**DOCKER**

DOCKER

* Docker is a platform that uses containerization to package applications with all their dependencies, ensuring they run consistently across different environments.
* Docker is a containerization tool.
* Traditionally for developing a application we are using hardware and then os and supported files in it and last preparing applications.
* In this traditional method for one hardware usage we can develop only one application.
* So to overcome it we came with virtualization.

VIRTUALIZATION

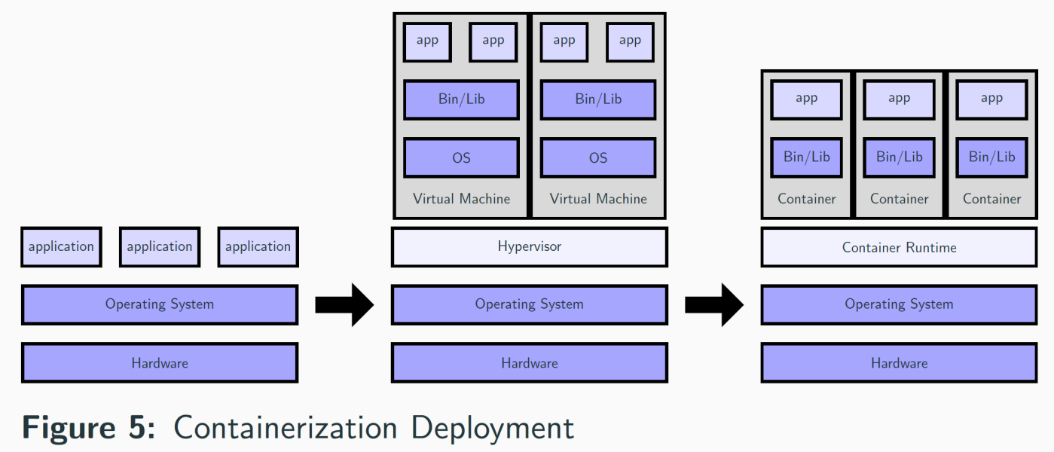
* Virtual machine is an third party we can utilise multiple os like windows, Linux, ubuntu, red hat simultaneously.
* It is virtualizing the end-end hardware in our laptop.
* It means we are using one hardware to develop multiple applications like the process is

1. Usage of one hardware
2. Here we are introducing **hypervisor** which will helps to create many applications.
3. Then os🡪runtime🡪applications.

* Docker will overcome the drawbacks of vmware.
* Here this step can be repeated no of times so that we can develop many applications.
* The disadvantage of this traditional method is **wasting of resource.**
* The advantage in virtualization is reducing the wastage of resource we are using multiple os to develop multiple applications.
* There is also one disadvantage in virtualization like for one application we are using one os so here also there is a resource wasting.
* So, to overcome this we are using docker.
* Docker is a server level virtualization tool.

CONTAINERIZATION

* The process in this docker is we are having one hardware and then we are introducing host os rather than multiple os for multiple applications and then we are using **docker demon** which is like a service and using libraries and creating applications.
* In the virtualization there is a problem facing like the developer will develop the application in one os and it is send to an tester who is using another os here will raise problem like the application is not opening from my side.
* So, here comes the topic of **containerization** that if we are storing our application in a container and sending to the tester it will work.
* Containerization is started because to reduce the resource wastage.
* The another advantage of containerization is auto scaling and auto healing.
* The advantage refers to when any container is crashed or down the containerization will help to create a new container in 1 minute because it is having os with 40mb 60mb 80mb.
* But in the virtualization it will take some time due to its os 8-12 GB, 4mb.



What dockers will do?

* Docker is a server level virtualization tool.
* Docker is created in cloud.
* In dockers the developers will develop the code and shipped to the container after building and application will be available to the end user once we deploy this containers onto web server.
* The containers consists of 1 server, 1os, 1process.
* We can create n no of the containers on hardware of the system.
* For one image we are having one container.
* For one project we are having any images and containers.
* Docker is a client server architecture.
* Developers and operation teams can also write the docker file.

Creating the docker image.

* There are 2 types in creating the docker image.

1. From docker file.
2. Using docker commands.

* Docker images will be stored in containers.
* Along with the containers images will be stored in docker hub.
* Docker image will be pulled into multiple environments.

Docker hub:

* Used to take the backup of both images and containers.
* From here we will start all the environments.

