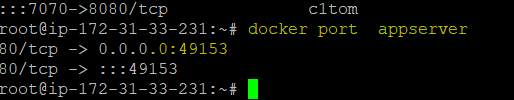
-p 🡪to assign any port

-P 🡪to assign random port



Without assign any port.





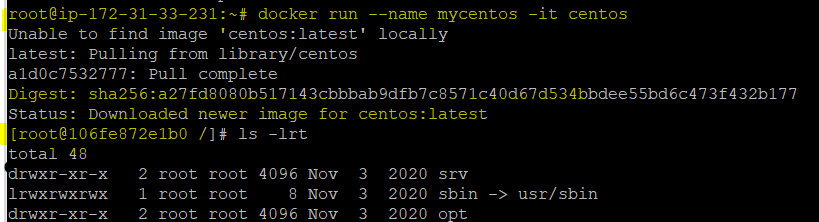
Interactive terminal :

To start centos as container

# docker run --name mycentos -it centos

|\_conatiner name

# exit ( To come back to dockerhost )



16-jul-2021

What is Env Variable where we used it ?

To start mysql as container, open interactive terminal in it, create a sample table.

# docker run --name mydb -d -e MYSQL\_ROOT\_PASSWORD=sunil mysql:5 (version)

-e 🡪env variable which we used doc from docker.hub

-d 🡪detach (without port)

Mysql:5 🡪version of SQl

# docker container ls

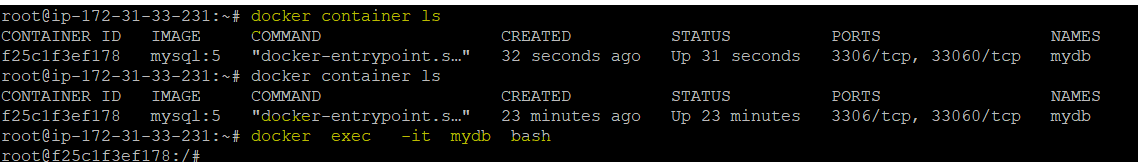
I want to open bash terminal of mysql 🡪 connecting to container

# docker exec -it mydb bash -🡪opening bash terminal for mysql container ;it🡪 interactive

To connect to mysql database

# mysql -u root -p 🡪-u 🡪user –p🡪password

enter the password, we get mysql prompt



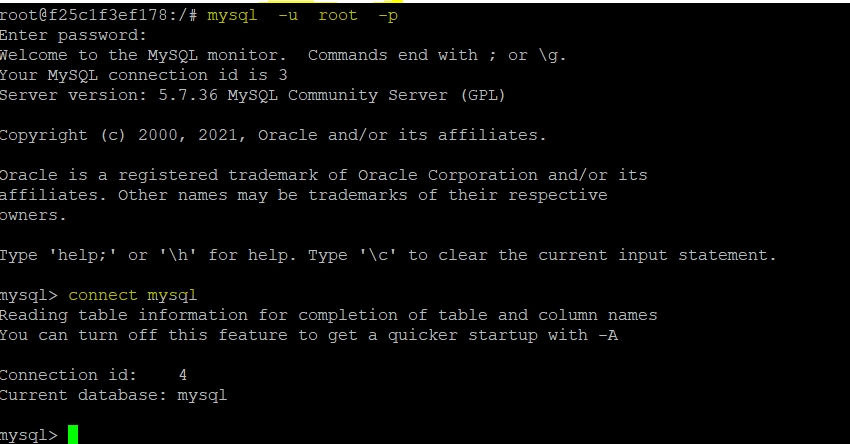
TO see list of databases

> show databases;

TO switch to a databse

> use db\_name

> use mysql



TO create emp tables and dept tables

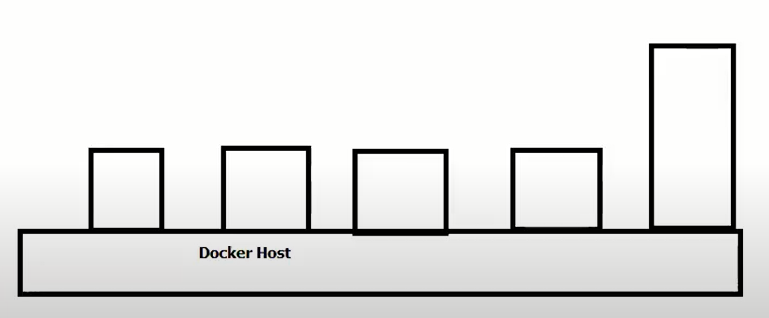
https://justinsomnia.org/2009/04/the-emp-and-dept-tables-for-mysql/

> exit

Exit 🡪from docker container

++++++++++++++++++

Multi container architecture using docker



One container can communicate with other container called multicontainer.

