TECHNICAL ⇌ GUFTGU

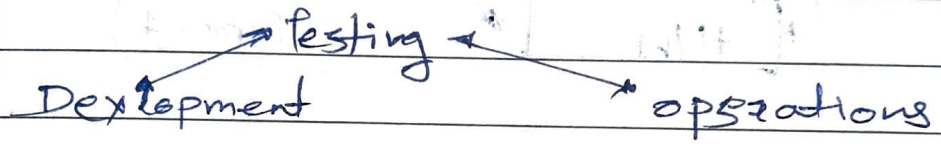
DOCKER

PAGE NO. 1
DATE : 26/09/24

→ Lecture - 1 ←

Container is like a V.M
Docker is a product of Docker engine

→ Need of Devops ←



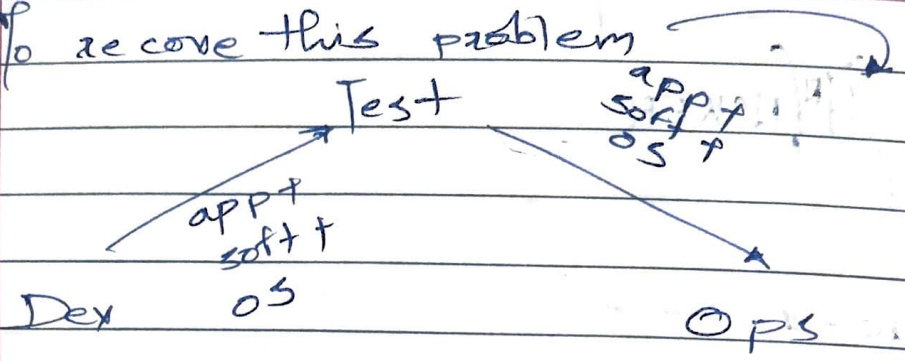
dependencies.

Development → develops app which needs 4 software of specific versions.

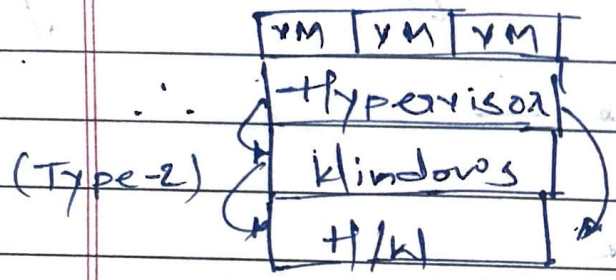
Testing & ops team should have same 4 software of specific versions to move app test & operate.

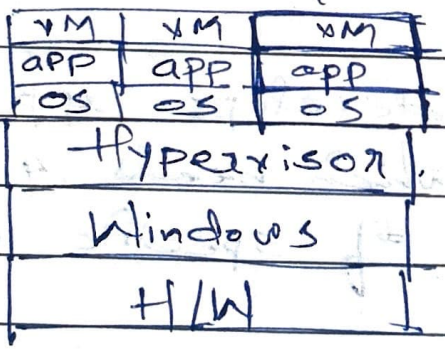
PAGE NO.
DATE :

To resolve this problem



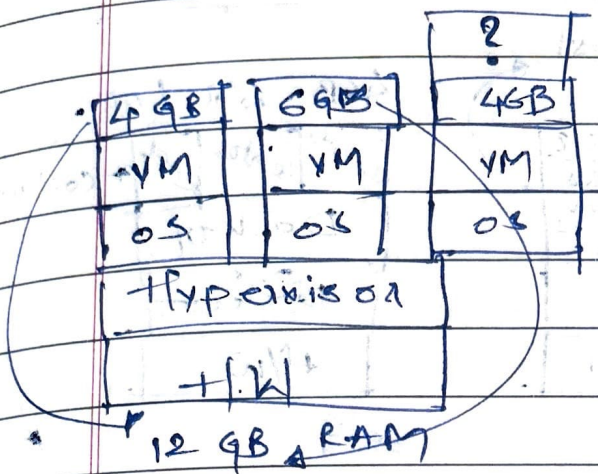
But OS can't be shared.





Now you can share whole VM to anyone by creating an image

Docker is like a advance version of virtualisation.



→ Not possible as not enough resources are available.

• Resources : RAM, CPU, storage etc.

In VM we have to first install ~~VM~~ OS.

