Docker:

automate everything in devops industry. developed in GO Language. Basically, developed for Linux. Working after Windows 10. OS level Virtualization.

Its Free of cost that why preferred over cloud.

Requirements:

OOP Concepts

Git Commands

Limitations:

Cross OS not Possible (Both Host OS & Container OS should be of same i-e Linux).

disadvantages:

Not Suitable for heavy GUI. Maybe it will mature after some time.

Similar OS required (Both Host OS & Container OS)

Virtualization drawbacks:

Once VM Developed, it will hold memory & wont free it. Efficiency not good. Limited # of VMs is allowed only as per storage specs.

1. Wastage of Resources.

2. Every VM needs OS to Work.

3. HW Limitations because of HW Virtualization.

Containerization:

advanced version of virtualization.

Containers are lightweighted because of no OS of itself (It will download just additional files for UI).

Docker Daemon:

The Docker daemon is a service that runs on your host operating system.

Docker Use:

First Developer will make container.

Two ways:

Docker File (No Code) Sent to Docker Engine which create image then container(VM) will build from this image.

Docker Hub: Different Images pushed by diff: persons, Community Service Platform. Just for Enterprise Level.

We will directly get image from Docker Hub. Its most probable, we find it.

When u run the image, it becomes container.

Changes always done in container.

Docker Terms:

Docker Engine/Daemon:

Docker Client: CLI where we write commands. i.e., Docker container start myContainer. will sent to Docker Engine.

Docker Host: System where we install the Docker.

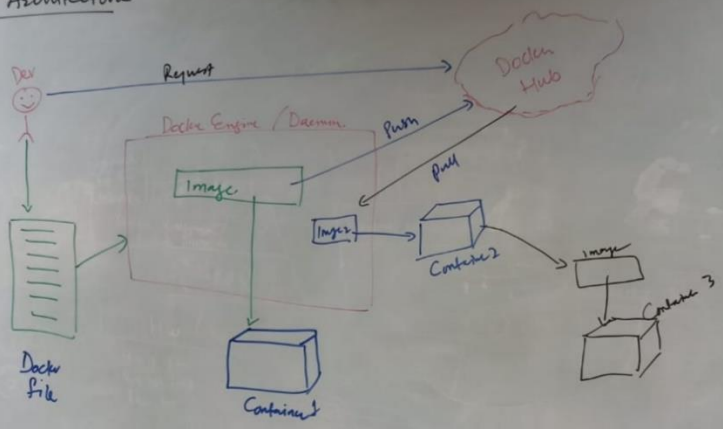
Docker Hub/Registry: Public Registry(), Private Registry(Paid)

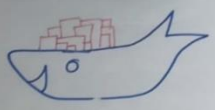
Image. 3 types: Docker File, Pull, Running Container

Container: a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Docker Compose: Communication between Multiple Daemons

Docker Architecture



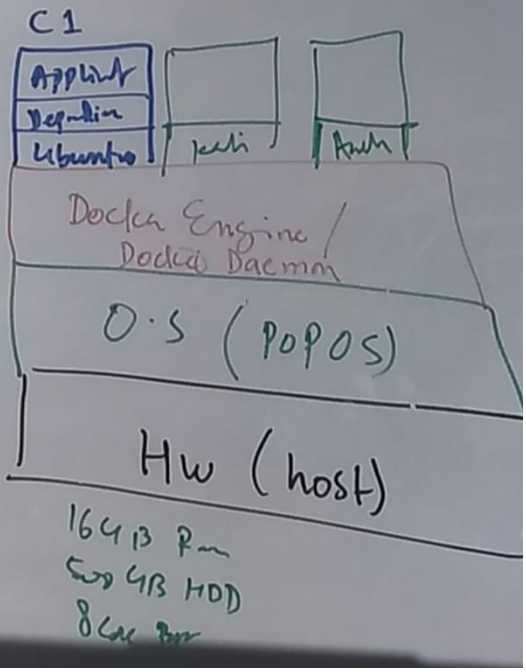


1. Docker Engine / Daemon
2. Docker client
3. Docker host
4. Docker hub / Registry
5. Image
6. Container
7. Docker Compose

Docker container start mycontainer

Containerization

-> Containers (like VMs)

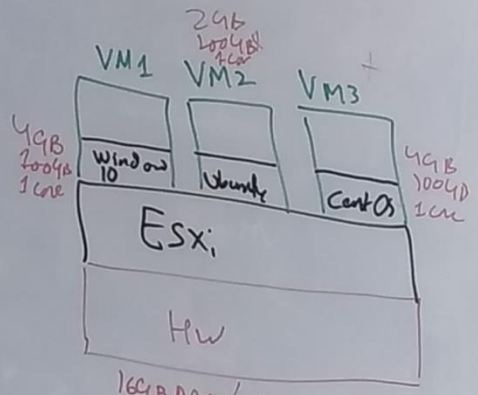


Virtualization

Virtualization

→ Software (Hypervisor)

- Type 1
- Type 2



8GB

VM4 ← 8GB

16GB RAM / 10GB = 6 VM

500GB HDD / 200GB = 2 VM

8 Core Processor / 2 Core = 5 VM