This can be done in 2 ways

1) --link

2) docker-compose

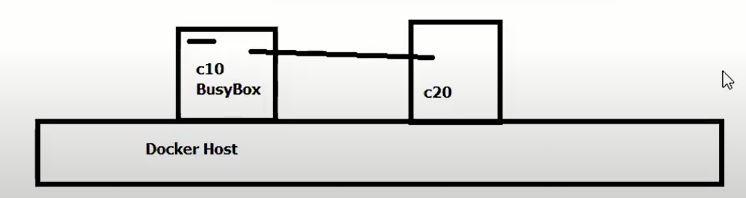
1) --link option

----------------------

Use case:

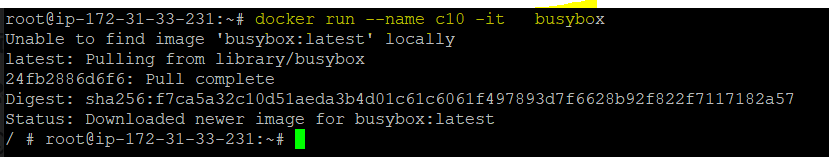
--------------

Start two busybox containers and create link between them



Create 1st busy box container

# docker run --name c10 -it busybox

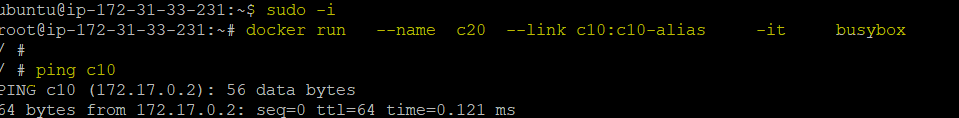


How to come out of the container without exit

( ctrl + p + q)

Create 2nd busy box container and establish link to c1 container

# docker run --name c20 --link c10:c10-alias -it busybox ( c10-alias is alias name)



How to check link is established for not?

/ # ping c1

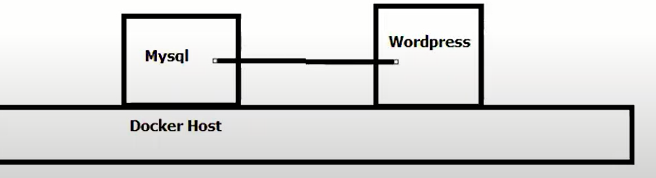
Ctrl +c ( to come out from ping )

( ctrl + p + q)+++++++++++++++++++++++++++++++++

Ex 2: Creating development environment using docker

Start mysql as container and link it with wordpress container.

Developer should be able to create wordpress website



1) TO start mysql as container

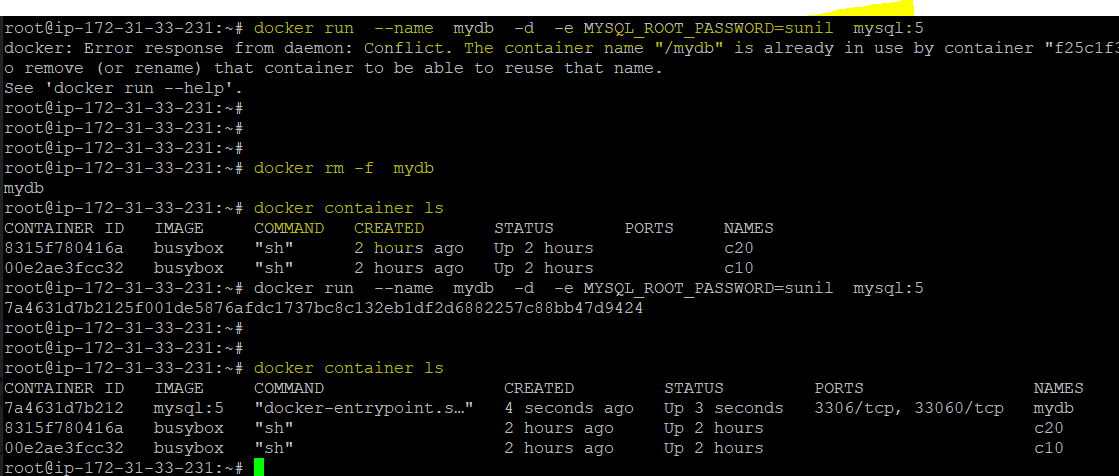
# docker run --name mydb -d -e MYSQL\_ROOT\_PASSWORD=sunil mysql:5

