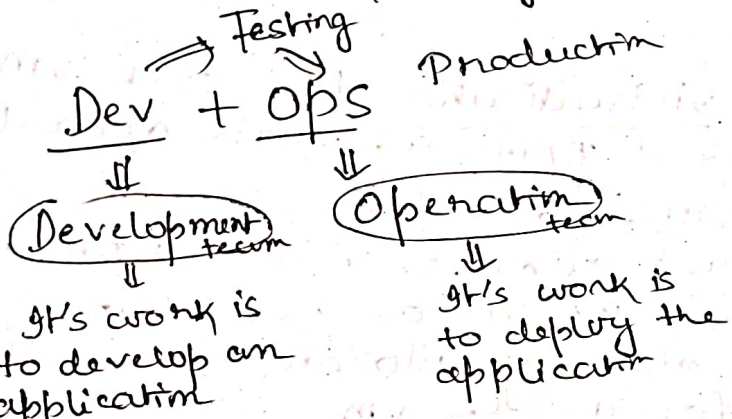
Docker

Container is like a virtual machine

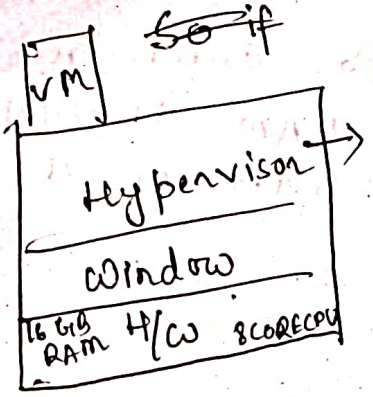
Docker is a tool which create this virtual machine

↳ It is Docker engine



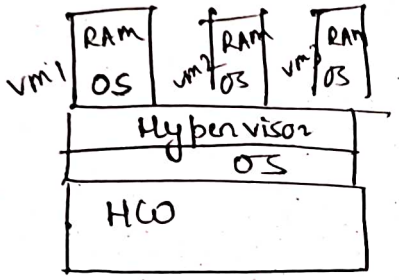
Suppose development team develop an application and sends it to testing team. Now testing team finds that your application is not working (It may be because some dependencies are not present) (It may be version error (version incompatible) To run an application, we need so many supported files in backend. Sometimes when your internet is on, it keeps downloading the file in backend.

But sometimes what happened, this software you can't install separately. It is already present in your OS. If we share OS, software with the application, then it will work. But we don't share our OS in real time.



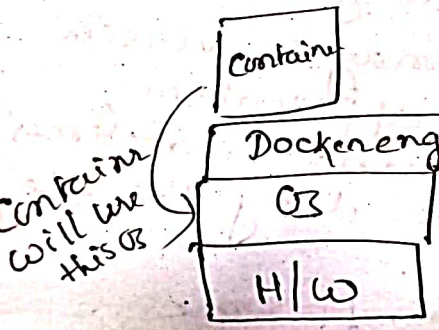
Using hypervisor, we have created the environment. Then we run the app on VM. In this VM, we have s/w, app, OS. Now we can share the VM with testing team. We create image and share it. Testing team will run this image on its VM and run this app. Now they will share this with Operation team. Testing team can add something and then it will send it to Operation team

Docker is a advance version of virtualization. Docker main task is containerization. So containerization is advance version of virtualization



In virtualization, for each VM, we need RAM, which is actually allocated from host machine. We cannot allocate more RAM, we use the VM or not, but if we already allocate, we cant use it for another VM

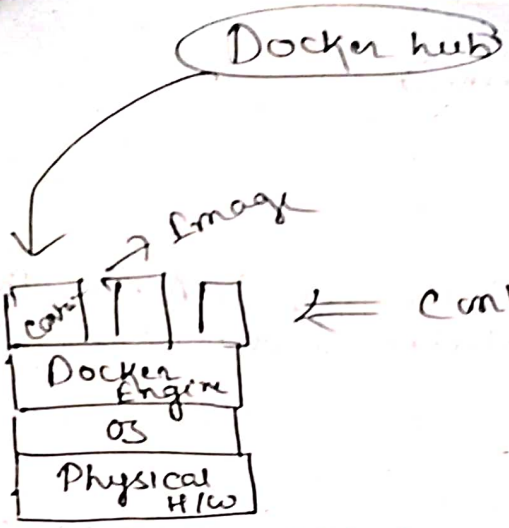
This problem is solved by Docker



→ similar hypervisor

If we use VM, we have to install OS. Then we create app.

But container does not have its own OS. It uses host machine's OS. In case of virtualization we preallocate OS and RAM. But for container if for running app, it needs 4 GiB RAM, it will use this. If for running app, it needs 8 GiB RAM, it will use this



If we need to run ubuntu on container, it will go to docker hub and bring ubuntu and run on container. It will not install

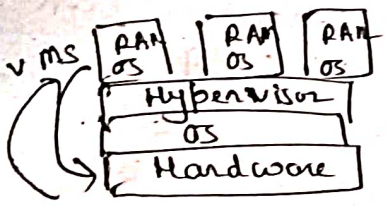
The development team will create an image of the container and send it to testing team. Testing team will also run it on container and send it to operation team.

