

Docker

Page No.	
Date	

Intro -

Before docker -

Containers -

Containers vs Virtual Machines -

History of Docker

Running Docker on Mac / Windows / Unix

What is Docker

Installation

Getting started.

Docker Runtime

Docker Engine

Orchestration

Docker / container Engine

Diff. n Dockerfile vs Engine

Open container Initiative

Docker Desktop

What is perms

Docker CLI

How the CLI works

How Docker Engine works

Downloading Docker Engine

Some Additional Docker Commands

Accessing a container locally

Docker Commit

Removing Docker Images

Layers

How to create Docker Image

Architecture of Docker Engine

Intro.

1>

Intro —

Docker N/w
Docker security
Docker Best practices
Kubernetes
Docker Composed env
Containers working

coming soon.

2>

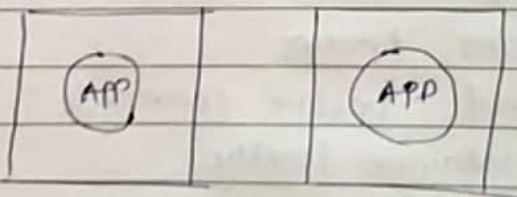
Before Docker —

At initial time Application run on server
But only one app. run on server

one application → one server.

let's 2000 people also using application
load will increase on server, then we
want more server.

means if we didn't have a way
to run multiple appl. on servers.



You need always want
new server to run
new application.

* Inho solve this problem VMware came
= Virtual Machine.

VM means u can multiple appl. on
one servers.

Now, let's talk about new problem that always be center as a developer, -

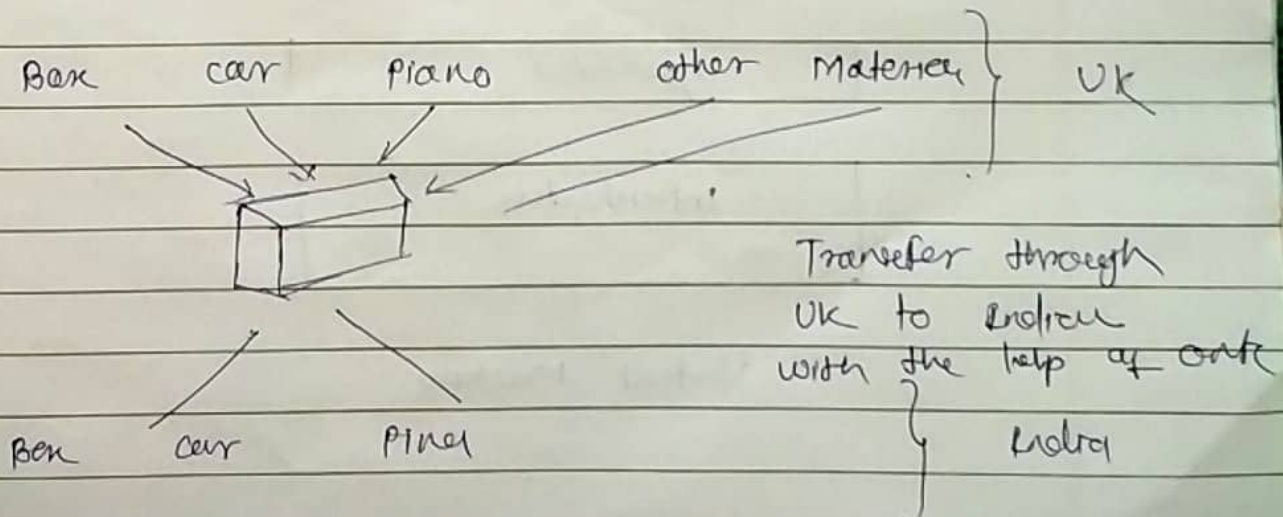
What happen, u built a application and u share with your friend and he said it's not working on my machine, because of different modules, packages like that. and this problem also solve docker.

