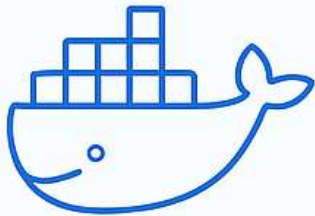


Docker Fundamentals

Build, Ship, Run Anywhere

WHAT IS DOCKER



Containerize your app
and run it anywhere

WHY DOCKER

PROBLEM

Apps run
differently in
each
environment



SOLUTION

Containers
ensure
consistent
execution



CORE COMPONENTS



DOCKER IMAGE

App blueprint



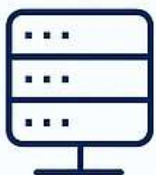
DOCKER CONTAINER

App instance



DOCKER ENGINE

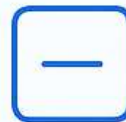
Container runtime



DOCKER REGISTRY

image storage

WORKFLOW



BUILD



PUSH



RUN

BENEFITS

- ✓ Consistency
- ✓ Isolation
- ✓ Portability
- ✓ Efficiency
- ✓ Scalability

Docker = Lightweight containerization that runs **anywhere**

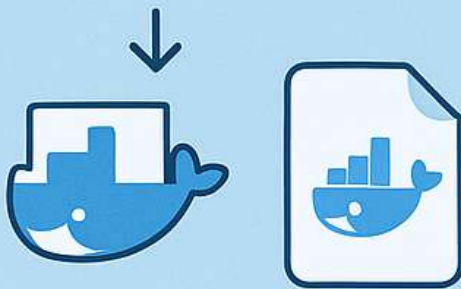
#Docker #Containerization

FROM CODE TO CONTAINER

HOW DOCKER WORKS



Developer
writes code



Builds image



Uploads image
to registry
(e.g., Docker Hub)



Pulls and runs
image in a
container

Build once, run everywhere



Consistency



Isolation



Portability



Scalability

Docker Introduction

What is Docker:

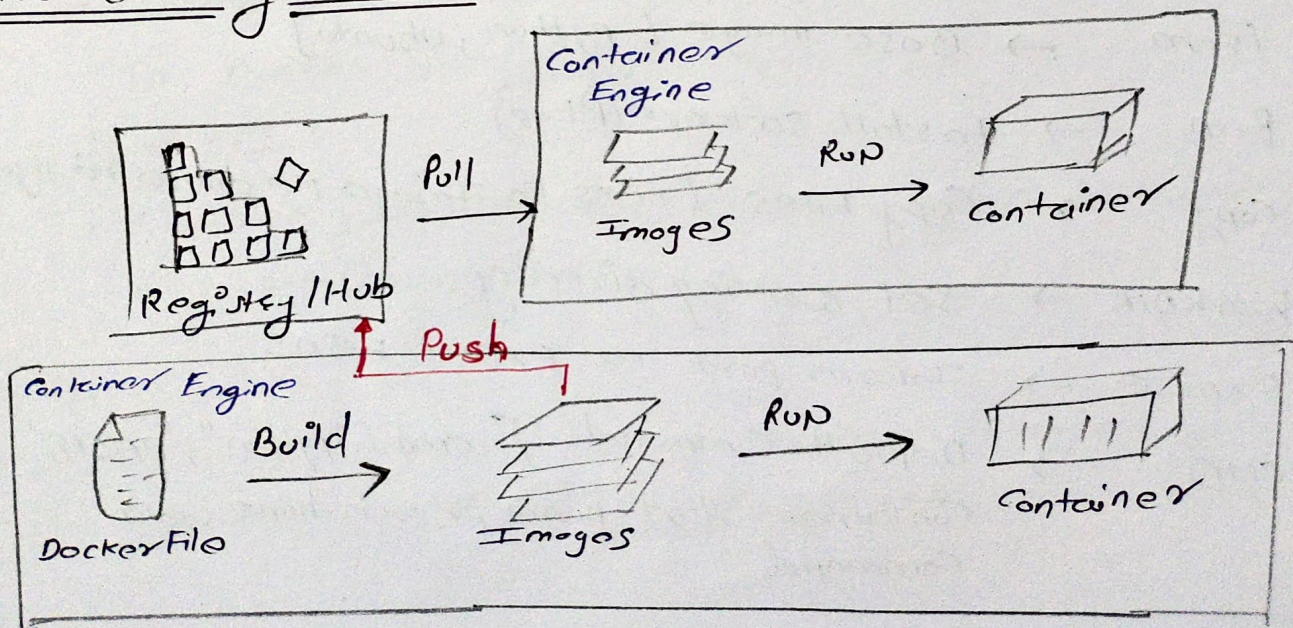
- Docker is a platform designed to help developers to build, share, and run container applications.
- Docker is a platform that lets you package, distribute, and run applications in isolated environment called containers.
- Think of a container as a lightweight, portable box that contains:
 - Your Application code
 - Libraries
 - Dependencies
 - Run time environment.