* Container is similar to VM, but it will provide the dependencies

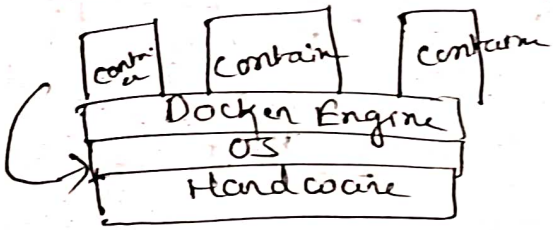
Advantages of Docker

① No pre-allocation of RAM

② VMware



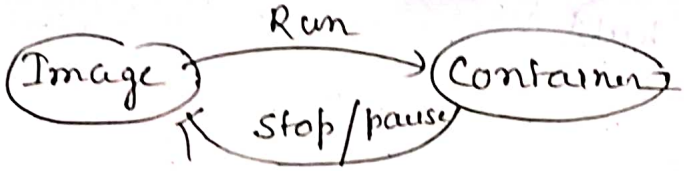
Containerization



② Continuous integration Efficiency :- Docker enables you to build a container image and use that same image across every step of ~~development~~ & deployment process.

{ The development team will create the image and send it to testing team. testing team will again send it

When image runs, it will be container. When you will send, you will send the image

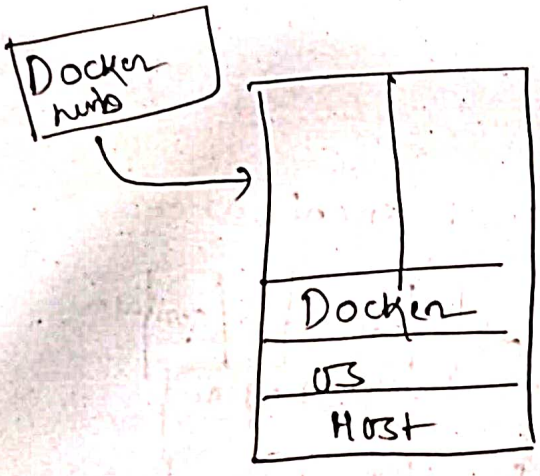


③ less cost :-
When coe create instances in AWS, that is also expensive than container. coe don't need any resources.

④ It is lightweight :-
It is using very feco resources of your laptop

⑤ It can run on physical H/w / virtual H/w or on cloud.
{ It can run on anywhere { windows or linux }