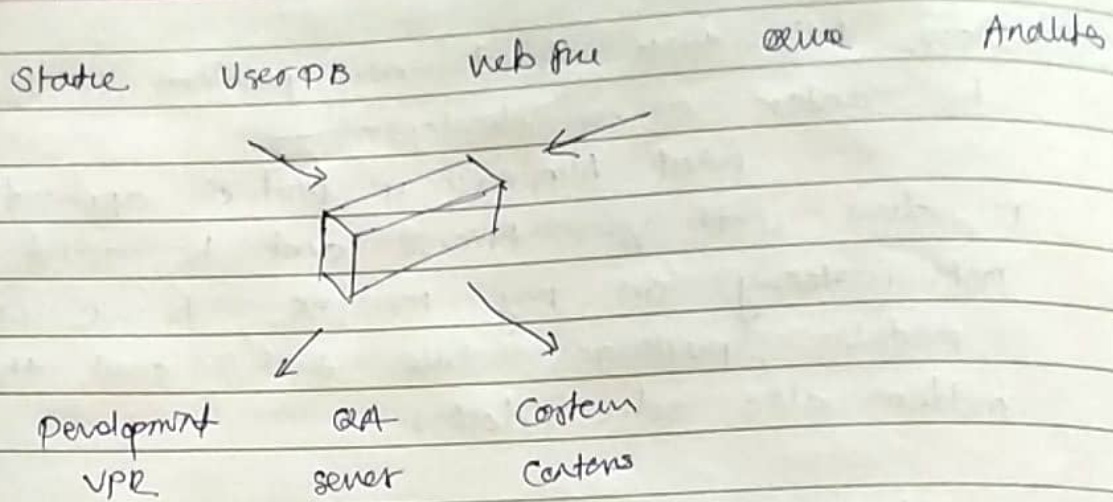
Resolution - Suppose we have computer with different hard disk on one hard disk install ubuntu and second one is windows. So ubuntu does not have any idea about what happen on windows.

8) Containers

One OS and we are running diff. appl. on that particular one OS - and this problem is solve by containers.

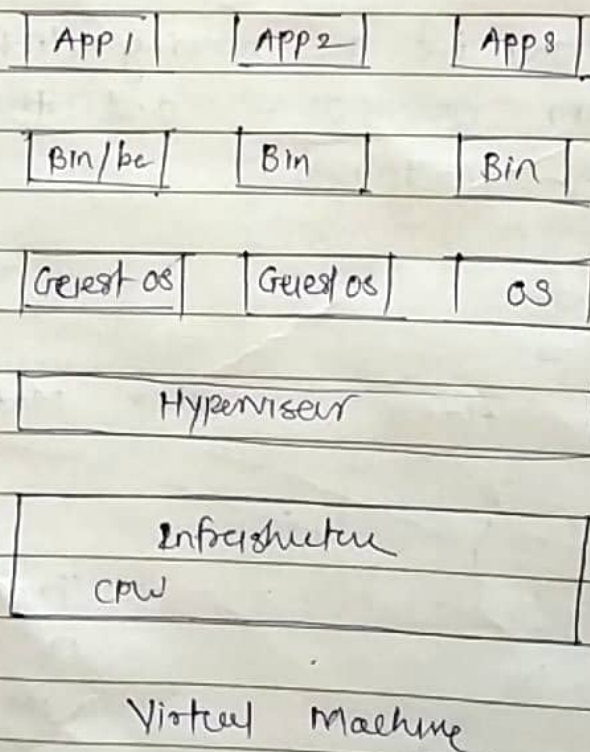
Containers -





eg. So, everything that you build a website that you put on the box and you send your friend or show your friend. Now, it make sure that web. is running on friends computer.

21) Containers Vs Virtual Machine —

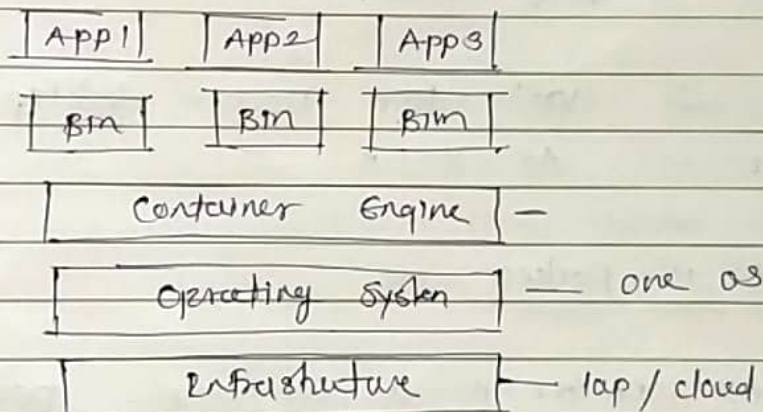


Hypervisor - create multiple machines on a host OS and it manages VM.

These VM have their own operating system and do not use the host operating system. They have some space also allocated.

On VM's the virtually divided our appl.ⁿ on diff.ⁿ OS / Infra.

Containers -



In container, We Run multiple appl.ⁿ on one operating system. using the container engine.

