**DOCKER**

In the microservice architecture we have many small microservices which may or may not be build using dirrenent frameworks and languages so we want to deply all of them irrespective of the language they are created in.

So docker came up with the concept of docker images.

In a docker image we have the jdk,the project and its dependencies which it needs to run and this image can be deployed anywhere may it be th local machine, some server or the aws cloud.

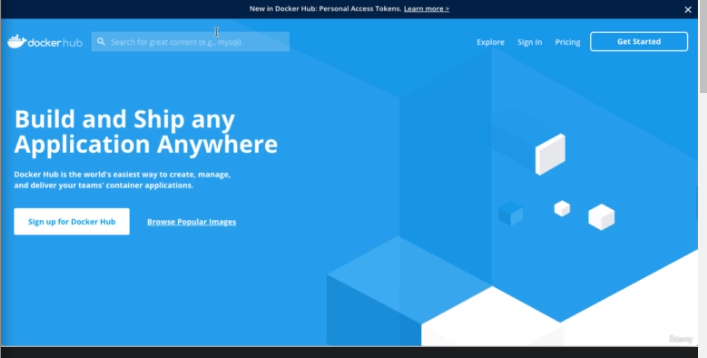
A screenshot of a computer

Description automatically generated

**DOCKER CONCEPTS**

we go to hub.docker.com

hub.docker.com is a docker registry



a registry containsa lot of repositories.anyone ca access a public registry and we will be using a private repository so that it can only be accessed by authorized people.

By default any container that we run is a part of bridge network in docker.it is internal docker network.

Now we will be ablee to access the application on port 5000.

QUICK REVIEW

1.when we exxecute the command the image is downloaded from the docker hub

2.Docker hub is known as a docker registry

3.A registry contains a number of repositories

4.Running version of an image is called a container and image is a static version

Stopping a running container

ctrlc will kill the conatiner

you want your application to be always running then we can do as below:

-d stands for detached mode means container will run in the background.

You will get a container id which u can use to see the docker logs



if you want to keep folllowin the logs of that application

docker logs -f the container id.

2. to see what containers are running

A black text on a white background

Description automatically generatedname blissful-wing was assigned to the container

3.Multiple containers from the same image

now we can run the same application on port 5001 which was previusly run on 5000.(one image many containers)

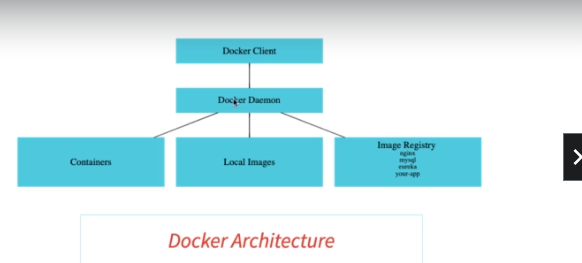
after this we can check how many containers are running y running the above command

3.now if we want to see how many docker images we have

A close-up of a screen

Description automatically generateddocker images will sow us the images which are local to us.It means the ones which have been pulled from the docker registry.

**UNDERSTANDING THE DOCKER ARCHITECTURE**

****

**docker client**

docker client is something like a docker desktop.docker client is responsible to send commands which will be executed by docker daemon.

**docker daemon**

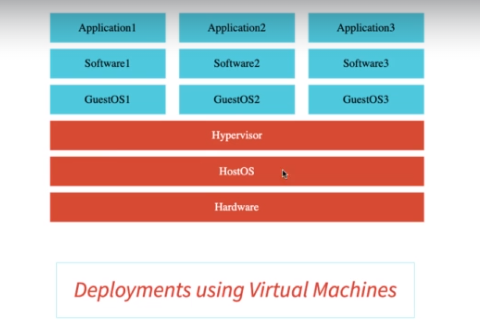
docker daemon is responsible for storing the local images ,running the images as containers and then storing them and retrieve them when needed from the image registry.