( if container is already in use , remove it

# docker rm -f mydb )

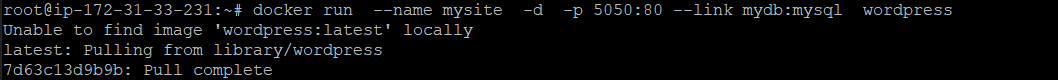
Check whether the container is running or not

# docker container ls



2) TO start wordpress container

# docker run --name mysite -d -p 5050:80 --link mydb:mysql wordpress



13.232.183.233:5050

Ex 3: Create LAMP Architecture using docker

L -- linux

A -- apache tomcat

M -- mysql

P -- php

( Linux os we already have )

Lets remove all the docker containers

# docker rm -f $(docker ps -aq)

# docker container ls ( we have no containers now )



1) TO start mysql as container

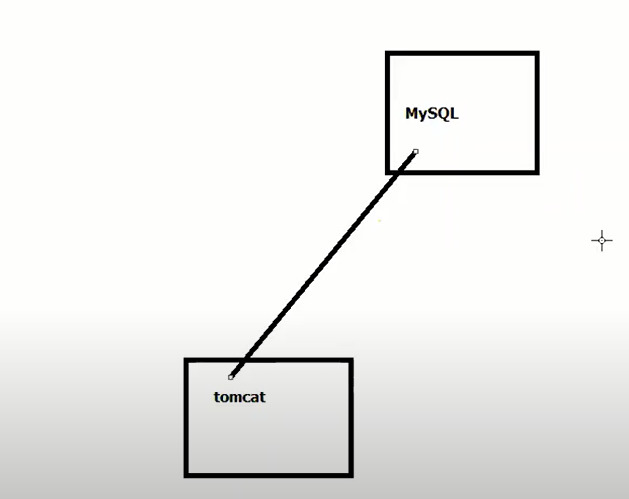
# docker run --name mydb -d -e MYSQL\_ROOT\_PASSWORD=sunil mysql:5

2) TO start tomcat as container

# docker run --name apache -d -p 6060:8080 --link mydb:mysql tomee

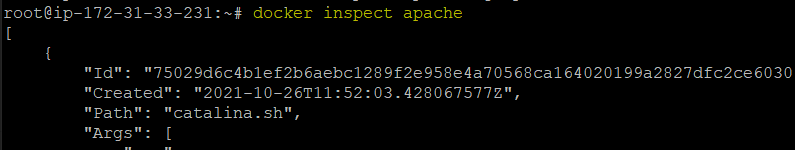
TO see the list of containers

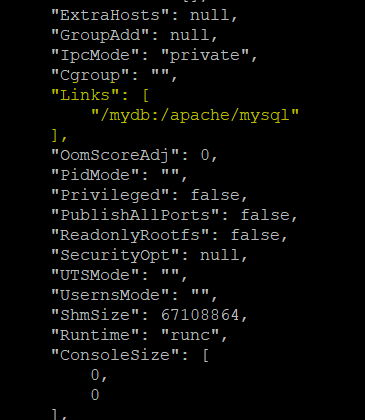
# docker container ls



To check if tomcat is linked with mysql

# docker inspect apache ( apache is the name of the container )





3) TO start php as container

# docker run --name php -d --link apache:tomcat --link mydb:mysql php

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

19/07 :

Ex 4:

Create CI-CD environment, where jenkins container is linked with two tomcat containers.

Lets delete all the container

# docker rm -f $(docker ps -aq)

To start jenkins as a container

# docker run --name devserver -d -p 7070:8080 jenkins/jenkins

to check jenkins is running or not?