5) History of Docker -

Docker is a tool to manage container things, eg. manage container, scale containers
Docker is a company.

6) Running docker on Windows -

Windows - docker peplay

eg. Linux based containerization is not run on window based docker.
vice versa.

Mac - VM's for Docker desktop.

Linux - As it is.

7) What is Docker -

At Docker is a company. Docker is a container platform that allows you to build test, deploy appli. n. quickly.

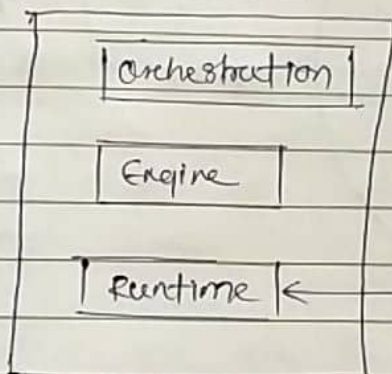
8) Installation -

I run some commands on Linux to installation.

g) Getting Started -

Docker is helping us to run our app. in on isolated environment. So container know it's self known as isolated env. - container may not know outside world.

10) Docker Runtime -



Allows us to stop and start containers.

Run time works with OS and stop / start the container

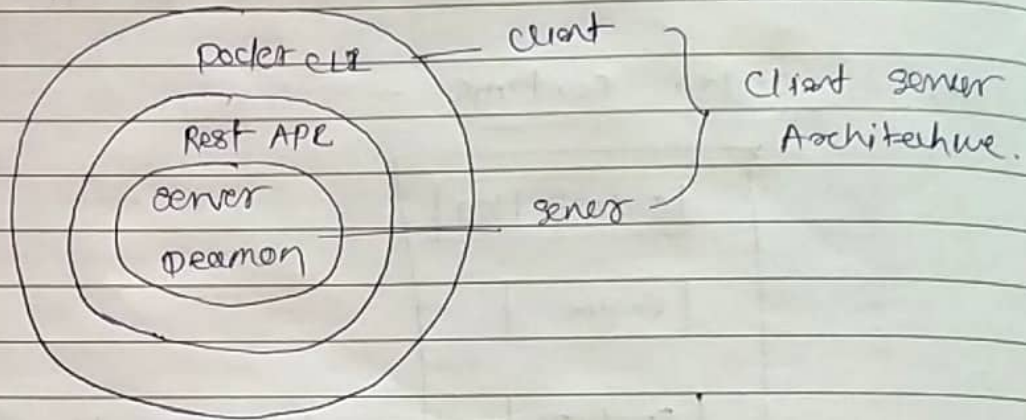
① Runtime - containerd - helps in managing container
eg. how to interact container with the n/w internet
pulling the images.

- run c - start / stop the container.

② Engine -

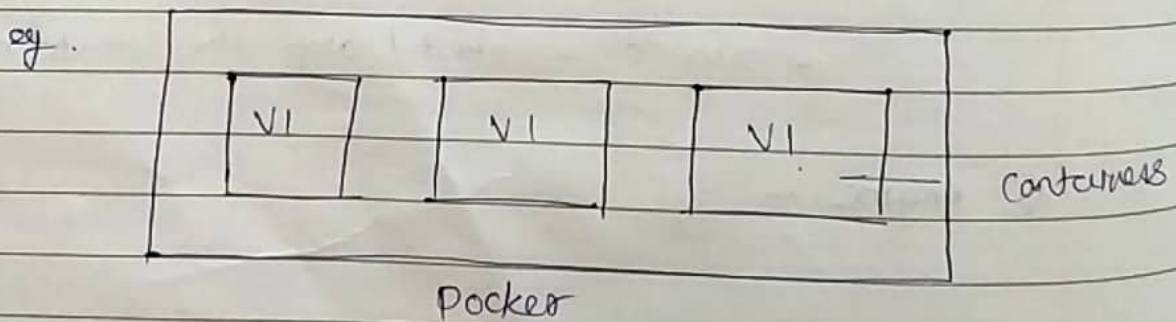
11) Docker Engine — Docker Daemon

Docker Engine we are using for to interact with docker.
eg. Docker daemon.



12) Orchestration — / Orchestration Engine —

let's say your application is running on 10,000 containers and you want to scale up to 20,000 containers, let's say few container got crashed then you need to restart those containers.