→ → →

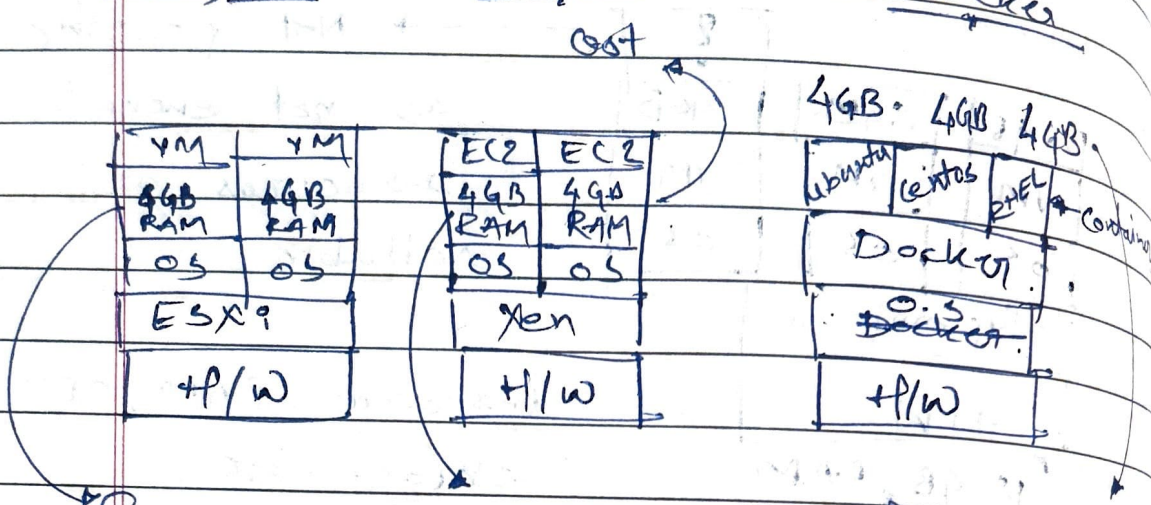
Containers
Docker Engine → like hypervisor.
O.S
Hardware

Containers don't have its own O.S. (it has small O.S) and uses O.S of Host

Containers have OS but with very less size ∴ it is said it is negligible.

PAGE NO.
DATE :

VMware → EC2 → Docker



Resources are permanently allotted.

Using Resources Only when needed

Docker hub → Container with all dependency

Convert to image

Send to testing team

→ Lecture 02 ←

- Docker is an open-source centralized platform designed to create, deploy and run applications.

- Docker uses container on the host os to run applications. It allows applications to use the same linux kernel as a system on the host computer, rather than creating a whole virtual o.s.

- We can install docker on any o.s but Docker engine runs natively on linux distributions.

- Docker is written in 'go' language

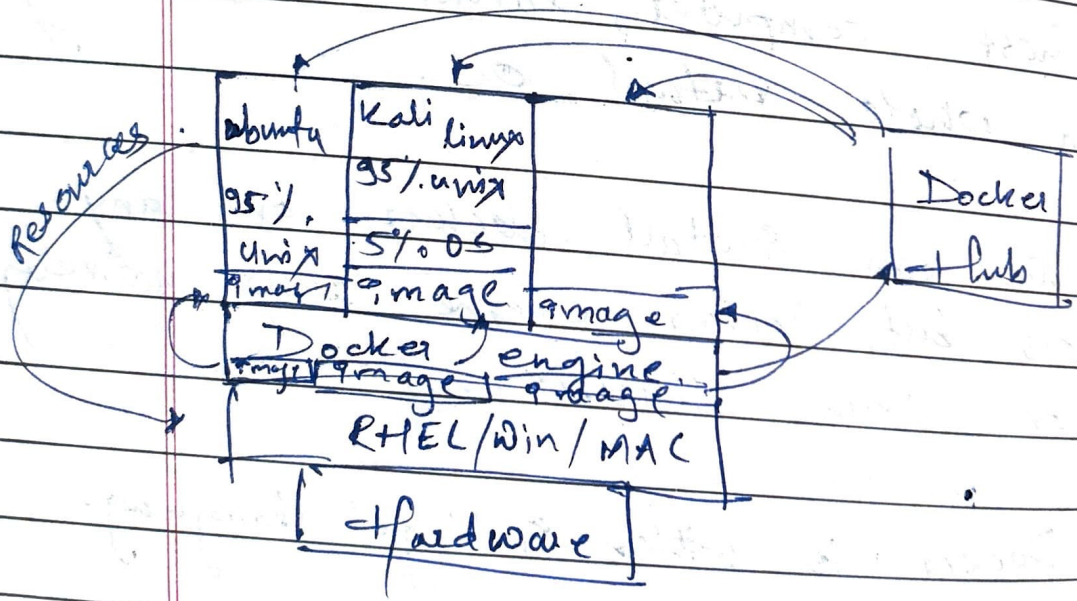
- Docker is a tool that performs os level virtulisation, also known as Containerization.

