DOCKER
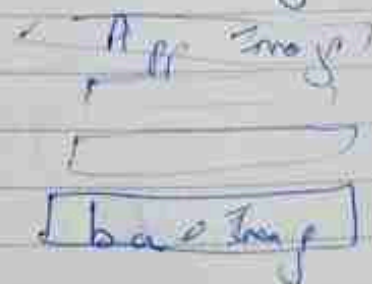
- Container: A way to package application with necessary dependencies and configuration.

- Portable artifact, easily shared and moved around

- Container stored in Docker Hub repository.

- Container has own isolated environment, packaged with all needed configuration, one command to install the app.

Container:

- Layers of Images, mostly linux base Image. app image on top

Image contains all configuration and dependencies. It is an artifact.

→ when we install docker, we get docker engine, docker cli, client, docker compose, docker machine and kitematic

Diff between Image and Container

Container is running env of for Image

Container is port binded : which means
it binds to the application
running inside of container.

Has its own virtual file system.

┌─────────────────────────┐
│   [Container]   │
│   │
│   Filesystem   env   │
│   (env │
│   vars) │
│   ▤   app   │
│   img.   │
└─────────────────────────┘

Pas 500:0

Commands
docker run -d redis
docker stop (id), docker ps, docker bat (id)
docker ps -a , docker image ver
docker run redis:4.0

HOST PORT   VS   CONTAINER PORT
To bind container port to host port

docker run -p 6000:6379
             ↓   ↓
            host   container port

For Troubleshooting
docker logs (id)

To change name of
To chng name of
docker run -d -p6000:6379 -- name redis-older redis:4.0

Through docker exec we can get the terminal
of running container

docker exec -it (id) /bin/bash

docker run - To create new container
start - To restart stopped container

* Docker Network

Docker Creates its own isolated network when container one running in.

docker network create mongo-network

docker run command	mongo-docker-compose yml
docker run -d \	version: '3'
--name mongodb \	service:
-p 27017 : 27017 \	mongodb:
-e mongoDB_INEr	image: mongo
--net mongo network \	port:
	27017
	environment:
	-- MONGO_Username=admin

» docker-compose -f mongo.yml up

* Docker file

* Blueprint for building images

```
FROM node              // it means install node
ENV MONGO-DB-USERNAME  // set env variable
RUN mkdir -p/home/app  // create /home/app folder
COPY ./home/app        // copy current folder file
                       //    to /home/app
CMD ["node", "server.js"]
                       // start off node server.js
```

Note: You can have multiple Run command
but CMD command is entry point.

* creating image.

docker build -t myapp:1.0 . or myapp:version.

To remove image, rmi
        container, rm

It means current directory

If we modify the dockerfile, we need to recreate/build
another image.

- Private Docker Registry Amazon ECR

docker login

* Image name in docker registry :-
  { registryDomain /image Name : tag }

In dockerhub   (inside public)
docker pull mongo:4.2   docker pull docker.io/library/mongo

To push to private registry, we use
docker tag ------

* Docker volume,
  Uses for data persistence

* folder in physical host file system is mounted
  into the virtual file system of Docker

* Changes done in host file system is changed on
  docker container

3 volume types

* docker run
  -v /home/mount/data : /var/lib/mysql/data

1) Host volume → _____Host_____   _____container_____

2) Anonymous volume — Nahi pata kaha save hoga

3) Named volume , docker run
   -v name : /var/lib/mysql/data

Docker Volume Locations

win — C:\ProgramData\ docker \ volume

Linux, mac — /var/lib/docker/volumes