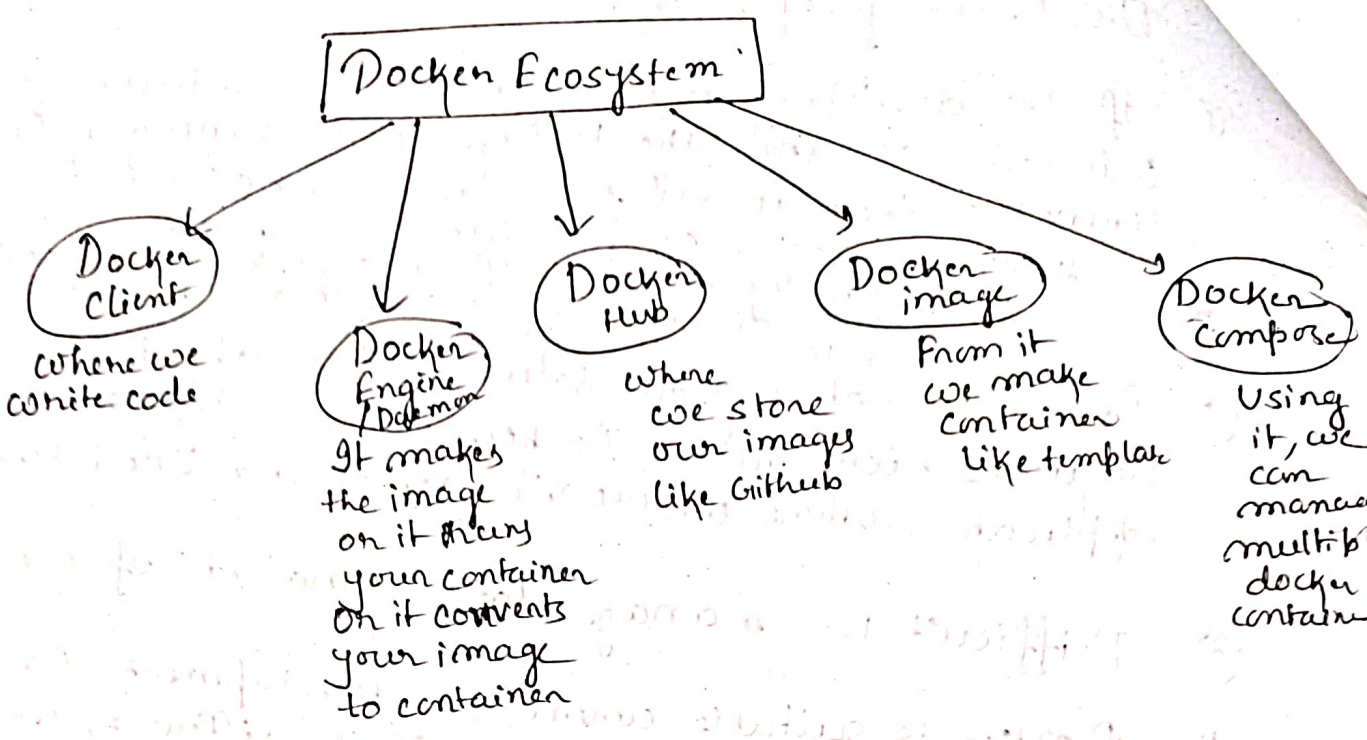
⑥ You can re-use



From Docker hub, I will bring image of ubuntu and I will bring that image to docker. Docker engine will store that image. One copy of this image will be used to make container. from next time container can use from docker engine

⑦ It takes very less time to create contain.

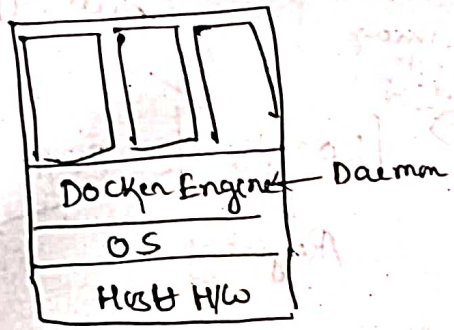
23:57



Components of Docker

① Docker Daemon / Docker Engine

* Docker Daemon runs on host OS.
* It is responsible for running container to manages docker service.
* Docker daemon can communicate with other daemons.



② Docker Client :-

* Docker users can interact with docker daemon through a client. (CLI)
* Docker client uses commands (CLI) and Rest API to communicate with docker daemon.
* When a client runs any server command on the docker client terminal, the client terminal sends these docker commands to docker daemon.
* It is possible for docker client to communicate with more than one daemon.
* It is possible for 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 :- The images are open for public. Public registry is called as docker hub.

② Private registry :- It is used to share images within the enterprise.

Docker images

Docker images are the read only binary templates used to create docker containers.

or

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

↓ contains

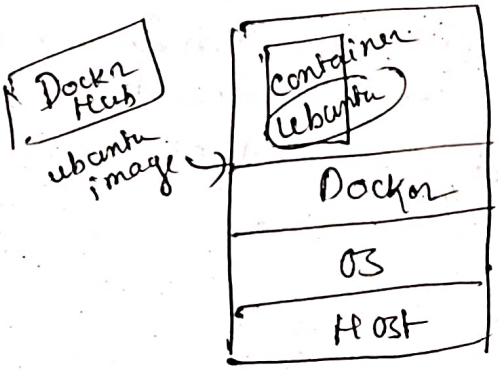
Ways to create an images

① Take image from docker hub

② Create image from docker file

③ Create image from existing docker container.

①



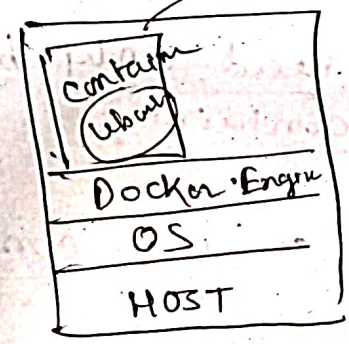
②

Docker file

Everything (dependencies) are written

When we run this docker file through client, Docker engine will create the image. From this image, it will create container

③



we can create image of this existing container.

If in this container there are 100 apps then if the created image is shared with anyone, s/he will not only get ubuntu, but also 100 softwares

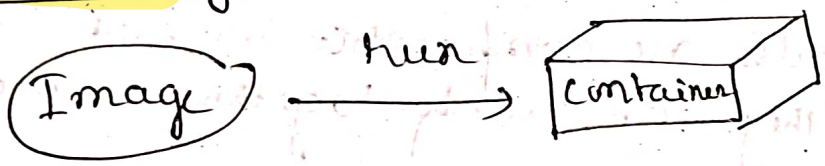
Docker container

* Container hold the entire package that is needed to run the application

or

We can say that, the image is a template and the container is a copy of that template.

* Container is like a virtual machine.
* Images become container when they run on docker engine



We can not change / modify images.
We can change on container.