

DOCKER

Before docker

- only 1 application on 1 server
- new server for every application

VMware solved this problem using virtual machines.

- now you can run multiple application on same server
- but still every virtual machine requires its own OS

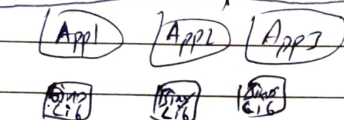
~~docker is trying to solve~~
problem you may be facing right now

- dependencies issues
(~~mod~~ diff packages, modules etc)
- setting ^ag up project on your own system.

Adding virtual machines req hardware, os etc., so to ~~hate~~ avoid wasting your computing resources containers are made.

Containers

→ no os req

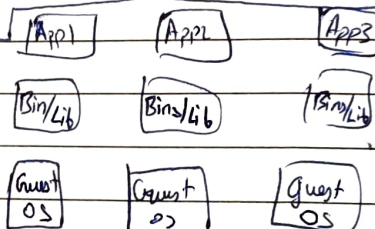


Containers Engine

OS

Laptop

Virtual machines



Hypervisor

Laptop/Server/etc

~~Container~~

this is known as Docker

Hypervisor:

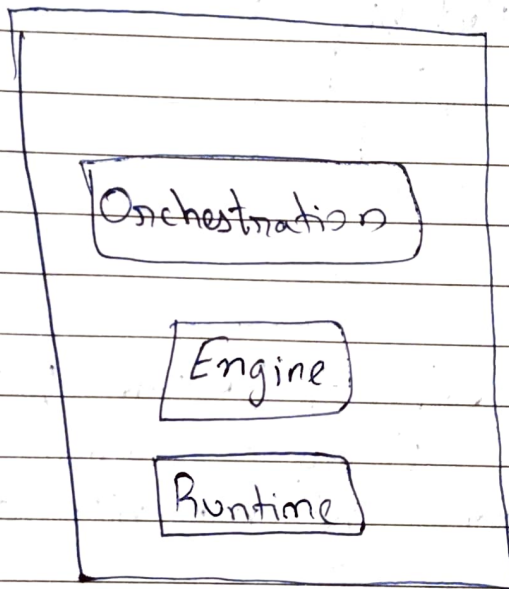
→ creates multiple virtual machine on host OS and manages virtual machine.

Docker

① Runtime
all
Th
⇒

Docker

contains



1) Runtime

allows to start & stop containers

Two types

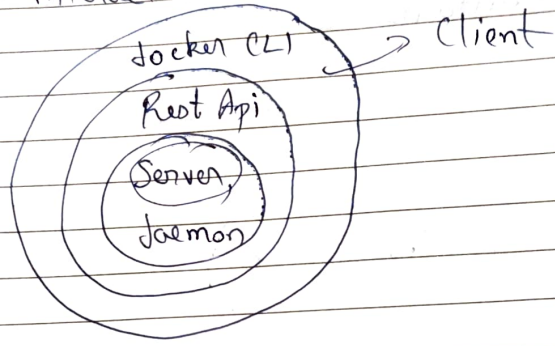
⇒ runc (Low level runtime)

- works with your OS and starts and stops the containers

⇒ containerd (higher level)

- managing runc
- helps containers to interact with networks
- container runtime for Kubernetes

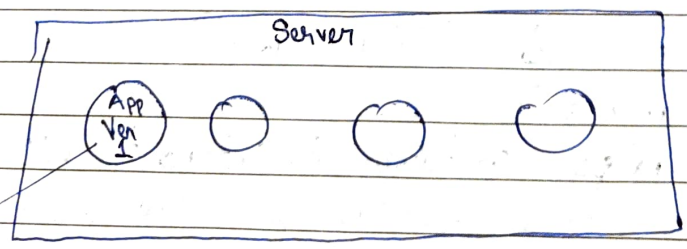
(II) Engine, Daemon
→ interact with docker



→ Using docker cli & then rest api we can make api call to daemon
→ daemon will work with containers runtime & do it.

