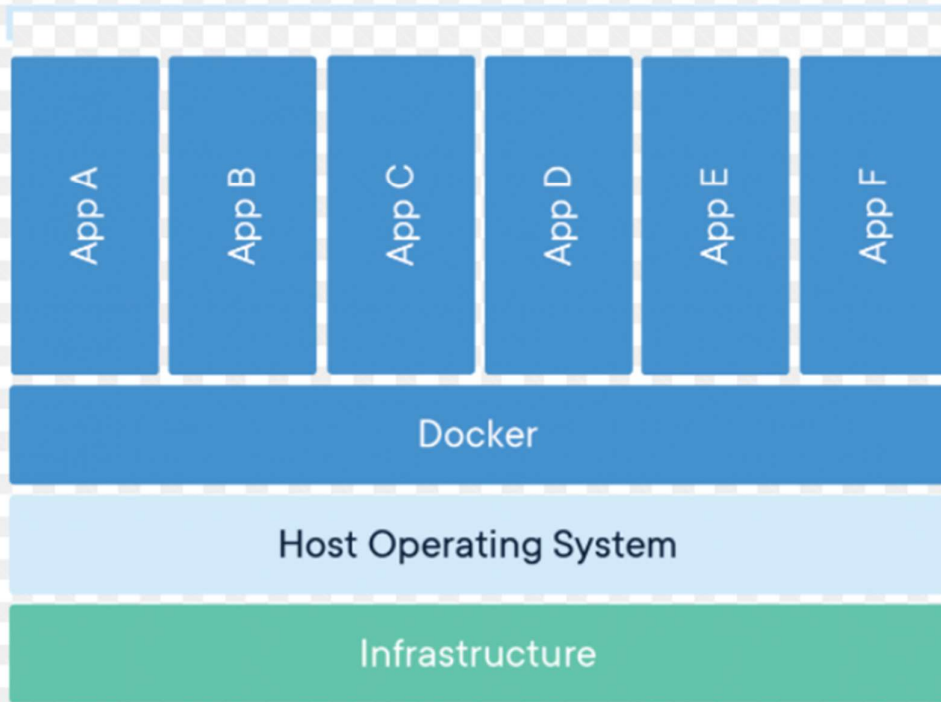


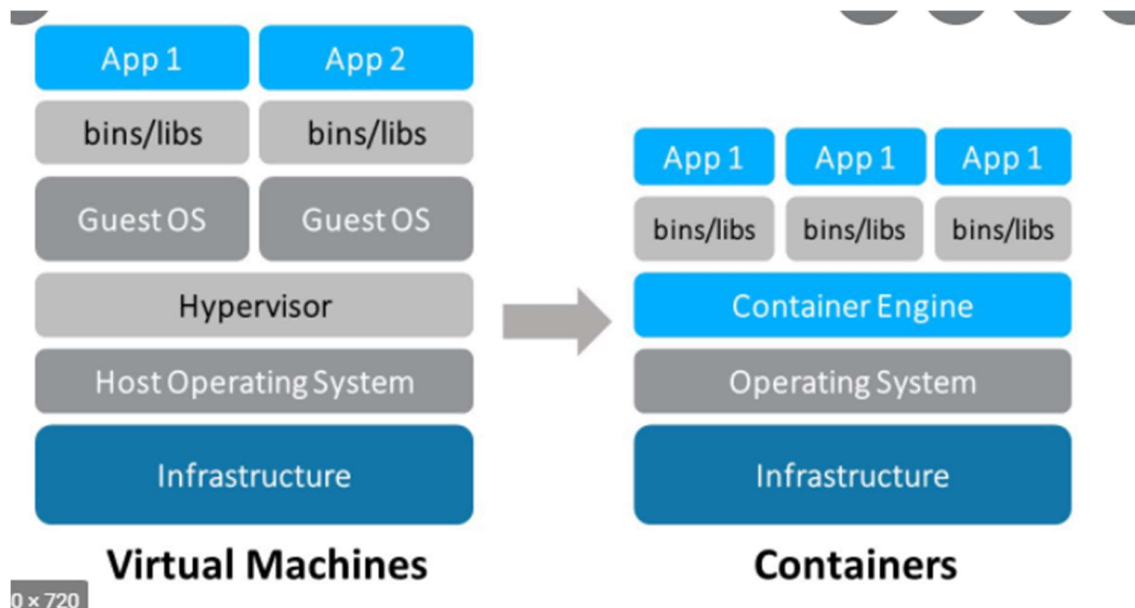
Docker

- ➔ Docker is an open-source centralized platform designed to create, deploy and run applications.
- ➔ Docker uses container on the host O.S to run applications it allows applications to use the same linux kernel as a system on the host compute, rather than creating a whole virtual O.S.
- ➔ We can install Docker on any O.S but Docker engine runs natively on linux distribution
- ➔ Docker written in go language.
- ➔ Docker is a tool that performs os level virtualization, also known as Containerization.
- ➔ Before Docker, many users faces the problem that a particular code is running in the developer's system but not in the users System.