Installation of docker

* Open AWS account and login.
* Create a instance named docker and start the instance and connect it with the git bash.
* Now install the docker by using the command **yum install docker -y** or

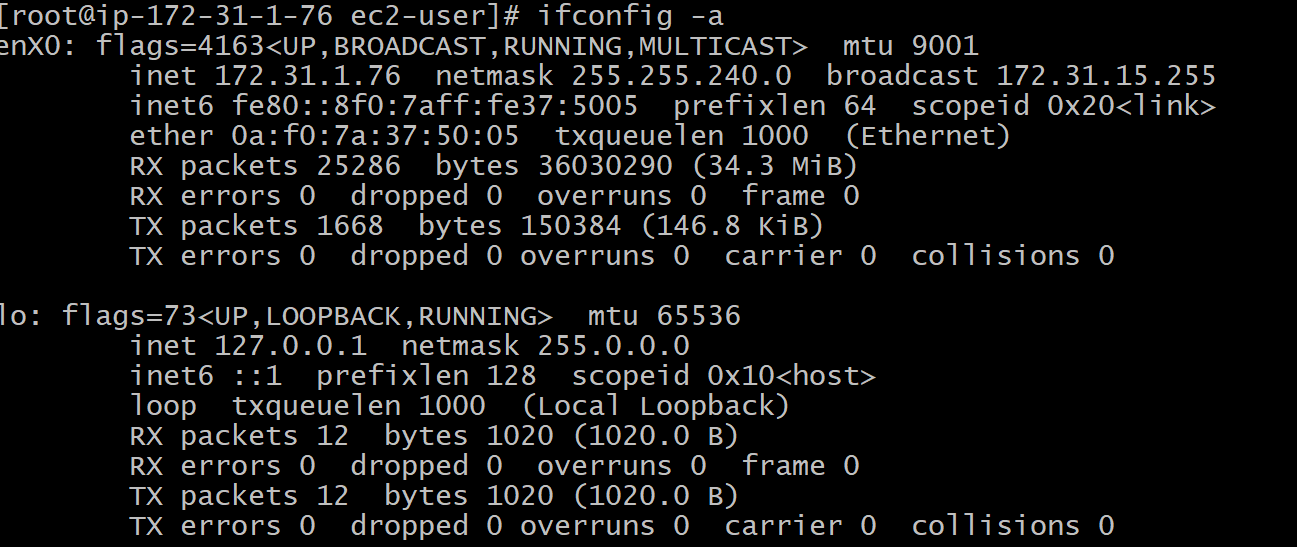
**Yum install -y docker.**

* By the above command installation will be successful and complete.
* Now create a docker hub account
* Create a two repositories in docker hub like dev and QA and also prod.

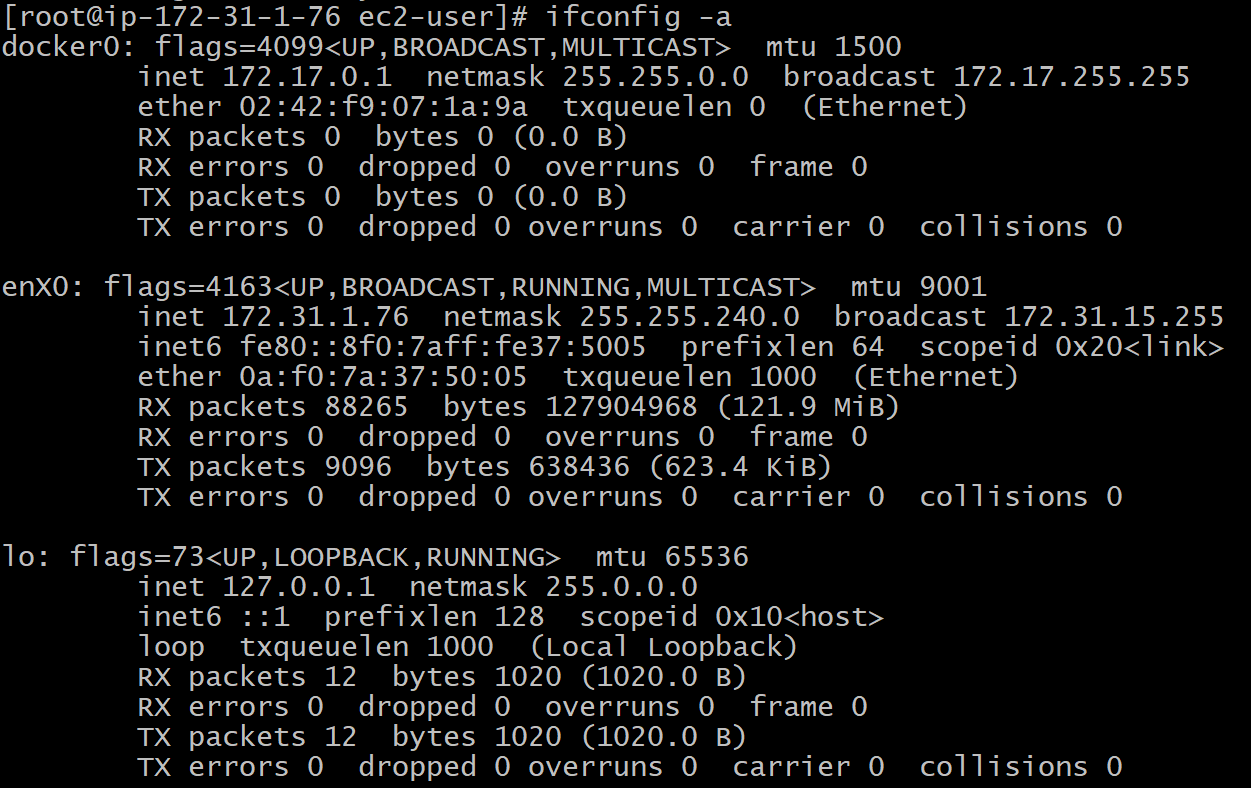
**Start using the commands**.

