Sr. No. _____ Date : _____

Docker

⇒ Before Dockers →

> Only one application on one server.

Vmware → server it meaning → You can run multiple applications on the same server.

- Problem in VMware is that you need to install the dedicated operating System.

Another problems:

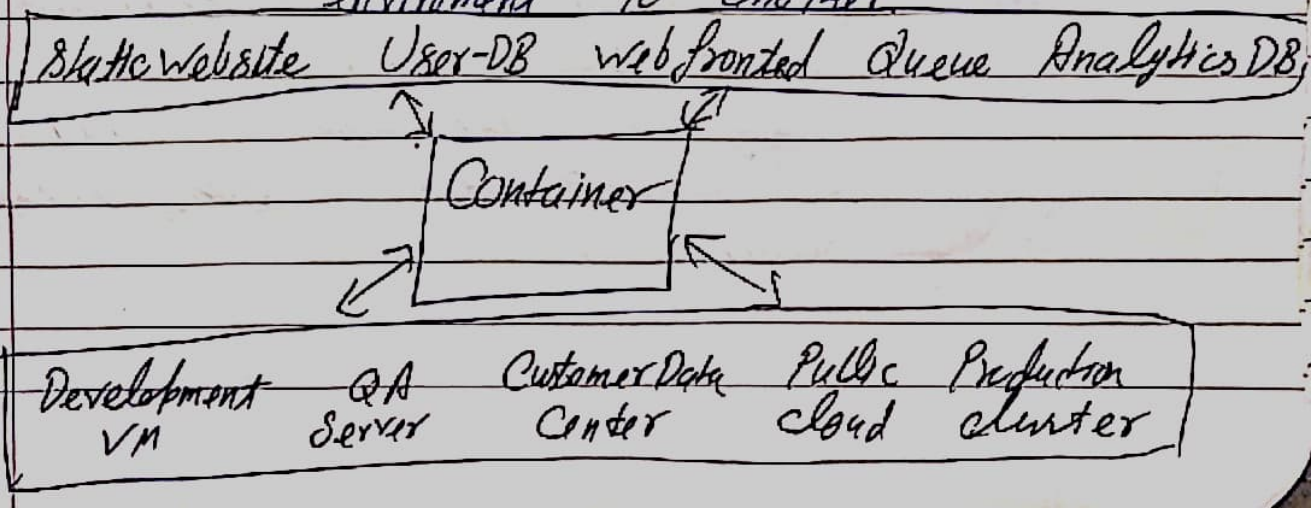
> Dependency management

> it works on my machine it doesn't work on your machine.

> Shift your code from one particular point to another particular point [Migration]

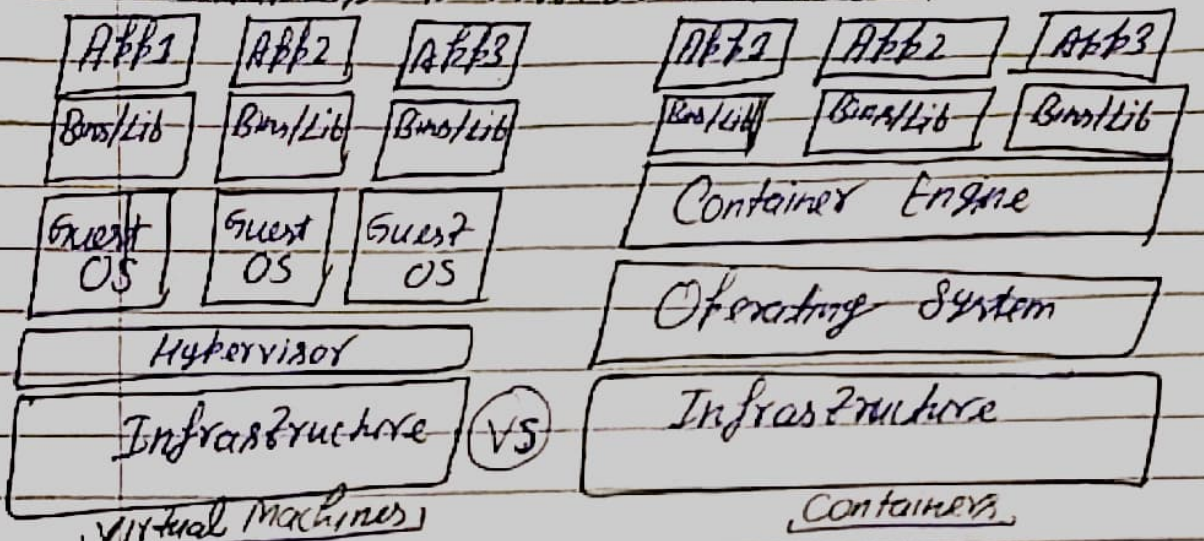
> You want to set up the project on your local system