- Before docker, many users faced the problem that a particular code is running in the developers system but not in the Users system

PAGE NO.
DATE :

- Docker was first release in March 2013. It is developed by Solomon hykes and Sebastian pahl.

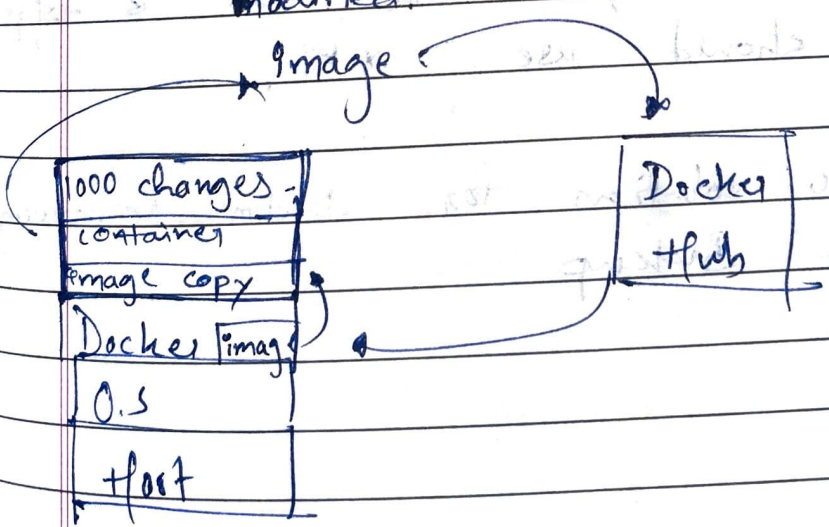
- Docker is a set of platform as a service that uses O-S level virtulisation whereas VM ware uses H/W level virtulisation.



Lecture 03

Advantages of Docker

- No pre-allocation of R+M
- 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 is light in weight
- It can run on physical H/W / virtual H/W or on cloud
- You can re-use the image
- It took very less time to create container

