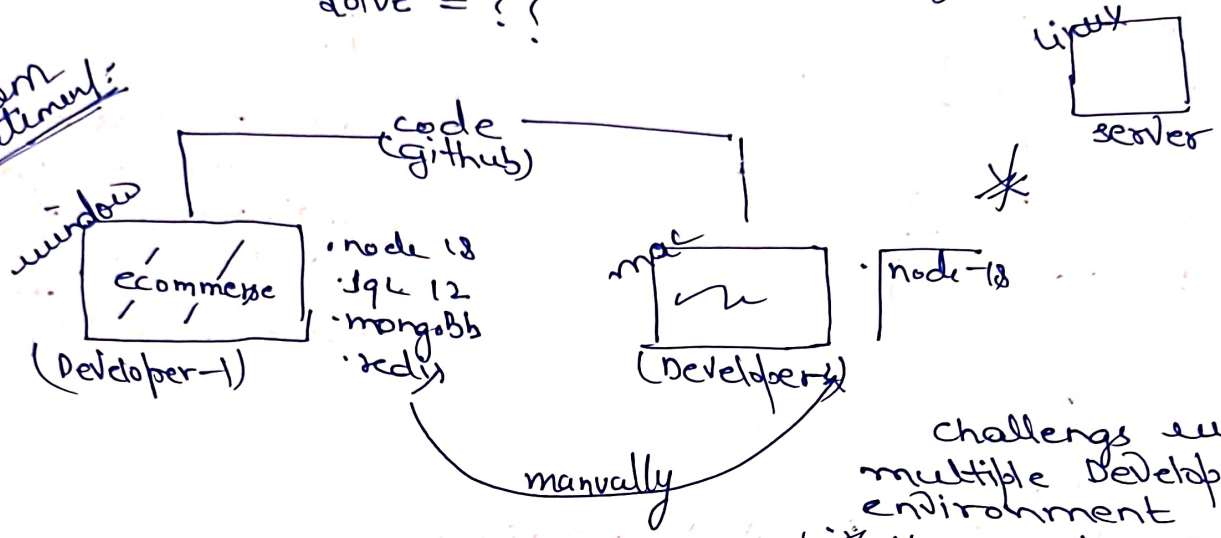
Docker

what problem it is trying to solve = ??

problem statement:



challenges with multiple Development environment

(i) it works on my machine

(ii) Dependency conflict
nodeV-18, V-20

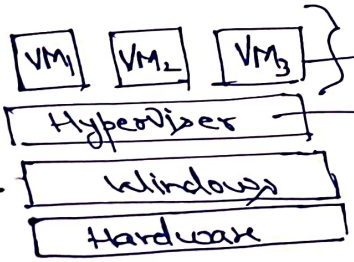
(iii) OS-specific issues
Eg: Window, mac, linux

possible solutions

(i) Virtualization
→ make same in all and by all developer as well as server (prodaction)

→ highly expensive.
→ very Ready.

full blown operating system



highly heavy

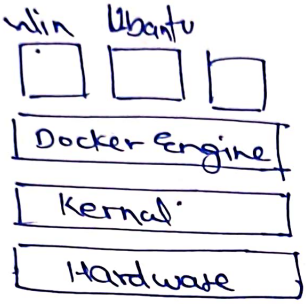
hardware level Virtualization

actual operating system

mediates access to real CPU, memory, storage and device.

VM gives container there own kernal.

(ii) Docker ation:



Note:
- Kernal talks directly to the hardware.
- Docker shares that Kernal for all its containers, that makes container much lighter and faster than VM.

Key Difference:

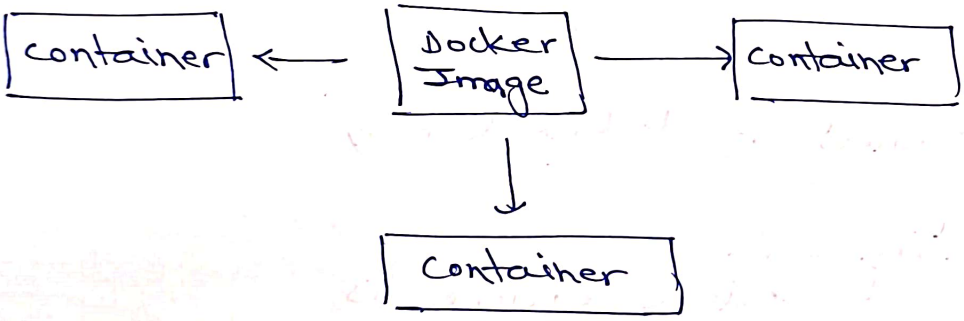
Docker uses isolated container while Virtualisation use VMs.

