Docker

# Definition:

Docker is an open-source platform designed to simplifying process of developing, deploying and running applications using containerization.

## Containers: Lightweight, standalone executable software packages that includes everything needed to run an application (e.g., code, runtime, system tools, libraries, settings)

# Components:

* **Docker Engine:** The core component that builds and runs containers.
* **Docker Images:** Read-only template used to create containers (e.g., ubuntu: latest, nginx: alpine)
* **Docker File:** A text file containing instructions to build a docker image.
* **Docker Daemon (dockerd):** Manages images, containers and networks.
* **Docker Hub:** A cloud-based registry for sharing and managing docker images.

# Building Block

## Docker Engine:

The heart of Docker, responsible for running containers.

Consists of:

* Docker Daemon (dockerd) – Background service that manages Docker objects (images, containers, networks, etc.).
* Docker CLI (docker) – Command-line tool to interact with the Docker Daemon.
* REST API – Allows programs to control Docker.

## Docker Images:

* Read-only templates used to create containers.
* Built from Dockerfiles (a set of instructions).
* Stored in registries (like Docker Hub).
* Example: ubuntu:22.04, nginx:alpine.

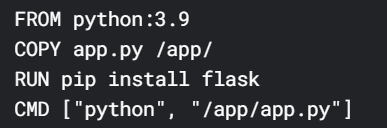
## Docker Containers

* Running instances of Docker images.
* Lightweight, isolated environments where applications execute.
* Can be started, stopped, moved, or deleted.

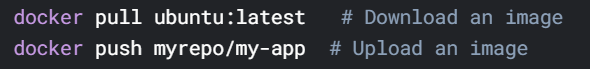
## Dockerfile

A text file containing step-by-step instructions to build a Docker image.

Defines:

* Base image (FROM).
* Dependencies (RUN, COPY).
* Commands to run (CMD (start the app), ENTRYPOINT).

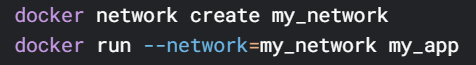
## Docker Registries & Repositories

* **Registries** – Stores Docker images (e.g., Docker Hub, AWS ECR, Google Container Registry).
* **Repositories** – Collections of related images (e.g., library/ubuntu, user/my-app).

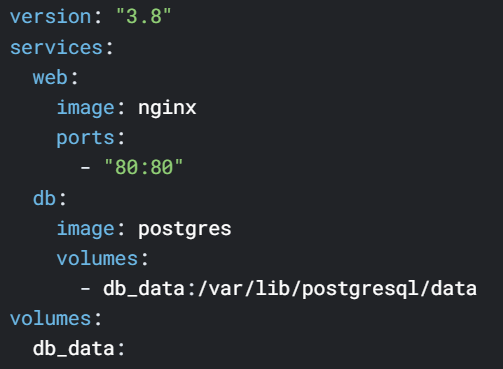
## Docker Volumes

* Persistent storage for containers (data survives container restarts).
* Used for databases, logs, or shared files.

## Docker Networks

* Enables communication between containers & external systems.
* Types:
  + **Bridge** (Default, isolated private network).
  + **Host** (Shares host’s network).
  + **Overlay** (For multi-host Docker Swarm/Kubernetes).

## Docker Compose

* A tool to **define and run multi-container apps** using a YAML file (docker-compose.yml).
* Simplifies managing services, networks, and volumes.

## Commands:

**Installation:**

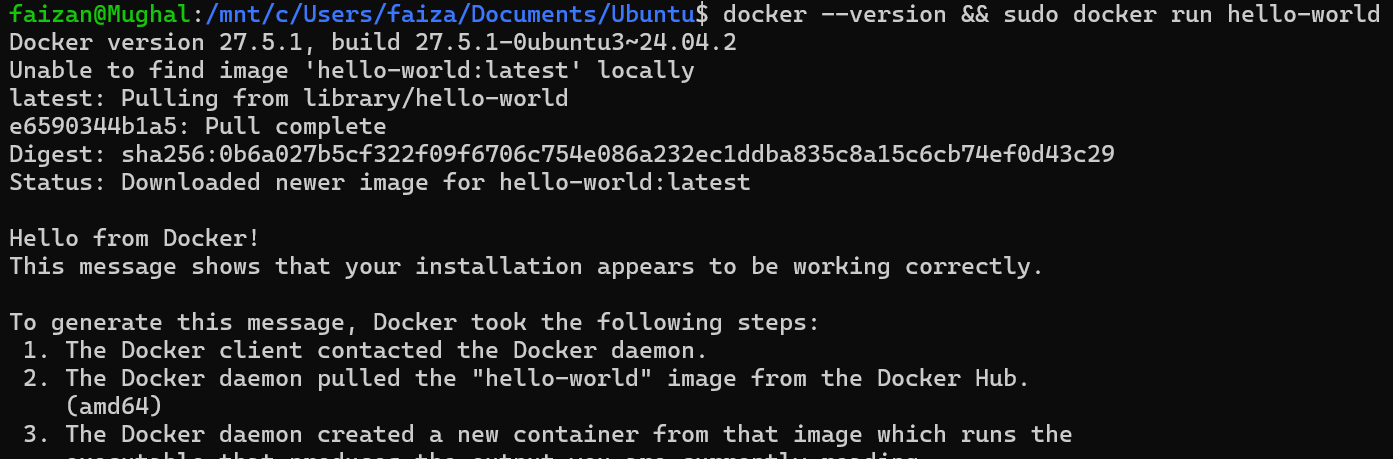
sudo apt update #ensures that your system has the latest package information

sudo apt install docker.io -y #installs Docker on your system (-y automatically answers "yes" to any prompts)