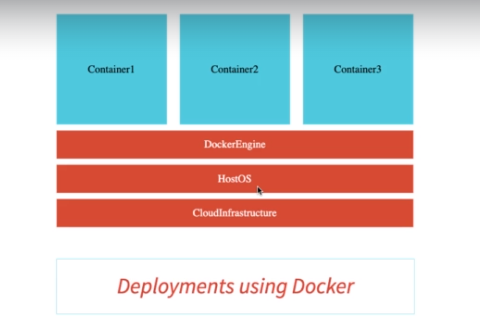
**WHY IS DOCKER POPULAR?**

Using docker on the local is very easy as well using docker on the cloud is also very easy.

We can very easily deply a docker applicaton to any cloud environment very easily as today basically every environment will give us docker conainer support.

Earlier it was very hectic to deply an application as application had to be deployed ona differet vm.

But now a days it has become quiet easy to do so as docker engine can be put on every osas.



V

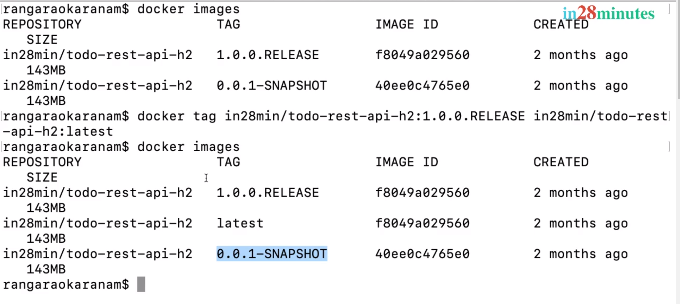
**PLAYING WITH DOCKER IMAGES**

**1.seeing the docker images**

we can run the command as docker image sto see the various images available to us.

2. **now if we want to rename the various docker images then we can use the below command as can be seen in the iamge below:**

we are changing the tag name from 1.1.0.release to latest



**3.docker pull**

we want to pull a mysql image then we can run below coomand then it will download the latest relaeese.

Docker pull mysql

pull will simply download the image from the regitry it willl not run it.

Mysql is an official image and an  **official image**  is an image which is produced by the docker community and are put on the docker website.

4.**history of an image**

we can check the history of an image by using tthe image id which is as listed below

A screenshot of a computer

Description automatically generatedthe image history shows how that image was created and what further operations were performed on that image.

6 **docker image inspect <imageid>**

when it was created ,what containers were created

etc.

7**.remove a docker image**

docker image remove <imageid>.image will be removed from our local.

**PLAYING WITH DOCKER CONAINER**

**1**.running a docker image as a container

this in turn will give us a docker container id

****2.**pausing a container**

docker container pause 6478<part of container id>

ddocker container unpause <part of container id>

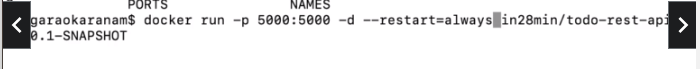
3. **REMOVE ALL STOPPED CONTAINERS**

docker container prune

4. **stop container**

docker contaier stop <container id>

**5.restart a container**



**PLAYING WITH DOCKER COMMANDS**

**1.docker events**

it shows what events are takng place.what events are happening in the backgound.

If some network connected,volume created etc.

**2.docker top**

this command is used to check what process is running and in a container

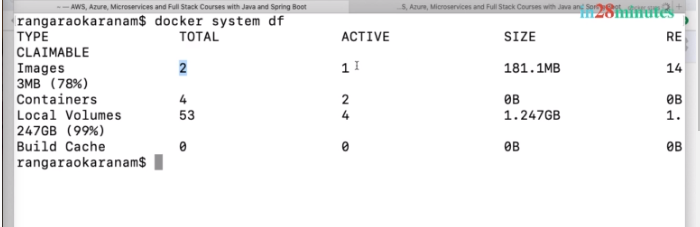
docker top <container id>

3. **docker stats**

to see mem and cpu utilization of images

4.**docker system df**

it will give info as below:



**A white rectangular box with black text

Description automatically generated**