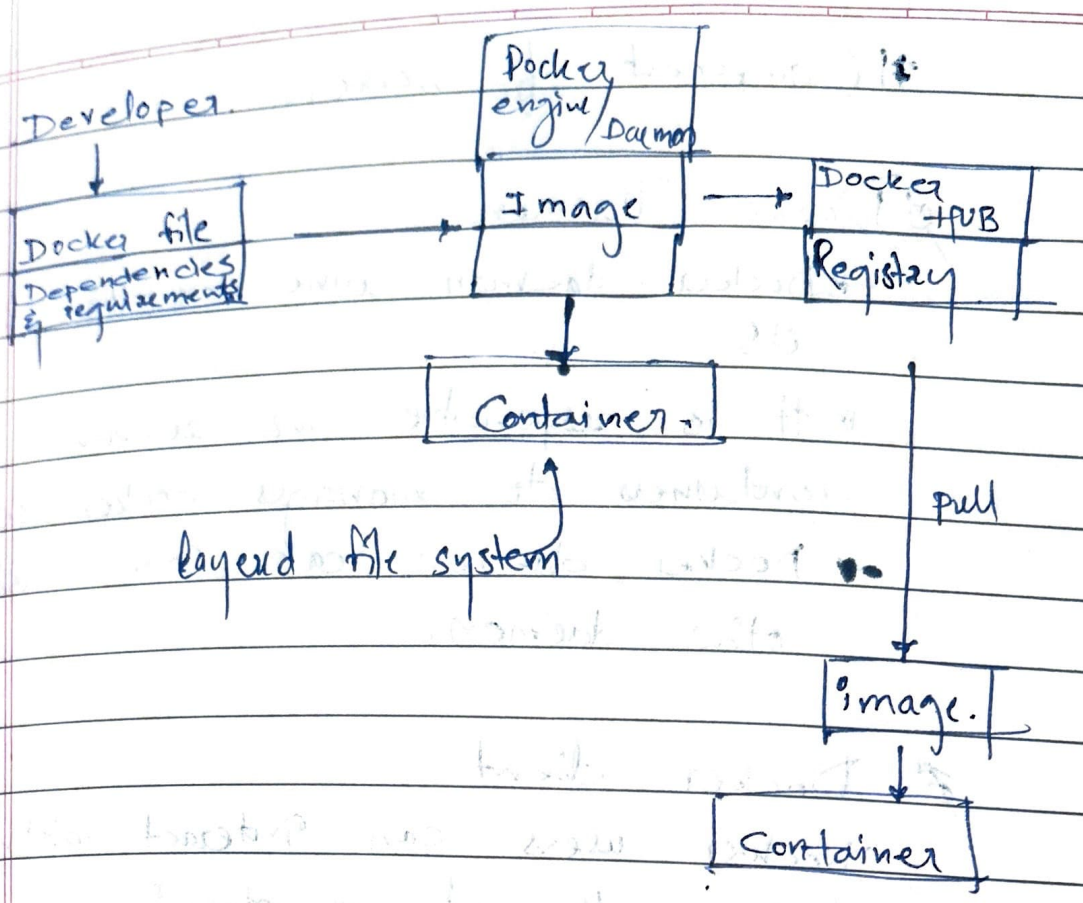


Changes can be done in containers not on images.

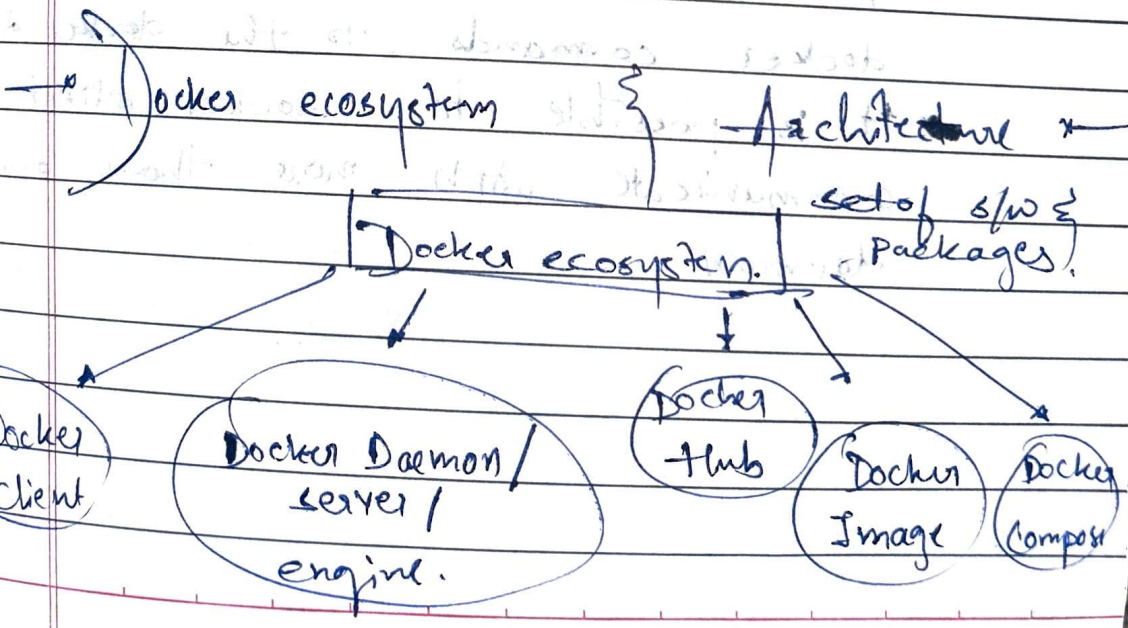
PAGE NO.
DATE :

Disadvantages of Docker

- Docker is not a good solution for applications that requires rich GUI
- Difficult to manage large amount of containers.
- Docker does not provide cross-platform compatability means if an application is designed to run in a docker container on window, 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 & Backup.



layered file system = works / downloads in sequence as written in Container file.



PAGE NO.
DATE :

Components of Docker.

Docker Daemon

- Docker daemon runs on the host OS
- It is responsible for running containers to manages docker services.
- Docker daemon can communicate with other daemon.

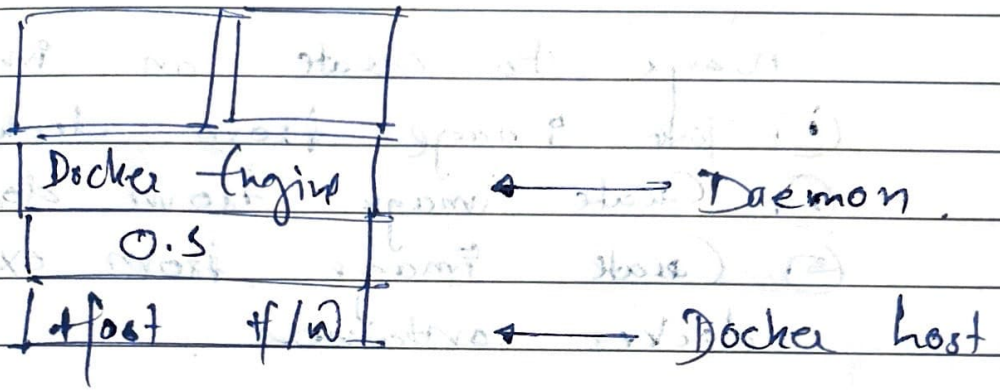
Docker client

Docker users can interact with docker through a client

- Docker client uses comands 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 comands to the docker 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: → Also called as docker hub
2. Private Registry: It is used to share images within the enterprise.