Installation of Docker

It means ?? → installing docker engine

docker container v/s image

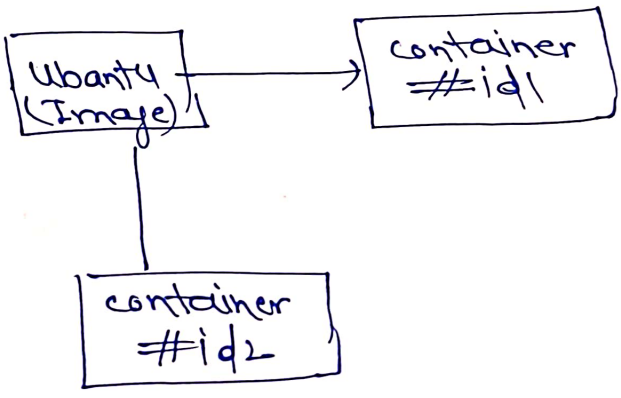
- image → doesn't have any data
- image → It is like a blueprint/template that contains everything to run a piece of software or machine
- container → isolated environment for running docker image



Concept:

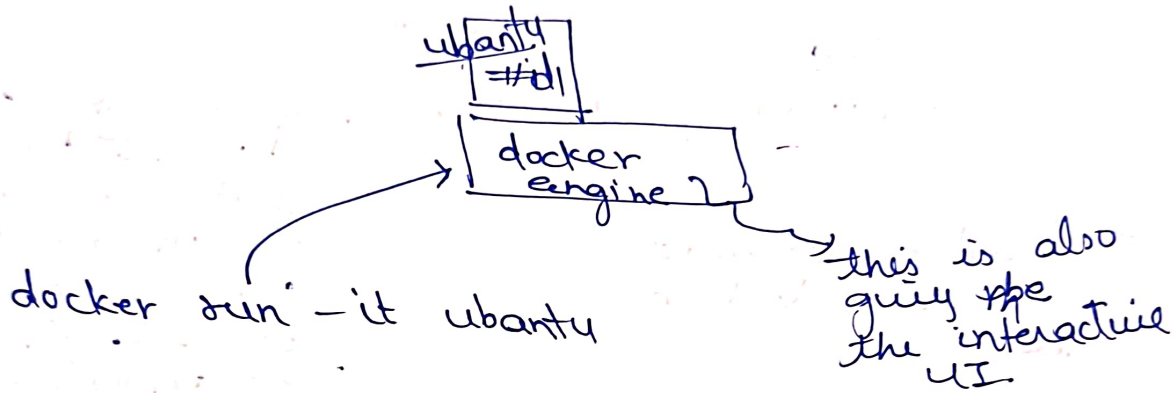
docker run -it ubanty → image

Visualise:-



Docker Engine and CLI

Note:
You do not have docker UI when working in a company (server in production)



Note:

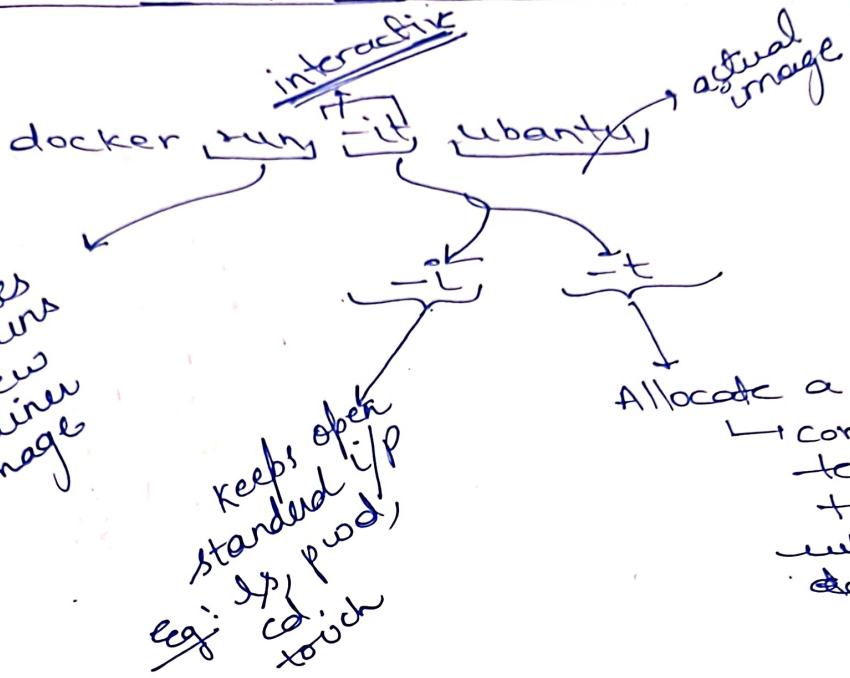
Error: cannot connect to docker demon

means = ?.?

↳ docker engine is not running

Concept:

Docker run CLI command



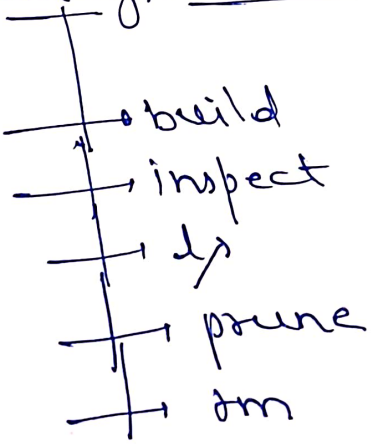
Note:

`docker pull <Image>`

pulls the image (manual pull)

Concept:

Managing Images with CLI



Docker Engine and CLI

Concept:

managing container with CLI

Note :

- opening the terminal automatically where doing docker run it by
  - entry point (Image) is bash.

check:

- inspect