Open browser

public\_ip:7070

http://18.138.58.3:7070

We need two tomcat containers ( qa server and prod server )

# docker run --name qaserver -d -p 8080:8080 --link devserver:jenkins tomee

to check the tomcat use public\_ip but port number will be 8080

http://18.138.58.3:8080

# docker run --name prodserver -d -p 9090:8080 --link devserver:jenkins tomee

to check the tomcat of prodserver

http://18.138.58.3:9090

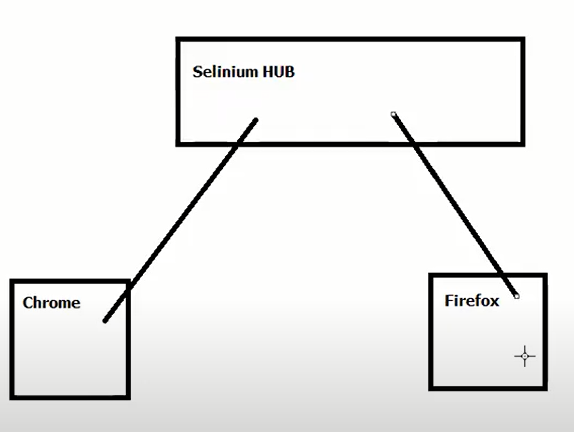
+++++++++++++++++++++++++++++

Creating testing environment using docker

Create selenium hub container, and link it with two node containers.

One node with firefox installed, another node with chrome installed.

Tester should be able to run selenuim automation programs for testing the application on multiple browsers.



To delete all the running containers

# docker rm -f $(docker ps -aq)

In Browser -- open - hub.docker.com

Search for selenium

We have a image - selenium/hub

To start selenium/hub as container

# docker run --name hub -d -p 4444:4444 selenium/hub

In hub.docker.com

we also have- selenium/node-chrome-debug ( It is ubuntu container with chrome)

To start it as a container and link to hub ( previous container)

# docker run --name chrome -d -p 5901:5900 --link hub:selenium selenium/node-chrome-debug

In hub.docker.com

we also have- selenium/node-firefox-debug

To start it as a container and link to hub ( It is ubuntu container with firefox)

# docker run --name firefox -d -p 5902:5900 --link hub:selenium selenium/node-firefox-debug

To see the list of container

# docker container ls

Note: firefox and chrome containers are GUI containers.

To see the GUI interface to chrome / firefox container

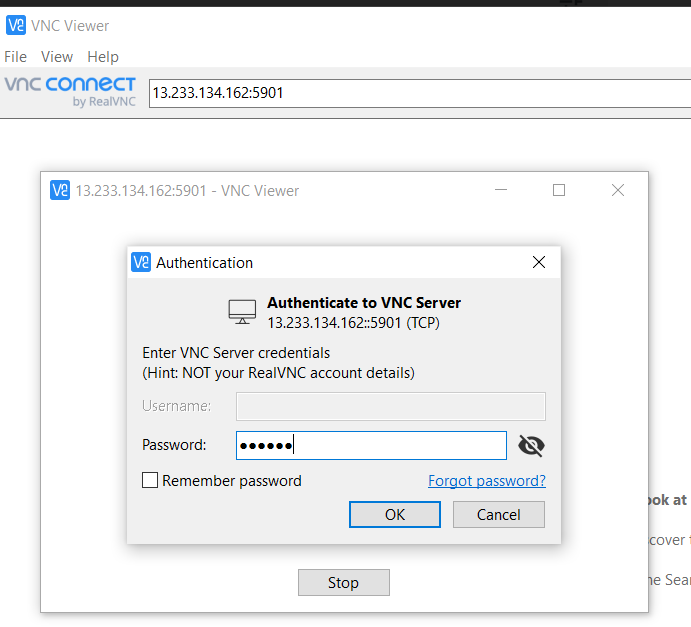
Download and install vnc viewer

In VNC viewer search bar

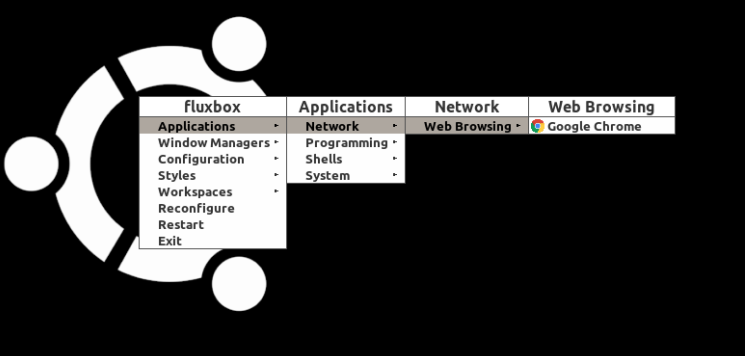
public\_ip\_dockerhost:5901

13.233.134.162:5900

Password – secret



For Chorme :



For firefox :

++++++++++++++++++++++++++++++++++++++++++++++++++++++

All the commands we learnt till date are adhoc commands.