- ➔ Docker was first release in March 2013, it is developed by Solomon hykes and Sebastian pahl.
- ➔ Docker is a set of platform as a service that uses O.S level virtualization whereas VMware uses Hardware Virtualization.

Containerized Applications



Difference between virtual machines and containers



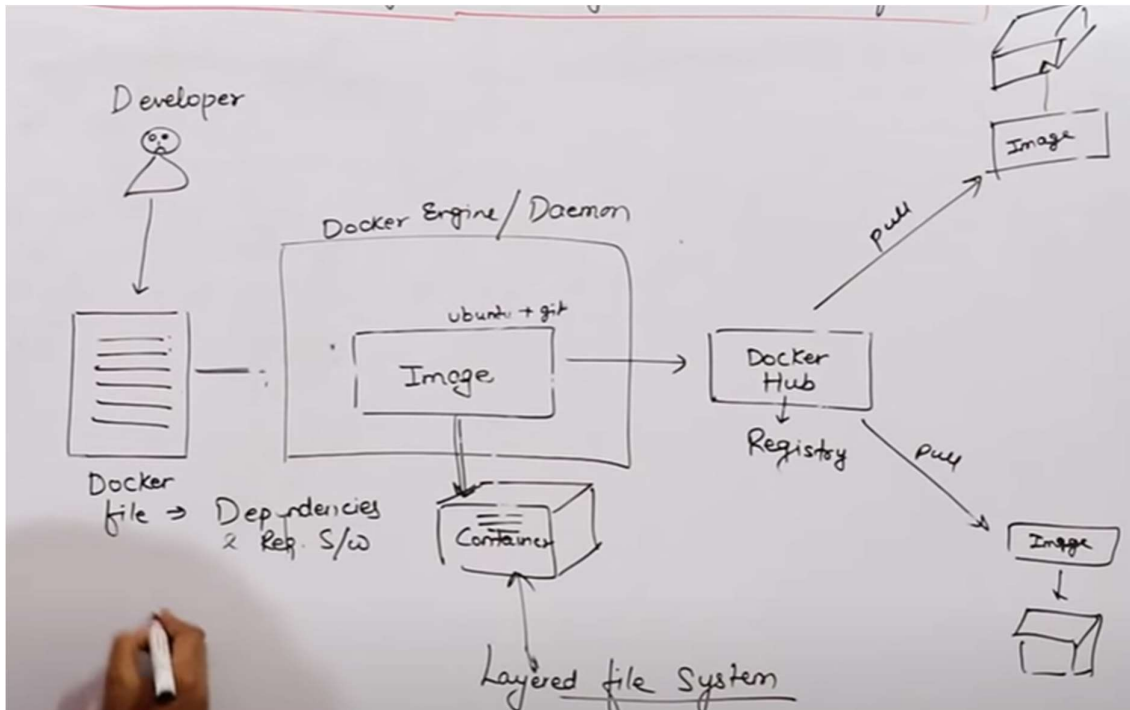
Advantages of Docker

- ➔ *No pre-allocation of Ram*
- ➔ *CI Efficiency -> Docker enables you to build a container image and use that same image across every step of the deployment process.*
- ➔ *Less Cost*
- ➔ *It can run on physical Hardware or Virtual hardware on cloud.*
- ➔ *You can reuse the image.*
- ➔ *It take very less time to create Containers.*