⇒ Container → Packages up code and all its dependencies so the application runs quickly & reliably from one computing environment to another.



Dr. No. _____ Date: _____

⇒ Containers vs Virtual Machines →



Hypervisor is used to create multiple machine on a host OS & it manages virtual machines [VM].

⇒ Running Docker on Windows →

Windows: Docker Desktop + WSL

↓

Windows Subsystem for Linux

⇒ Runing Docker on MacOS →

VM & Docker Desktop

⇒ On Linux → As it is

Note: If a windows app is containered it will not run on linux based docker, Similarly for Linux based docker it will not run on windows based docker

Sr. No. _____ Date: _____

⇒ What is Docker →

Docker is a Container platform that allows you to build, test & deploy applications quickly.

⇒ Installation → docs.docker.com

⇒ Docker Runtime →

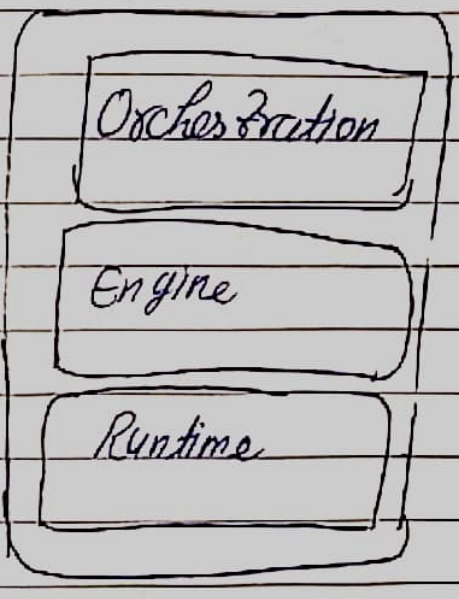
Runtime basically allows us to start and stop Containers

2 type of Runtime!

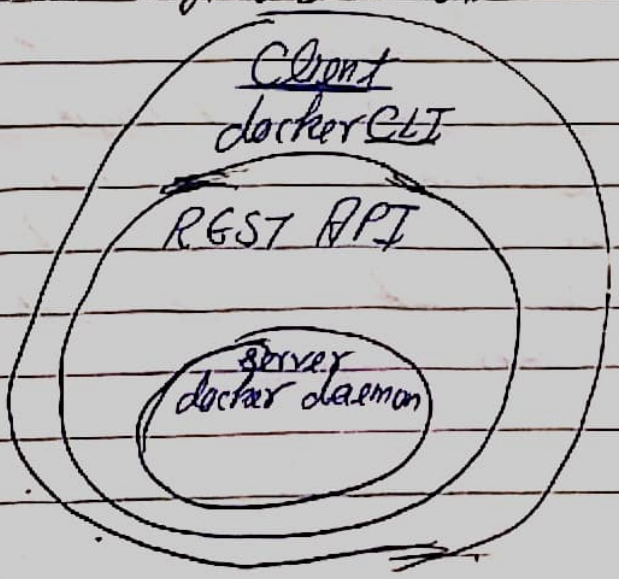
1 - runC [low-level] →
It works with OS & start and stop the Containers.