In the previous use case we have installed two containers ( chrome and firefox)

Lets say you need 80 containers?

Do we need to run 80 commands?

Instead of 80 commands, we can use docker compose

++++++++++++++++++++++++

Docker compose :

This is a feature of docker using which we can create multicontainer architecture using yaml files. This yaml file contains information about the containers that we want to launch and how they have to be linked with each other.Yaml is a file format. It is not a scripting language.

Yaml will store the data in key value pairs

Lefthand side - Key

Righthand side - Value

Yaml file is space indented.

Sample Yaml file

---

logiclabs:

trainers:

sunil: Devops

raj: Python

Coordinators:

lakshmi: Devops

rani: AWS

...

++++++++++++++++++++++

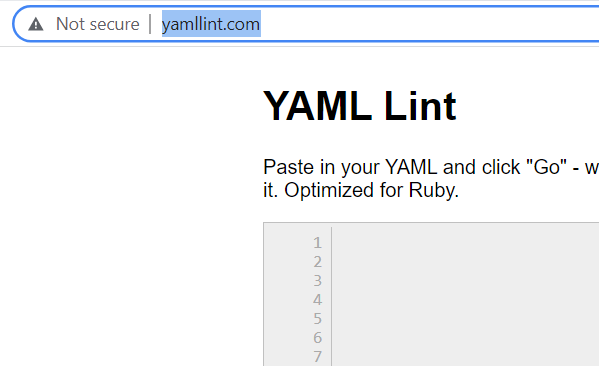
logiclabs -- root element

+++++++++++++++++++++++++

To validate the abvove Yaml file

Open <http://www.yamllint.com/>

Paste the above code -- Go button



++++++++++++++++++++++++++==

Installing Docker compose

-----------------------

1) Open https://docs.docker.com/compose/install/

2) Go to linux section

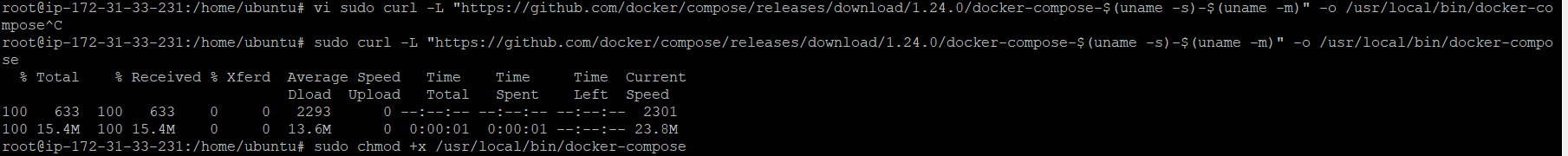
Copy and paste the below two commands

# sudo curl -L "https://github.com/docker/compose/releases/download/1.24.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

# sudo chmod +x /usr/local/bin/docker-compose

How to check docker compose is installed or not?

# docker-compose --version



+++++++++++++++++++++++++

Create a docker compose file for setting up dev environment.

mysql container is linked with wordpress container.

# vim docker-compose.yml ( Name of the file should be docker-compose.yml)

---

services:

mydb:

image: mysql:5

environment:

MYSQL\_ROOT\_PASSWORD: sunil

mysite:

image: wordpress

ports:

- 5050:80

links:

- mydb:mysql

...

