→ or Ansible / Puppet

Docker?

Docker is an open-source technology used mostly in

1] Developing
2] Shipping
3] Running applications.

↳ • Docker architecture and components
  • Containers and Registry
  • Docker swarm
  • Docker compose
  :

Docker introduction?

→

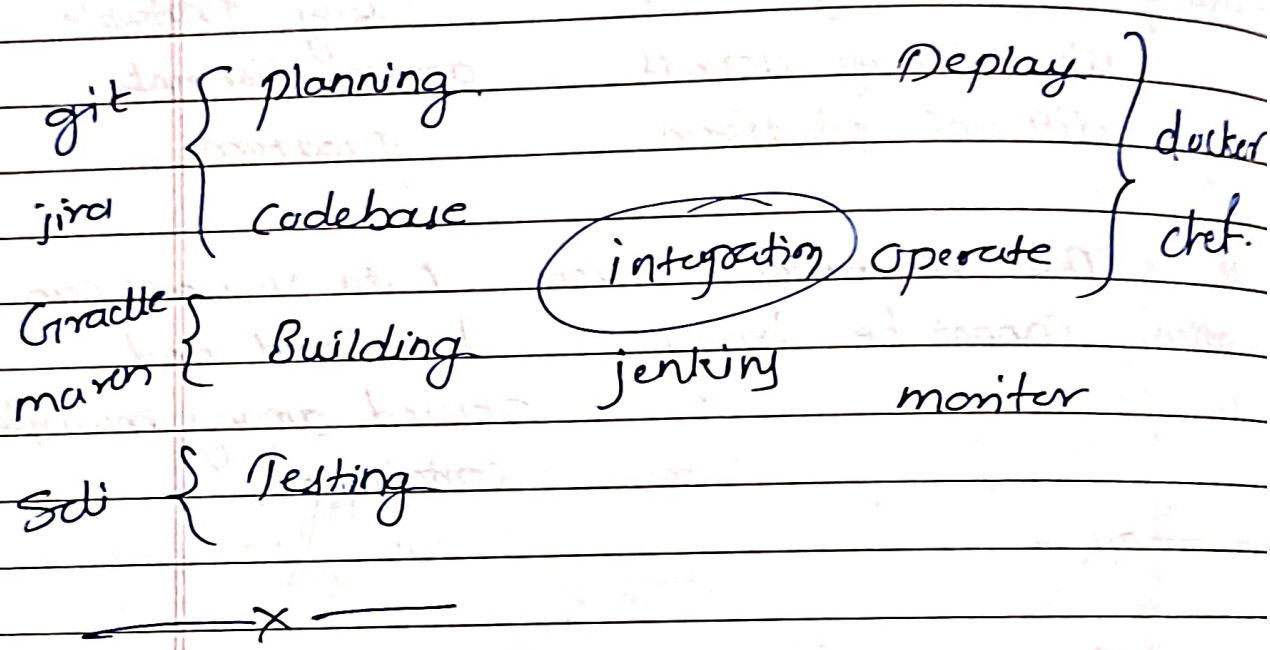
	~~Doc~~ VM	Docker
OS support	• occupies ~~colot~~ a lot of memory space	Docker containers occupies less memory space,
BootupTime	• long boot up time	Short
Performance:-	Running multiple VM leads to unstable performance	containers have a better performance as they hosted in a single Docker engine

cale	Difficult	Easy
fficiency	low	high
ortability	compatibility issues while porting across different platform	Easily portable across different platform.
ata olume	Data ~~allocation~~ volumes cannot be shared	Data volumes can be shared and reused among multiple containers.

Docker is a tool which is used to automate the deployment of application in lightweight containers so that application can work efficiently in different enviroments.

containers :- is an software package that consist of all the dependencies required to run an application.

- devops is a collabration between development and operation team which enables continous delivery of application and services to our end user.



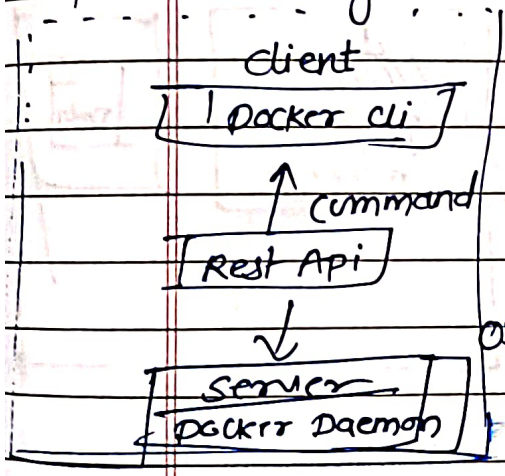
Container is a software package that consists of all dependencies required to run an application.

multiple container run on the same hardware → docker → High productivity

docker → maintains isolated application

docker → Quick and easy configuration

How docker works?