2 - Containerd → It managing runC and Containerd does all the other thing like data, Connecting with network etc.

It also used in Kubernetes runtime



⇒ Docker Engine [daemon]

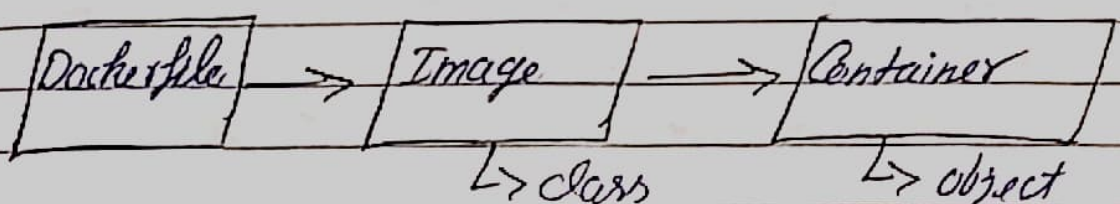


Sr. No. _____ Date: _____

⇒ Docker Orchestration → Docker orchestration is a set of practices and technologies for managing Docker Containers large scale.
e.g → if you want scale application 10,000 Container to 20,000 Container then you use Docker Orchestration.

⇒ Docker / Container Image ⇒ That includes everything needed to run an application: code, runtime, system tools, system libraries & SetAnys.
Docker Image is run to create a docker Container.

⇒ Dockerfile → Describe steps to create Docker image



Explanation → When you have your own application that you want to containerize you have to write docker file for that and it will converted to an image that image you can share to other peoble.

Sr. No. _____ Date : _____

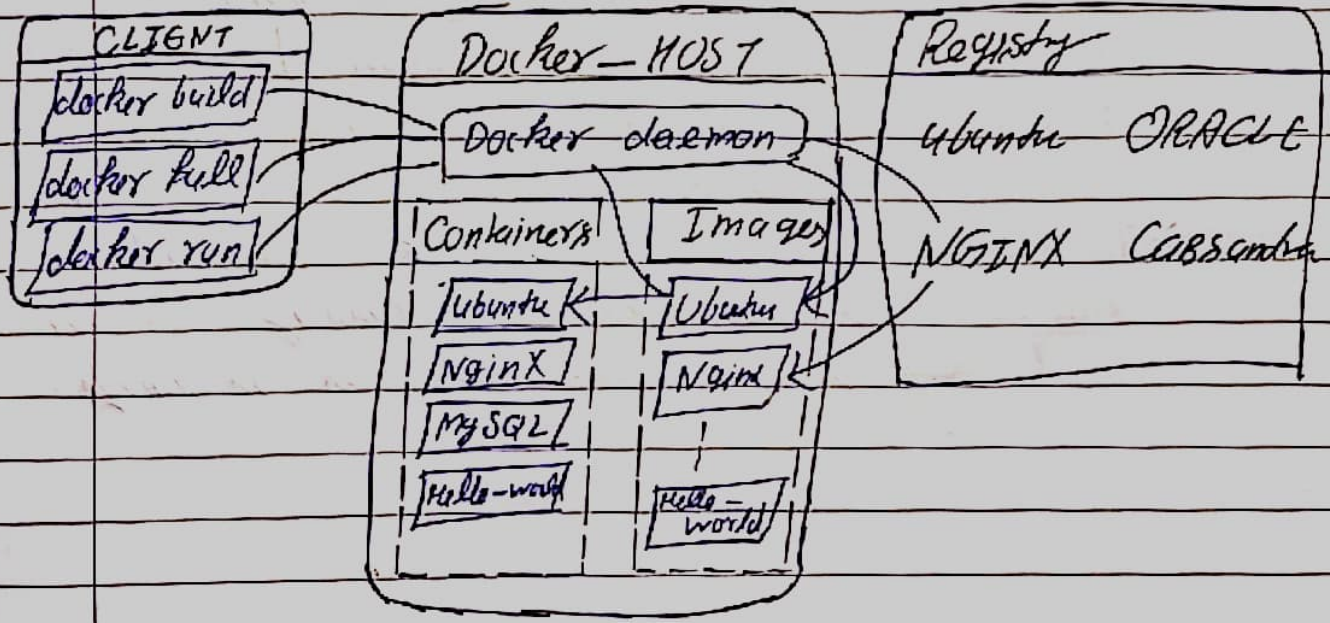
⇒ Open Container Initiative (OCI) → OCI is Linux Foundation Project to design open standard for Containers. OCI currently Contains two specification : runtime-spec and Image-spec.