All container's are V1 appl.ⁿ and u want V2 appl.ⁿ, what do you do stop the entire

Page No. _____
Date ____/____/____

applications, are you updated once by once. This is very time consuming process. If we do manually. What if this is happening automatically at its own. Then this process come into the picture Orchestration Engine. The most popular one is Kubernetes -

In Simple terms, It's allows to manage container

13) Container Image -

Ex. eg. let's say, you are eating something, just like noodles and your friend is lives in another country. He lives in Bangalore. then you say for your friend this noodles is soap is amazing. How can I give my noodle soup to him. Take my noodle and put in box and give then. and this not makes any sense. By the time noodles reach to friend it will be very bad.

eg. so like wise you cannot share running appl.ⁿ to your friend.

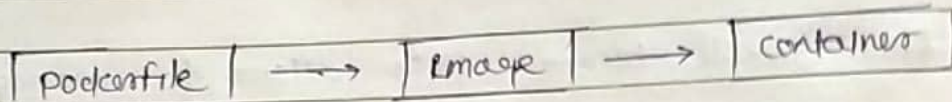
What we can do, we can share the instruction. This instruction part mean we have dish means actual appl.ⁿ that is running on the server.

Image - It's just a file it contains all the instruction.

14) Dockerfile -

Dockerfile is your recipe.
Run this command, copy the files, export

Dockerfile -



When u have your appl'n that u need to containerize. You have to docker file to that so you can converted to image and that image you can share with others

15) Open Container Initiative -

There are many many container runtime which one should I use - container P Arch T. If I'm using container P does not mean that I not using R means what is happening here, They come up with the solution of Open Container Initiative.

Open Container Initiative - project under the Linux foundation.

Kubernetes using the run time container run time interface

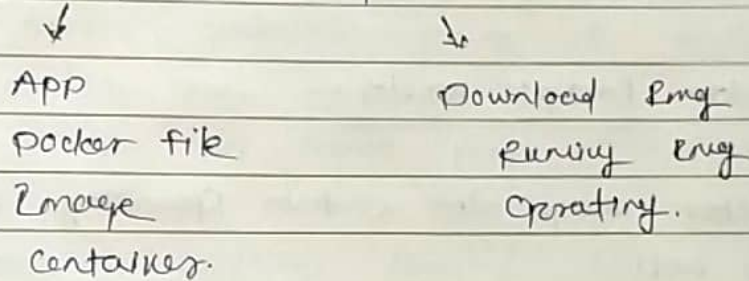
16) Docker Desktop -

`docker run -it ubuntu`

`docker ps` → Running Containers

17) What is DevOps -

Development + Operation -



18) How CLT Works -

`docker run` hello-world

↓ ↓

Run an image Image Name.
to create container

`docker images` - To look at the Images.

Client → Docker Daemon → Registry
We fire a Repository of
command over Docker commands.
here

U can go there to check images of particular things.

eg: U want to run mongo DB on your machine but you don't want to install it.

Container is inside my tool - docker -
 docker is tool → Basically it's communicating my own OS only. but it's isolated.

18) How docker Images Works -

Docker image also contain Operating system file as well

19) downloading docker Image.

* docker pull mysql -

if I don't provide version then it will automatically latest version.

* docker images.

* docker run -it ubuntu

Interactive enviro. → don't exit out of it since I'm in container.

* ps docker ps

* docker container -ls

} list of container

docker container exec -it [ID of another running container]

- Means we can go inside the container with the help of this command.

docker stop [ID of container] - To stop container

docker ps -a

docker remove m [ID]

docker logs [ID]