- Docker Engine or Docker is the base engine installed on your host machine to build and run containers using docker components and services.
- It uses a client-server architecture

- Docker client and server communicate using Rest API

classmate
Date
Page

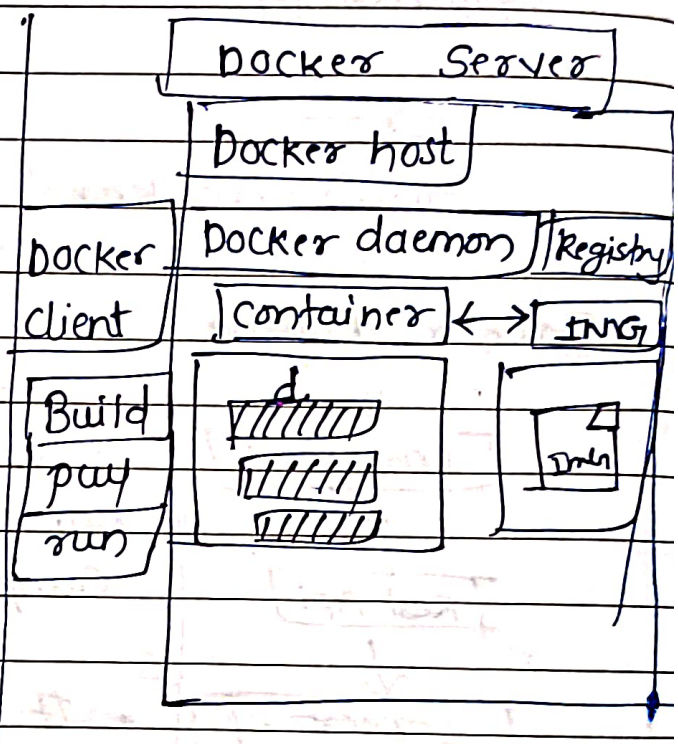
lesson 3 : Docker component

Docker client → Docke image → Docker → Docker
and server container Registry.

Docker client and Server

- a docker client is accessed from the terminal and a docker host runs the Docker daemon and registry

- A user can build docker images and run docker containers by passing commands from the docker client to docker server



Docker Image.

- Docker Img is a template with instruction which is used for creating Docker containers.

- Docker Img is built using a file called docker file.

Docker
Registry.

er Server
r host
daemon | Registry
ainer ↔ IMG

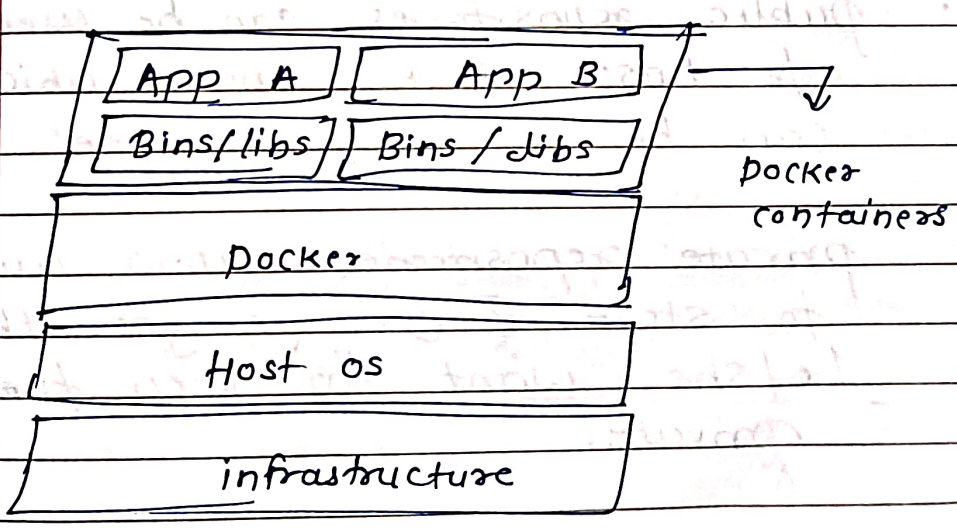
te with
for
.
ng a

- Docker image is stored in a docker hub or in a repository.
(like registy.hub.docker.com)

— x —

Docker containers.

- Docker container is a standalone, executable software package which includes application and their dependencies.



- numerous Docker container run on the same infrastructure and share operating system OS with their other container

- Here, each application runs in isolation.

✱ Docker Registry.