PAGE NO.
DATE :

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./container

Ways to create an image
1. Take image from docker hub
2. Create image from docker file
3. Create image from existing docker containers.

Docker Container.

→ Containes hold the entire package that is needed to run the application.

or

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

→ Containes is like a virtual Machine.

→ Images become Containes when they run on docker engine.

PAGE NO.
DATE : 27/05/24

Lecture NO. 04

Basic commands in docker.

To see all images present in your l/m
docker images

To find out images in docker hub
docker search jenkins → image name

To download images from d.h to l/m
docker pull jenkins

To give name to container.
docker run -i -t --name bhupinder ubuntu /bin/bash

- Create + start
- -i → interactive mode
- -t → terminal
- --name → if not given container will take default name
- ubuntu → run commands.
- /bin/bash

To check service start or not
service docker status.

To start container.
docker start bhupinder.

To go inside container
docker attach bhupinder → Cont. name.

To see all containers.
docker ps -a → all

To see only running containes.
docker ps → process status

To stop container.
docker stop bhupinder.

To delete container
docker rm bhupinder.

To install docker.
yum install -y docker.

PAGE NO.
DATE :

→ lab –

yum install -y docker
which docker.
docker -v # docker --version

service docker status.
docker info

service docker start
service docker status
docker info

docker images.
docker ps.
docker ps -a

go to google hub.docker.com
search for ubuntu.
#
docker run -it ubuntu /bin/bash.
docker # exit # cat /etc/os-release
docker ps -a
docker images.
docker run -it ubuntu /bin/bash
docker exit
docker ps
docker ps -a

docker pull jenkins.
↑ Name as mentioned on docker hub

docker search ubuntu.

docker run -it --name bhupinder ubuntu /bin/bash.
docker ps -a # exit

docker start bhupinder
docker ps
docker attach bhupinder.
exit

docker start bhupinder.
docker stop bhupinder.
docker rm bhupinder.

→ rm needs stoping of container
+ exit commands automatically stops container.
# docker

Location of podman pull image_name & containers in local system ???

PAGE NO.
DATE :

→ Lecture no. 05 ←

- If want to see difference between the base image & changes on it then;

docker diff blupi container update image
Container name

O/P Options:
C → change
A → Append Addition
D → deletion

C	/root
A	/root/.bash_history
C	/tmp
A	/tmp/myfile
D	Deleted data.

- Creating image out of container.

Docker commit newcontainer update image
Container name. image name

docker images.

Lab →

yum install docker -y
service docker start
docker info

docker run --name blupicontainer -it ubuntu /bin/bash
ls
cd /tmp
touch blupifile
ls
exit
docker diff blupicontainer.

docker commit blupicontainer updateimage
docker images.

docker run -it --name rajput cont. updateimage /bin/bash.
ls
cd /tmp
ls.
exit

PAGE NO.
DATE :

1) Dockerfile

Dockerfile is basically a text file it contains some set of instruction. — Automation of Docker image Creation.

Docker components.

FROM → for base image. — this comand must be on top of the dockerfile

RUN → To execute comands, it will create a layer in image

MAINTAINER → Author / owner / description.

COPY → Copy files from local system (docker VM) we need to provide source, destination (we can't download file from internet & any remote repo)

ADD → Similar to copy but it provides a feature to download files from internet, also we extract file at docker image side.

EXPOSE → To expose ports such as port 8080 for a tomcat, port 80 for nginx etc.

WORKDIR → To set working directory for a container.

CMD → Execute commands but during container creation.

ENTRYPOINT → Similar to CMD, but has higher priority over CMD, first commands will be executed by ENTRYPOINT only.

ENV → Environment variables.

ARG → ??

To create image out of docker file.
docker build -t myimg .

PAGE NO.
DATE :

lab

vi Dockerfile
FROM ubuntu
RUN echo "Hello" > /tmp/test

docker build -t name .

docker run -it --name tc name /bin/bash

cd /tmp
ls.
cat test.

vi Dockerfile
FROM ubuntu
WORKDIR /tmp
RUN echo "Hello" /tmp/file.
ENV myname bhupivelu.
Copy testfile /tmp
ADD test.tar.gz /tmp

echo $myname.