• Why do we need Docker?

• Problems:- Inconsistent environments - Apps behave differently in dev, test, prod

• Solⁿ:- Containers include everything [code, dependency] so run the same everywhere.

• How exactly Docker is used?



!:- Docker EK tool hai jisse Hum apna App ek Container ke ander pack krke kahi bhi some turike se chala sakte hain.
— laptop, server, ya cloud me.

Docker Image :-

- A Docker Image is a Read-only blueprint with everything (code, libraries, configs) needed to run an app.
- Image ek template hoti hai jisse container banta hai.

Lifecycle of Image.

1. Create → via docker build & Dockerfile
2. Store → locally or in a registry (Docker Hub)
3. Distribute → push / pull between machines.
4. Execute → Run Containers from image.

Docker File

A Docker File is a text File with Instructions to build an Image. — each command creates a layer.

- DockerFile ek recipe hai jisse image banti hai.

Important Command :-

- From → Base image [python, ubuntu]
- Run → Install software (Pkgs)
- Copy → Copy Files [Files ko Image me dalne ke liye]
- WORKDIR → Set working directory.
- Expose → Inform port ex. EXPOSE 8808
- CMD → Default Command [cmd ["python", "app.py"]]
Container start hone pr run hone wala Command.

Docker Container :

A Container = a running Instance of an Image

It's lightweight, portable, and isolated — ensures consistency everywhere.

- Container ek chalta hua Image hota hai — Not a lab app ka actual running version.

Docker Engine :-

:- Core part of Docker that Manages Images & Containers.

Components of Docker Engine:

1] Docker Daemon :

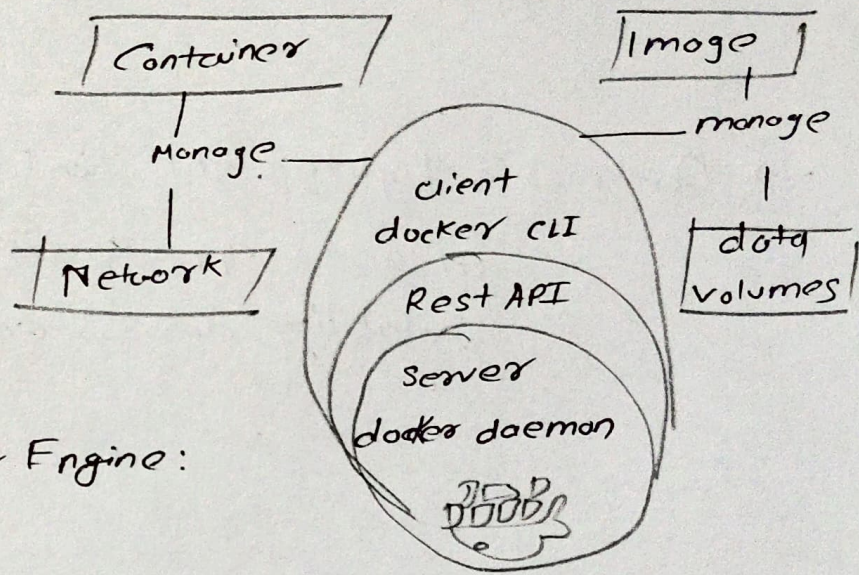
- Run in background, manage container, images, Netw. etc.

2] Docker CLI :

- Command Line tool to interact with Docker
Ex. Docker Build, docker run.

3] Rest API :

- Enables programmatic communication between CLI & Daemon.



Docker Registry :

- A place to store & share Docker Images.
Ex. Docker hub, AWS ECR, Azure ACR.

In Short :

- Docker Image : Blueprint of your Application.
- Docker Container : Running Instance [chalta hua app]
- Dockerfile : Recipe to build Images
- Registry : place to store/store Images
- Docker Engine : The Core System that runs it all

→ Docker = Lightweight Containerization tool that ensures app consistency, isolation & scalability across environments.