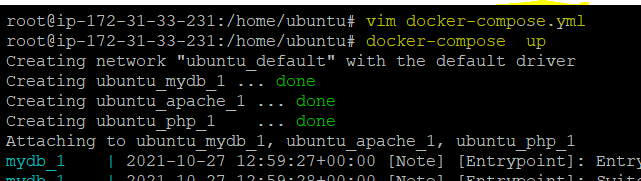
:wq

Lets remove all the running container

# docker rm -f $(docker ps -aq)

How to start the above services from yml file

# docker-compose up



We got lot of logs coming on the screen. to avoid it we use -d option

# docker-compose stop

Remove the container

# docker rm -f $(docker ps -aq)

# docker-compose up -d

To check wordpress

public\_ip:5050

++++++++++++++++++++++++++++

To stop both the containers

# docker-compose stop

++++++++++++++++++++++++++++++

Create a docker compose file for setting up LAMP architecture

# vim docker-compose.yml

---

version: '3'

services:

mydb:

image: mysql:5

environment:

MYSQL\_ROOT\_PASSWORD: sunilsunil

apache:

image: tomee

ports:

- 6060:8080

links:

- mydb:mysql

php:

image: php

links:

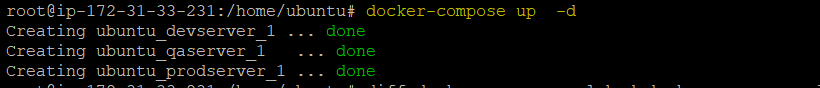
- mydb:mysql

- apache:tomcat

...

:wq

# docker-compose up –d



To see the list of the containers

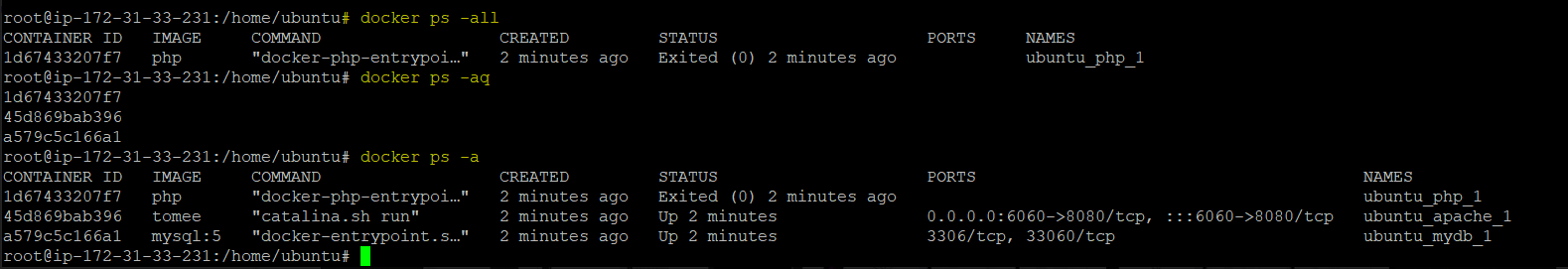
# docker container ls

( Observation - we are unable to see the php container)

# docker ps –a 🡪show all the container live in background

#docker ps –aq 🡪only show container ID (image id)

#docker ps –all 🡪only show the background container



++++++++++++++++++++++++++++++++++++

Ex: Docker-compose file for setting up CI-CD Environment.

jenkins container is linked with two tomcat containers

# vim docker-compose.yml

---

version: '3'

services:

devserver:

image: jenkins/jenkins

ports:

- 7070:8080

qaserver:

image: tomee

ports:

- 8899:8080

links:

- devserver:jenkins

prodserver:

image: tomee

ports:

- 9090:8080

links:

- devserver:jenkins

...

:wq

# docker rm -f $(docker ps -aq)

# docker-compose up -d

# docker container ls

To check

public\_ip:7070 ( To check jenkins )

public\_ip:8899 ( Tomcat qa server )

public\_ip:9090 ( Tomcat prod server )

13.126.58.183:7070 13.233.134.162

13.126.58.183:8899

13.126.58.183:9090

+++++++++++++++++++++

Docker-compose file to set up testing environment.

selenium hub container is linked with two node containers.

# vim docker-compose.yml

---

version: '3'

services:

hub:

image: selenium/hub

ports:

- 4444:4444

chrome:

image: selenium/node-chrome-debug

ports:

- 5901:5900

links:

- hub:selenium

firefox:

image: selenium/node-firefox-debug

ports:

- 5902:5900

links:

- hub:selenium

...

:wq

Lets delete all the running containers

# docker rm -f $(docker ps -aq)

# docker-compose up -d

# docker container ls

As it is GUI container,

we can access using VNC viewer

Open VNC viewer

52.77.219.115:5901

password: secret

++++++++++++++++++++++++++++++++++++++++++++++