* Open git bash and after connecting with the Linux type Sudo su .
* Now use the command **ifconfig -a**
* By using the above command we can see the before the installation of docker if we give that command it will only 2 files present in it like enxo and lo.
* But after installation of docker it will display the docker file along with those 2 files.

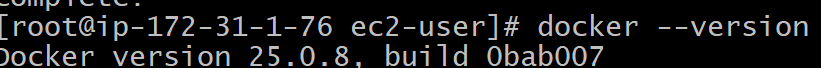
**Before docker installation**

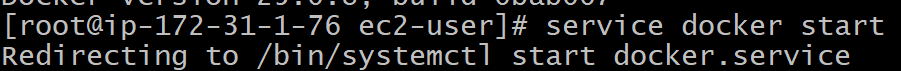


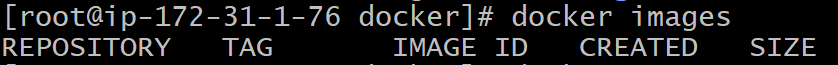
After docker installation



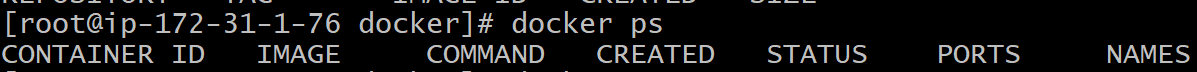
* For checking the version of docker we use **docker –version**



* The main important thing to be noticed is the docker root directory is **/var/lib/docker.**
* By the way first we should start our docker to run all the commands by using **service docker start.**
* To check is there any images present in the docker we use command like **docker images** and **docker images ls.**
* In this we can see the parameters involved in docker image.



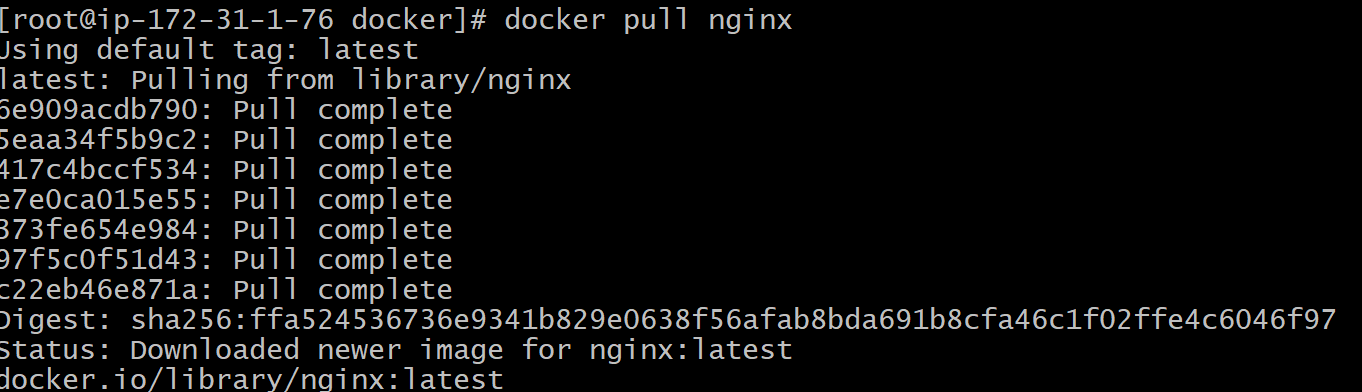
* **Docker ps** is used to know about the containers which are in running stage present.

