Introduction to Docker Command Line

# 1. What is a Docker Image?

A Docker Image is like a blueprint for containers. It contains:  
  
📦 The application (code, dependencies, configurations).  
🔧 The environment (OS, libraries, runtime).  
🔄 Instructions for execution.  
  
💡 Think of a Docker Image like an Apartment Layout:  
 - The blueprint (image) defines the structure.  
 - The apartment (container) is created from the blueprint.  
 - You can create multiple apartments (containers) from the same blueprint (image).

🚀 Example: Pulling a Docker Image

docker pull ubuntu

# 2. What is a Docker Container?

A Docker Container is a running instance of an image. It is an isolated environment that runs applications efficiently and securely.  
  
💡 Think of a Container as an Apartment:  
 - Each apartment (container) is built from a blueprint (image).  
 - Each apartment is isolated from others but shares infrastructure (Docker Engine).  
 - You can have multiple apartments (containers) from one blueprint (image).

🚀 Example: Running an Ubuntu Container

docker run -it ubuntu bash  
ls  
cat /etc/os-release  
exit

# 3. Understanding Docker Image Layers (Union File System - UFS)

Docker optimizes storage by using layered images:  
✅ Shared layers → Shared across containers to save space.  
✅ Copy-on-Write (CoW) → Changes stored in a new layer.  
  
💡 Think of Image Layers like an Apartment Building:  
 - Foundation & walls (Base OS Layer) → Shared.  
 - Furnishings (Application & Configuration) → Customized.  
  
🚀 Example: Viewing Image Layers  
docker history ubuntu

# 4. Managing Docker Images

* 4.1 Listing Available Images:  
   docker images
* 4.2 Removing an Image:  
   docker rmi ubuntu
* 4.3 Cleaning Up Unused Images:  
   docker image prune

# 5. Creating Your Own Docker Image (Dockerfile Basics)

A Dockerfile is a script with instructions to build a custom Docker image.  
  
📌 Think of a Dockerfile like an Apartment Contract:  
 - Defines what gets installed.  
 - Ensures consistency.  
 - Allows automation.

* 5.1 Writing a Simple Dockerfile:  
  mkdir my-docker-app && cd my-docker-app  
  nano Dockerfile  
    
  # Use Ubuntu as the base image  
  FROM ubuntu  
    
  # Install Python  
  RUN apt update && apt install -y python3  
    
  # Set default command  
  CMD ["python3", "--version"]
* 5.2 Building Your Docker Image:  
   docker build -t my-python-app .
* 5.3 Running Your Custom Image:  
   docker run my-python-app

# 6. Managing Docker Containers

* 6.1 Listing Running Containers:  
   docker ps
* 6.2 Listing All Containers (Including Stopped Ones):  
   docker ps -a
* 6.3 Stopping a Running Container:  
   docker stop <container\_id>
* 6.4 Removing a Container:  
   docker rm <container\_id>
* 6.5 Cleaning Up Unused Containers:  
   docker container prune