Disadvantages of Docker

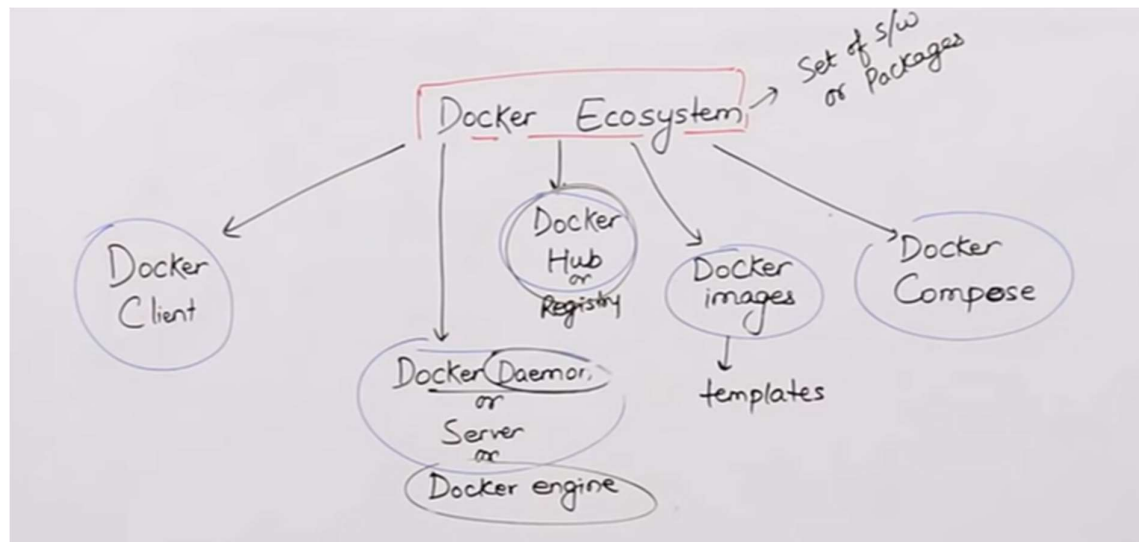
- ➔ *Docker is not a good solution for application that requires rich GUI.*
- ➔ *Difficult to manage large amount of containers.*
- ➔ *Docker does not provide Cross – platform compatibility means if an application is designed to run in a Docker.
Container on windows, then it can't run on Linux or vice-versa.*
- ➔ *Docker is suitable when the development O.S and testing O.S are same if the O.S is different, we should use Vm.*
- ➔ *No solution for data recovery and backup.*

Architecture of Docker



→ Container is a layer file system.

Docker is a Ecosystem



COMPONENTS OF DOCKER

Docker Daemon =>

- ➔ *Docker daemon runs on the host O.S.*
- ➔ *It is responsible for running containers to, manage Docker services.*
- ➔ *Docker Daemon can communicate with other daemons.*

Docker Client =>

- ➔ *Docker users can interact with Docker daemon through a Client. (like CLI)*
- ➔ *Docker client uses commands line and Rest API to communicate with the Docker daemon.*
- ➔ *When a client runs any server command on the Docker client terminal, the client terminal sends these Docker commands to the Docker daemon.*
- ➔ *It is possible for the Docker client to communicate with more than one daemon*

Docker Host =>

- ➔ *Docker host is used to provide an environment to execute and run applications, It contains the docker daemon, images, containers, networks and storages.*

Docker Hub/Registry

➔ *Docker registry manages and stores the Docker images.*

➔ *There are two types of registries in the Docker.*

1 => Public Registry = *Public registry is also called as Docker hub.*

2 => Private Registry = *It is used to share the images within the enterprise or company employee.*

Docker Images :-

➔ *Docker image are the read only binary templates used to create docker containers.*

Or

Single file with all dependencies and configuration required to run a program.

Ways to create an Images

1= Take image from Docker hub

2= Create image from Docker file.

3= Create image from existing Docker Containers

Docker Container

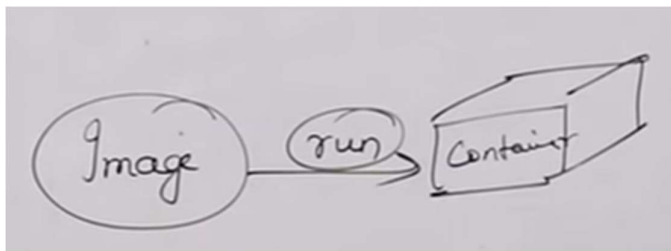
→ Container hold the entire packages that is needed to run the application

Or

In other words, we can say that, the image is a template and the container is a copy of that template.

→ Container is like a Virtual Machine, because containers helps you to run your application.

→ Images becomes container when they run on Docker engines



→ We can't do modify in image only container can modify your image.