- Docker Registry is an open source server-side service used for hosting and distributing images.
- Docker also has its own default registry called Docker Hub.
- Here, images can be stored in either public or private repository.
- Public repositories can be used to host Docker images which can be used by everyone.
- private repositories allow users to store Docker images that he/she want to keep them private.
- pull and push are the commands used by users in order to interact with docker Registry.
- In order to build a container pull command is used to get a docker image from the Docker repository.

Docker pull <image>:<tag> : pulls an image from DTR.

- with push command a user can stored the docker image in a docker Registry.

- Docker ~~pull~~ push <image> : tag
  → pushes an image to DTR.

Summery :-

- A dockerfile creates a docker image using a build command.
- A docker image contains all the project's code.
- using a docker image, any user can run the code in order to create containers.
- once a docker image is built it's uploaded in a registry or a Docker Hub.
- From the docker hub, user can get the docker images and build a new container.

~~Lesson 4~~:

Containerazation :- It is the process of packaging an application and everything it needs (like code, libraries and setting) into one standard box called a container.

Before container, developers often said "It works on my computer but app didn't work on the same other computer. Container Solve this with environment consistency, Fast Scalibility and lightweight deployment across platform.

Docker :- Docker is an open-source tool that helps you create, run and manage containers.
It puts your app and everything. It needs into a containers, so the app run the same on any computer.

Why docker 1] consistency across Envir.
2] Faster Deployment
3] Lightweight and Efficient
4] Easy Scalibility.

Vm Vs Docker.

- Vm and docker both are used to run application in isolated enviro. but there architecture performance and use cases are very different.

- "A vm runs a full operating system along with virtual hardware. It needs more memory, more time to start and use more system resources.

- Docker runs the app using the host's os. It doesn't need a full os. So its light weight, start quickly and uses fewer system resources. so its faster and better for devops.

Docker file.

1) A docker file is simple text files.
2) It has list of instruction (like step) to build a docker image.
3) Each line tells docker what to do. (install software, copy files.

4) when you run docker build, Docker reads the dockerfile and creates an image from those instructions.

—x—

Command.

docker --version → Show version.
docker info → system-wide docker info
docker help → help for all command.
docker <command> --help → Specific command

Images :-

docker images → list all images.
docker Search <name> → Search image on dockerhub.
docker pull <image> → download img.

docker rmi -f <image> → remove img.
docker image prune → remove unused img.

classmate
Date
Page

containers :-

* docker ps :- list all running containers.
* docker ps -all :- list all containers.

* docker run <image> :- create + start container.
* docker run -it <img> bash :-
  Run in iterative mode.
* docker run -d <img> :- Run in background
* docker exec -it <container> bash
  Login to container
* docker logs <container> :- Show logs.
* docker Stop <cont> :- Stop contai-
* docker Start <contain> :- Start contai-
* docker restart <cont> :- Restart —11—
* docker rm <contai> :- removed Stop —11—
* docker rm -f <cont> :- remove forcefully
  docker container prune :- removed all cont.

—11—X——X—

Build Image

* docker build -t <name> <tag>
  ↳ Build Image from dockerFile.
* docker tag <img> <repo> :<tag>
  Tag to image
* docker push <repo> : <tag>
  push img to repo.
* docker commit <container> <img>
  create img from container.

networking

* docker network ls
  ↳ list network
* docker create network <name>
  ↳ create new network.
* docker network inspect <name>
  ↳ show network details.
* docker network rm <name>
  ↳ Remove network

—x—

Volumes (Data Storage)

* docker volume ls
  ↳ list volumes.
* docker volume create <name>
  ↳ create volume
* docker volume inspect <name>
  ↳ show volume details.
* docker volume rm <name>
  ↳ remove volume

* docker run -v <volume> :/path <img>
  mount volume in container.

— x —

clean up :-

* docker system prune
  ↳ Removed unused data.

* docker system df
  ↳ show disk usage

— y —

docker composed.

* docker compose up -d
  ↳ start all services in background

* docker compose down
  ↳ stop and remove containers.

* docker compose ps
  ↳ list composed containers.

* docker compose logs
  ↳ show log of all services.

* docker compose build.
  ↳ Rebuild services.

Docker File components.

Docker File components tell Docker how to build an img step by step Each component perform a specific task.

FROM :- For base image. This command must be on the top of the docker file.

RUN :- To execute command. it will create a layer in img.

MAINTAINER :- Author / owner / description

COPY :- Copy files from local system. we need to provide source, destination

ADD :- Similar to copy. it provides features to downloads files from internate, also we extract files at docker image side.

EXPOSE : To expose port such as port 8080 for tomcat, port 80 for nginx etc

WORKDIR :- To set working directory for a container

CMD :- execute command but during container creation.

ENTRYPOINT :- Similar to CMD, but has higher ~~priority~~ priority over CMD, first commands will be executed by entrypoint only.

ENV Enviroments Variables.