docker containers prune -f = delete all the stopped

docker run

docker run nginx proxy www.civo.com -

20) Accessing container locally -

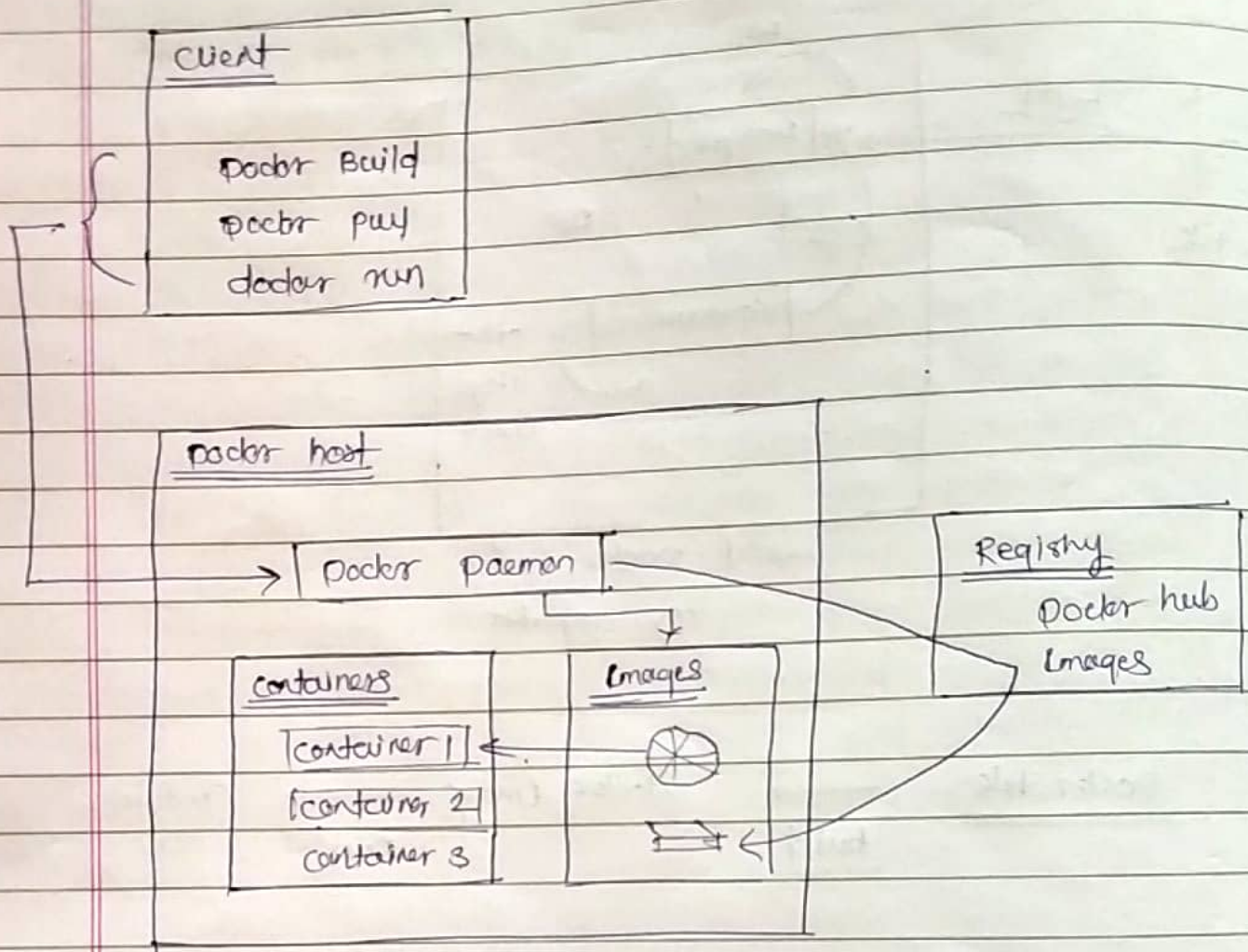
docker run -it -p 8080:80 nginx

→ Run so the nginx is run on 80 port and I wanna access it on my port eg. 8080 forwarding the port.

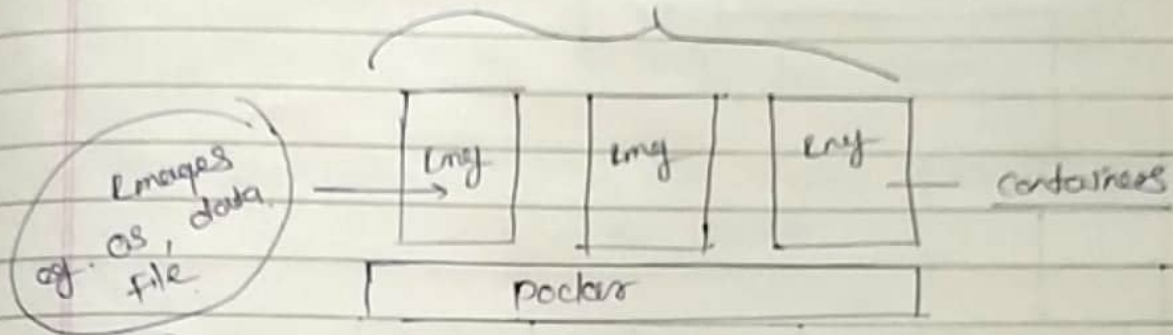
2) Docker Commit :

eg: You made some changes within your container and you need to share with your friend and make sure that those changes also be visible.

then we use commit



Kubernetes — Advanced version of
podman swarm
↓
podman swarm — manager



Project — Azure cluster using
3 namespaces — Vault install
kubernetes
goharbour

Ingress → Traffic (private → public)
load balancer server