⇒ How the CLI works →

$ docker run hello-work
↳ Imagename

If you ~~this~~ running "hello-work" for first time, then it will download the image the from the docker registry. Next time you run this command then it will not download this image again because image is stored locally.

$ docker image → This command show all the image that are locally downloaded



Sr. No. _____ Date : _____

⇒ Accessing a Container locally →

$ docker run -p <hostPort>:<ContainerPort> imageName

e.g → $ docker run -d -p 8080:80 nginx

The traffic is inside this Container [nginx], the port 80 forward all the requests that you're making to the localhost:8080 inside the Container port 80.

⇒ Docker Commit → You essentially create a new image with an additional layer that modifies the base image layer.

$ docker Commit -m "added new file" <ContainerID> <set-newName>
1 2

When executing the $docker Commit command we will need to provide two parameters: the name of running Container [ContainerID] and the name of our desired image output [name_ubuntu].

→ In simple word, Docker commit Command allows user to take a running Container and save its current state as an image.

Sr. No. _____ Date : _____

⇒ Removing Docker images →

$docker image -q List/show all the images

$docker rmi $(docker image -q)

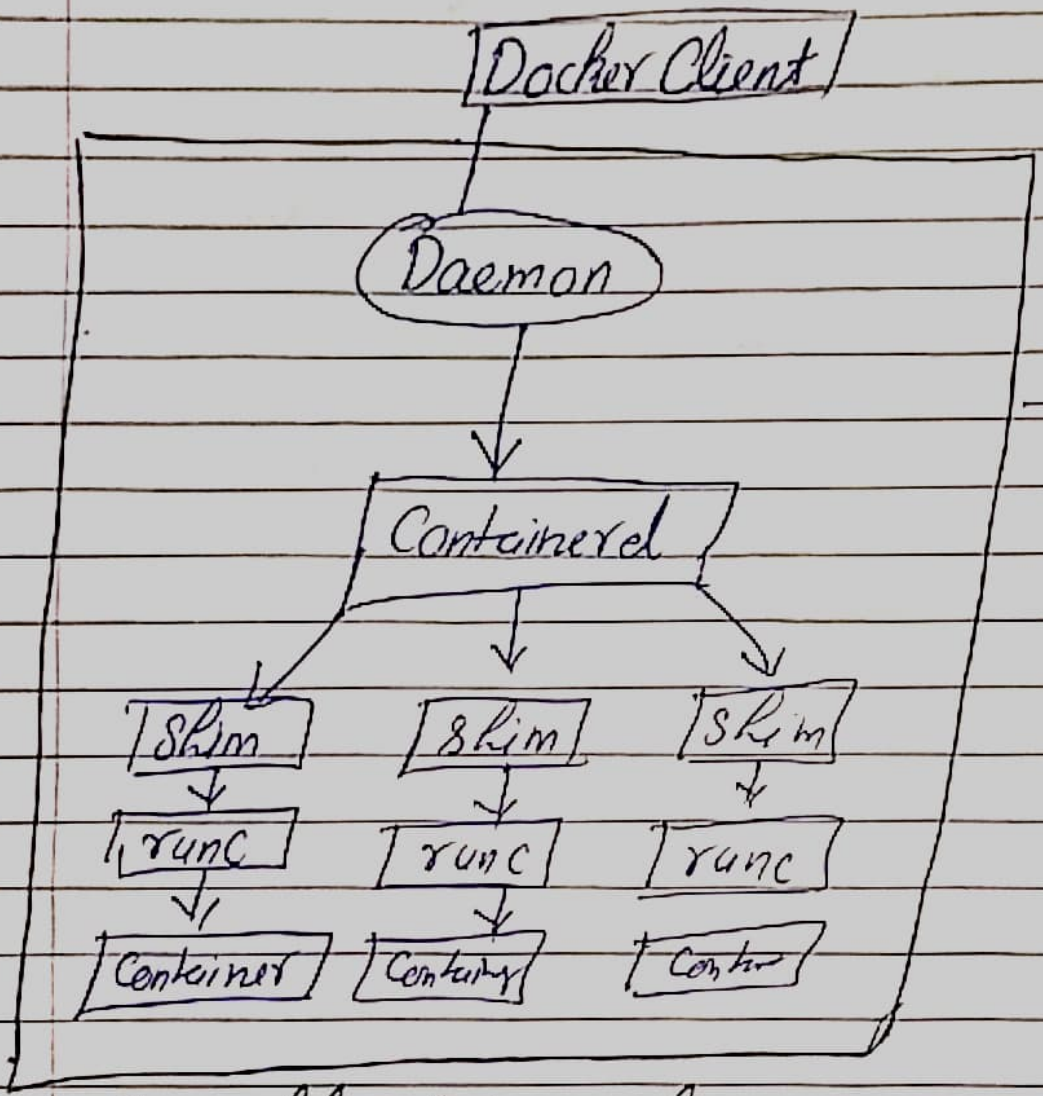
⇒ Layers → Images are built in layers. Each layer is an immutable file, but is a collection of files & directories. The last layer can be used to write out data to. Layer receive an ID, calculated via a SHA 256 hash of the layer contents.

⇒ How to create Docker image →

- Create a file name "Dockerfile"
- By default on building, docker searches for 'Dockerfile', $docker build -t myimage:1.0 .