****

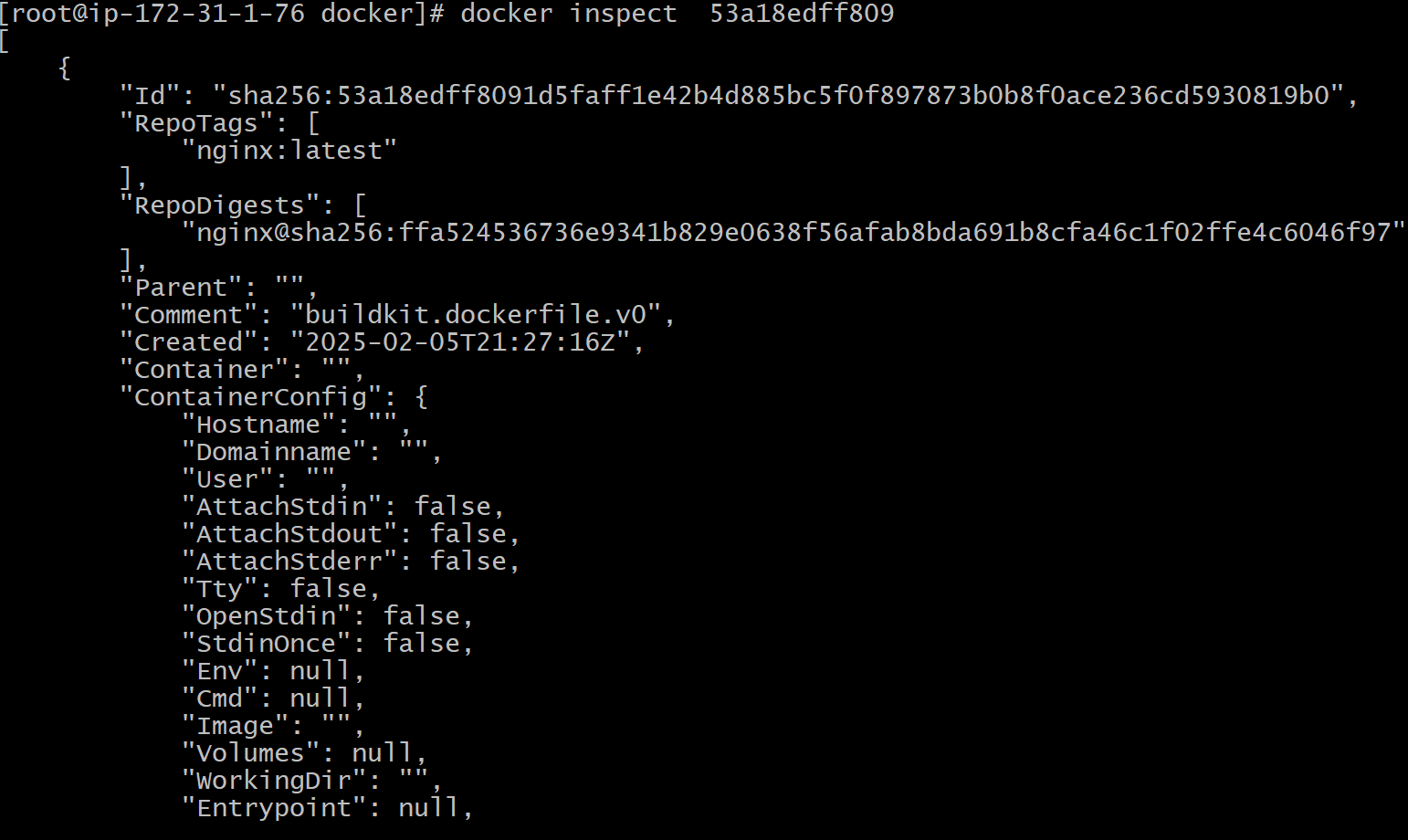
* **Docker ps -a** is used to know about the current running containers and all the containers present.
* To pull the image there are having 2 types

1. Pulling docker image from online ( official website docker hub using commands ).
2. By writing the docker file.

* Now pulling an image from the online.
* To pull an image called nginx which is a webserver.
* **Docker pull nginx** is used to pull the image.

