What is Docker?

Docker is an open-source platform designed to automate the development, shipping, and deployment of applications inside lightweight, portable containers.

Why Docker?

- Consistency: Runs the same on any environment (dev, test, prod).
- Isolation: Each container is independent and does not interfere with others.
- Efficiency: Lightweight compared to virtual machines, uses less system resources.
- Portability: Runs anywhere — on your laptop, server, or cloud.
- Fast Deployment: Applications and dependencies are packaged together.

Key Docker Concepts

Concept	Description
Image	A lightweight, standalone, and executable package that includes everything needed to run a piece of software.
Container	A runtime instance of an image. It's the actual execution environment.
Dockerfile	A script containing instructions to build a Docker image.
Docker Hub	A cloud-based registry where Docker users share images.
Volumes	Used to persist data outside containers.
Docker Compose	Tool to define and run multi-container Docker applications using a `docker-compose.yml` file.

How Docker Works

Build: Write a `Dockerfile` to create your image.

Ship: Push the image to Docker Hub or private registry.

Run: Pull and run the image as a container on any system with Docker installed.

Basic Docker Commands

```
bash

CopyEdit
```

```
docker --version                 # Check Docker version
docker pull ubuntu               # Download an image
docker images                    # List available images
docker run -it ubuntu bash       # Run a container interactively
docker ps                        # List running containers
docker stop <container_id>      # Stop a running container
docker build -t myapp .          # Build image from Dockerfile
docker-compose up                # Start services with Docker Compose
```

Use Cases

- Microservices architecture
- CI/CD pipelines
- Cloud-native apps
- Dev/test environments
- Containerizing legacy apps