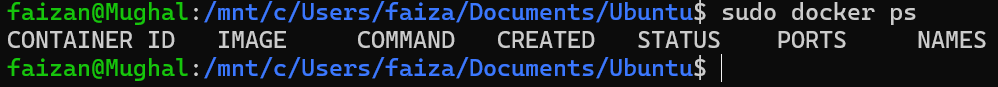
**Staring**

sudo systemctl start docker && sudo systemctl enable docker #start docker and ensures it starts at every system boot.

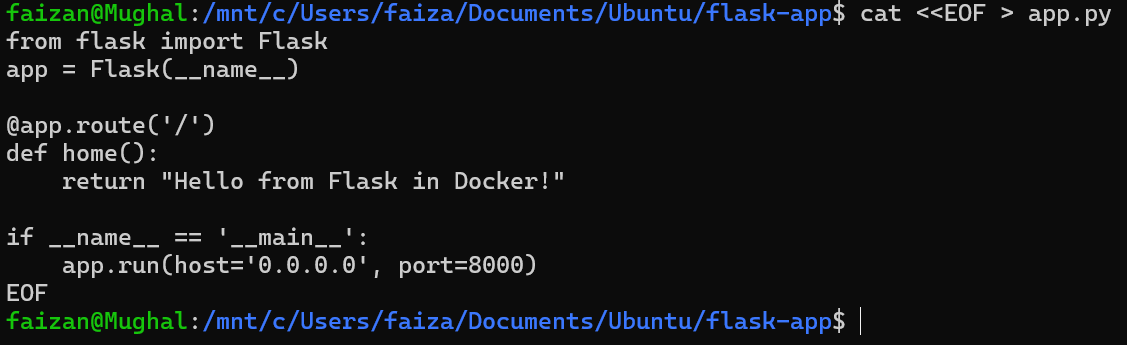
**Verifying**

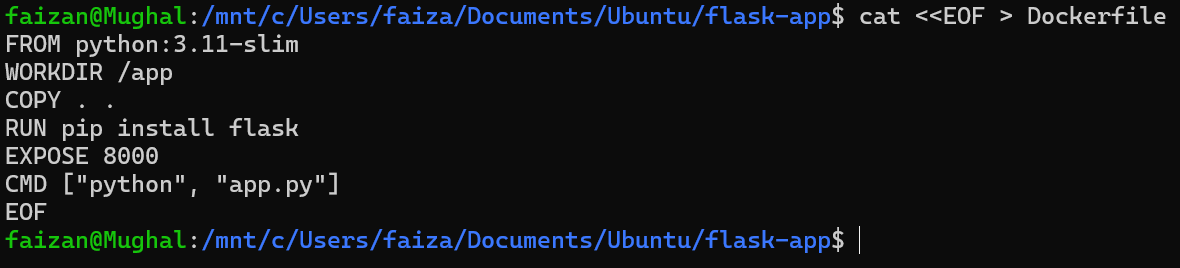
****docker –version && sudo docker run hello-world

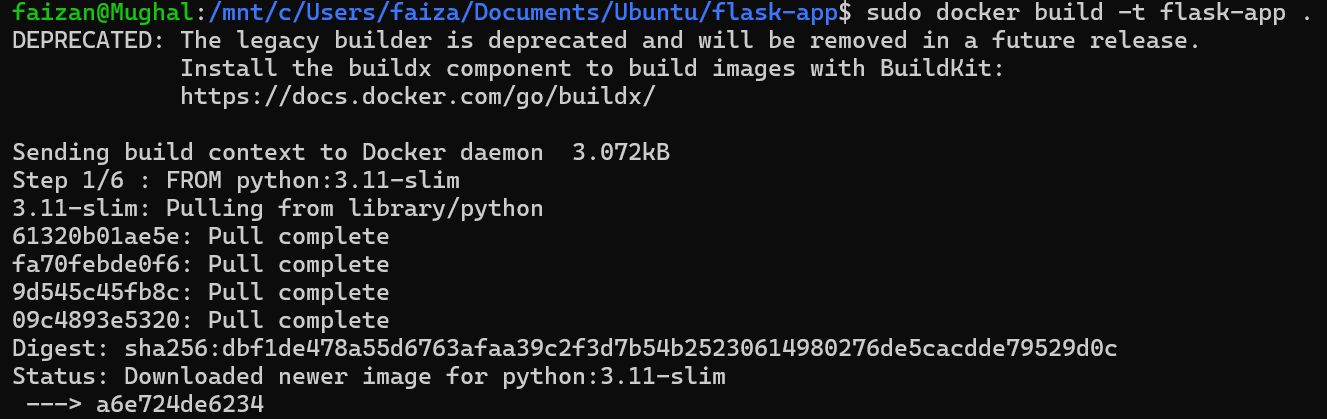
**Check Running Docker Container**

****

TASK: 1

Creating app.py

Creating Docker File

Builds a Docker Image

Run

sudo docker run -p 8000:8000 flask-app

Stop Containers

sudo docker <container id>