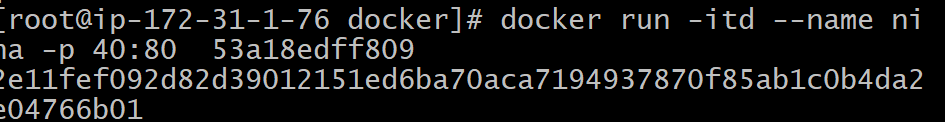
****

* The important thing is in the image id it will consider the first 12 numbers or characters in it.
* To know the particular total information about the image or container we use the command like **docker inspect imageid**

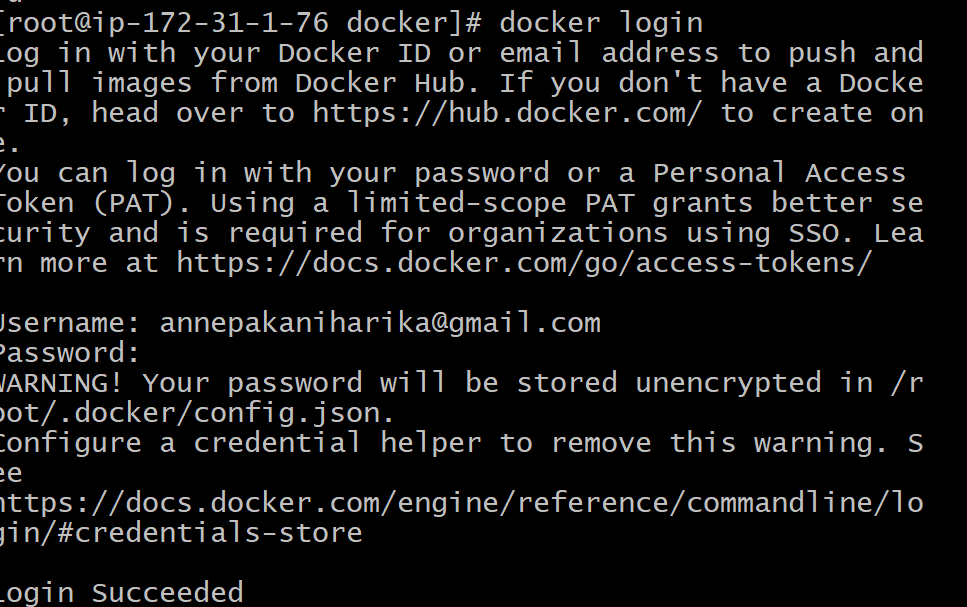
****

**To create a container**

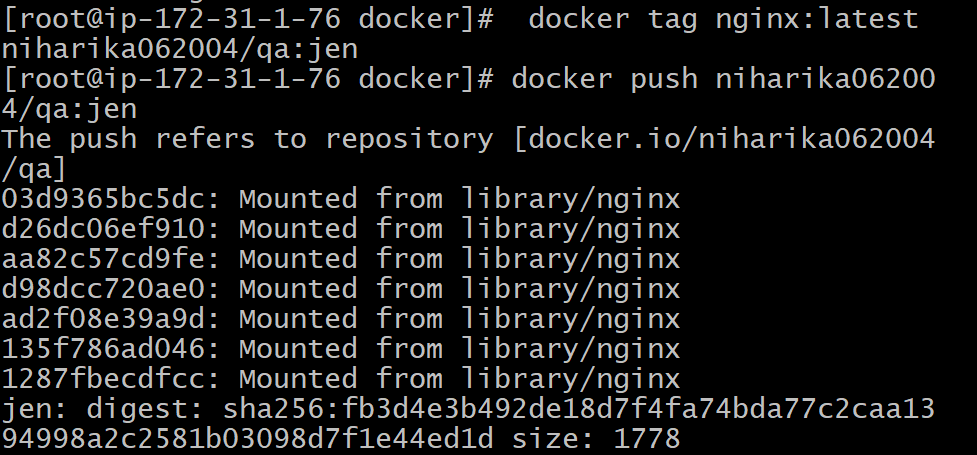
* The syntax is docker run – itd – name niha -p <cid:apn> imageid
* Itd: interactive detached mode
* Apn: port number



* To login into the docker we use **docker login**



* Next we need to tag the image that to enter into a repository we use docker tag nginx:latest niharika062004/qa:jen
* For pushing we use docker push username/qa:jen



* The command docker exec -it containerid /bin/bash