- During building of the image the command in RUN section of Dockerfile will get executed $docker run $image ID
- The command in CMD section of Dockerfile will get executed when you created a container out of the image

```
FROM ubuntu
RUN apt-get update
CMD ["echo", "Hello guys!"]
```

⇒ Architecture of Docker Engine ⇒



Binary files that are linux system

- Daemon → dockerd
- Containerd → docker-containerd
- Shim → docker-containerd-shim
- runC → docker-runc.

Sr. No. _____ Date: _____

⇒ Docker Commands →

- $ docker pull <image name>
  → This command is used to pull image from docker repository.

- $ docker run -it <image name>
  → This command is used to create a container from an image

- $docker ps
  → This command is used to list the running containers
- $docker ps -a
  → This command is used to show all running & exited containers
- $docker exec -it <container ID> bash
  → This command is used to access the running container.
- $docker stop <container ID>
  → This command stop a running container
- $ docker commit -m "--" <container ID> <new-image name>
  → This command creates a new image of an edited container on the local system
- $docker login
  This command is used to login to the docker hub repository.

Sr. No. _____ Date : _____

- $docker push <image name>
  → This command is used to push an image to the docker hub.

- $docker images.
  → This command lists all locally stored docker images.

- $ docker rm <Container ID>
  → This command is used to delete a stopped ~~Comman~~ Container

- $docker rmi <image-ID>
  → This command is used to delete an image ~~fo~~rom local storage

- $ docker build -t myimage <path of Docker file>
  → This command is used to build an image from a Sheatic docker file.