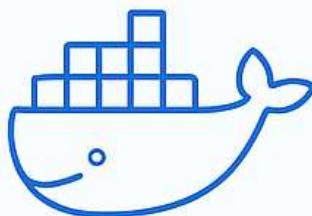


# Docker Fundamentals

Build, Ship, Run Anywhere

## WHAT IS DOCKER

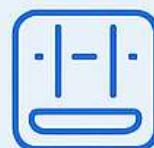


Containerize your app  
and run it anywhere

## WHY DOCKER

### PROBLEM

Apps run  
differently in  
each  
environment



### SOLUTION

Containers  
ensure  
consistent  
execution



## CORE COMPONENTS



**DOCKER IMAGE**  
App blueprint



**DOCKER CONTAINER**  
App instance

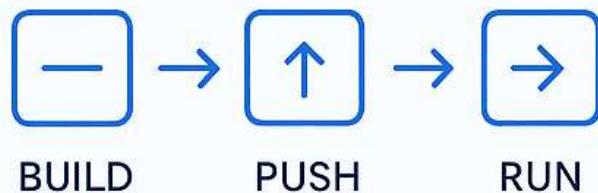


**DOCKER ENGINE**  
Container runtime



**DOCKER REGISTRY**  
image storage

## WORKFLOW



## BENEFITS

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

Docker = Lightweight containerization that runs **anywhere**

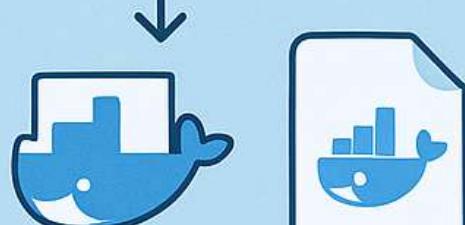
#Docker #Containerization

# FROM CODE TO CONTAINER

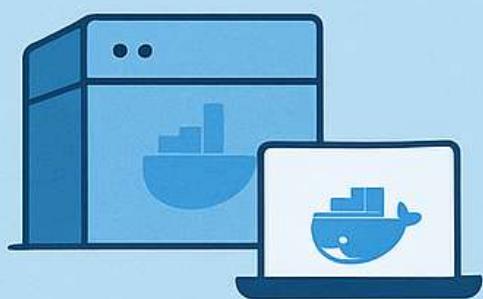
## HOW DOCKER WORKS



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

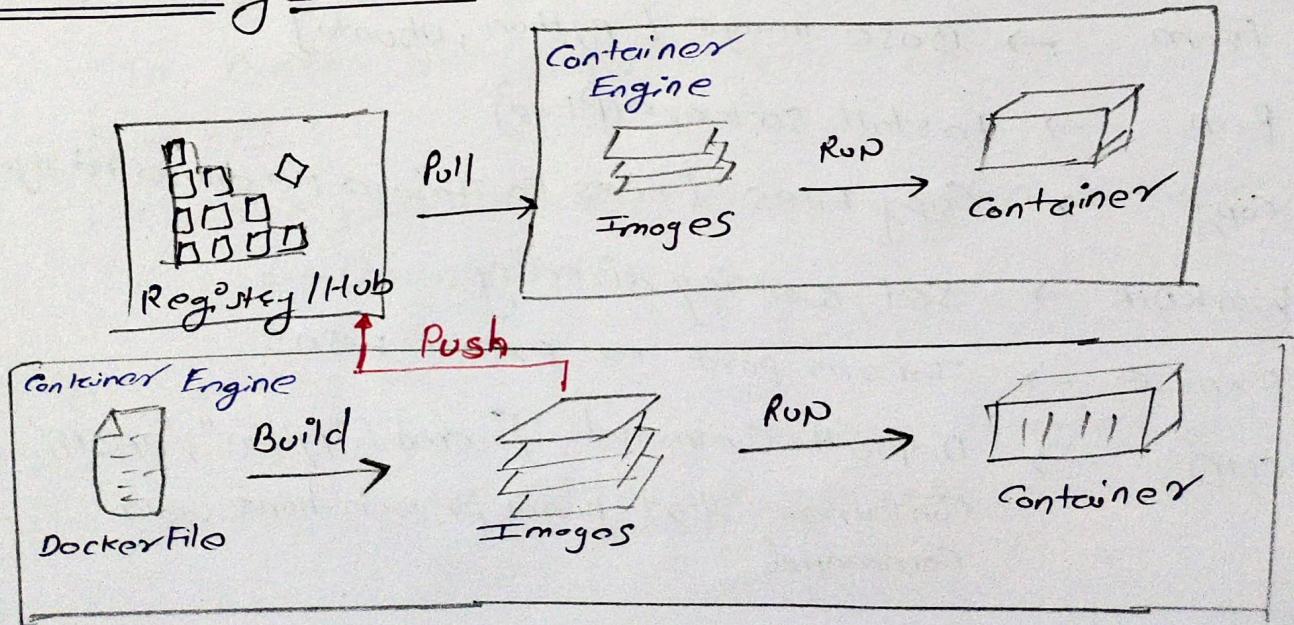
## What is Docker:

- 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 kohi bhi some tarike 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 of 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 Dockerfile 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 8805

CMD → Default command [cmd ["python", "app.py"]]

Container start hone pr run hone wala command.

## # Docker Container :-

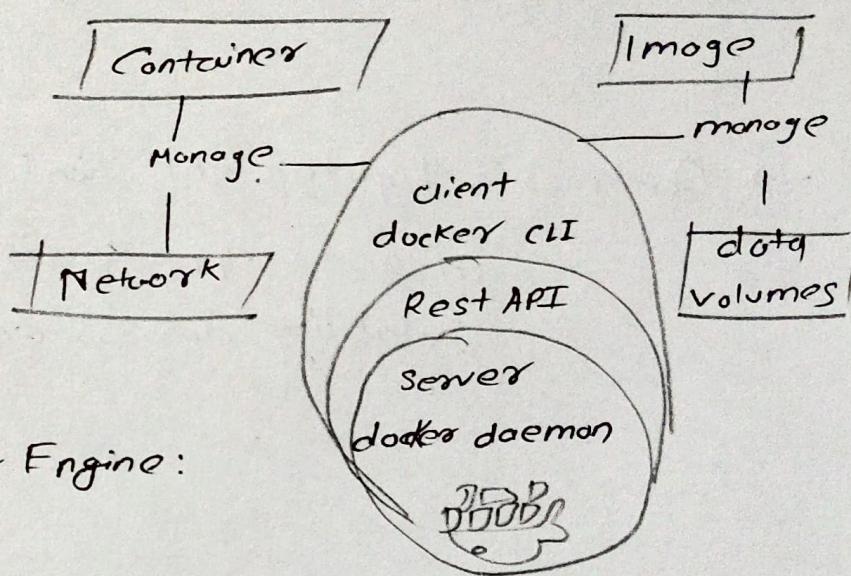
A Container = a running instance of an Image

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

- Container ek chalta huw Image hota hai — Motab 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/share Images
- Docker Engine : The core system that runs it all

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