→ docker CLI
• basically with what we interact with

(III) Orchestration



Apps version 1

→ Suppose you need to update these apps to version 2, you can't really stop the server

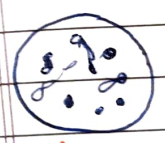
on do 1 b
So here o
labour.
These open a
engine like
much more m

Example

Me (Delhi)



↓ made pizza



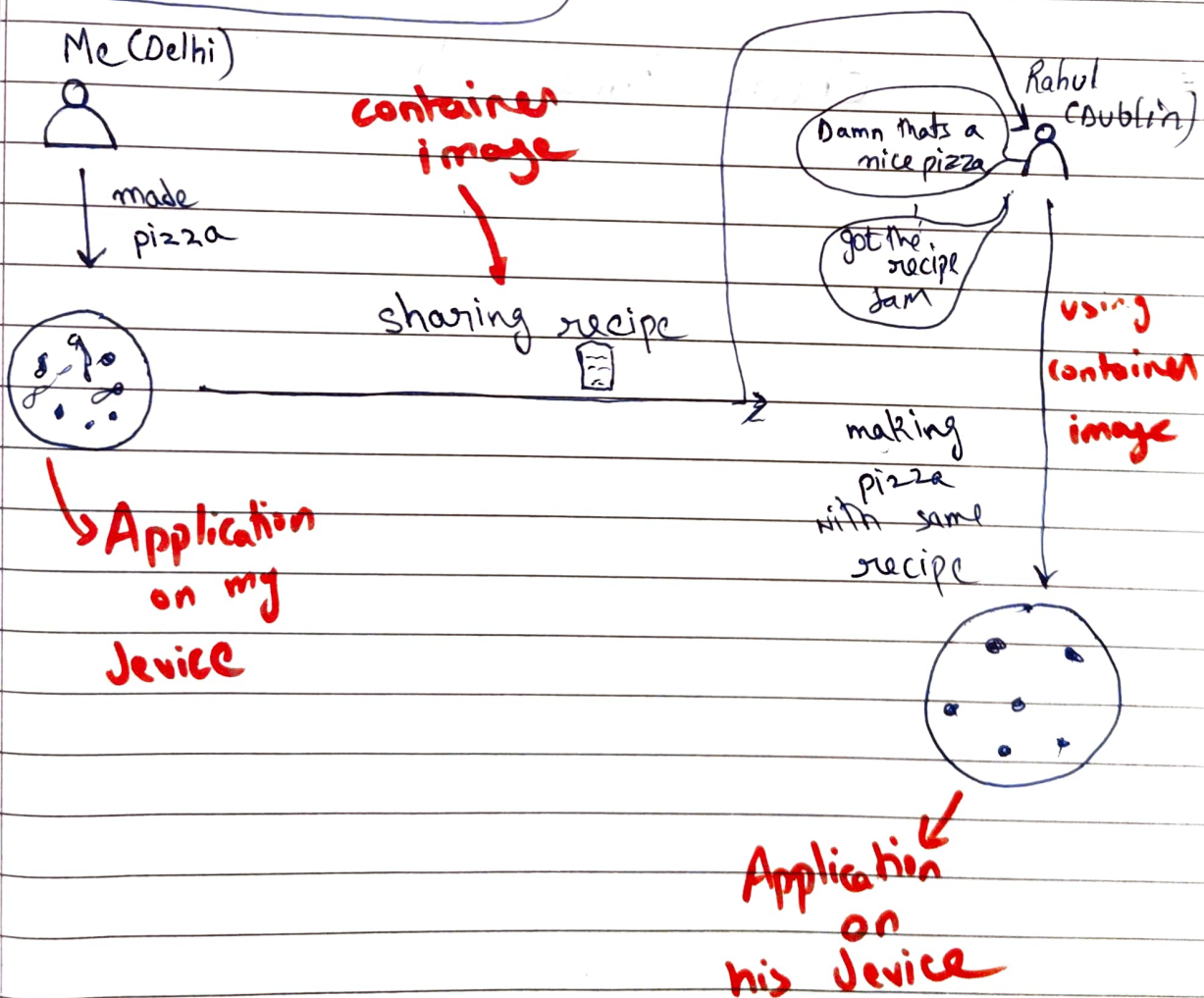
→ App on Jev

cont
cont

on do 1 by 1 which is quite time consuming.
So here orchestration comes to save manual labour.

These oper^m are done automatically using ~~s~~ orchestration engine like Docker swarm & also Kubernetes (it does much more than that)

Example of Docker



container image \Rightarrow instruction for running the app
container \Rightarrow instance of the image

