Introduction to Docker

1. What is Docker?

Docker is a platform that allows you to create, deploy, and run applications in containers.

Containers are lightweight, portable, and consistent environments for your application,

ensuring it works the same way anywhere.

2. Installing Docker

- Windows/Mac: Download Docker Desktop from https://www.docker.com/products/docker-desktop.

- Linux: Follow instructions on Docker's installation guide (https://docs.docker.com/engine/install/).

3. Basic Docker Concepts

- Docker Images: A snapshot of your application (like a template).

- Docker Containers: Running instances of Docker images (like running a program from a template).

- Dockerfile: A script with instructions to build a Docker image.

- Docker Hub: A repository for sharing Docker images.

4. Running Your First Docker Container

- Open your terminal or command prompt.

- Run a simple container to test Docker is working:

```
docker run hello-world
```

5. Docker Commands

Here are some basic Docker commands to get started:

- Check Docker version:

```
docker --version
```

- List running containers:

    docker ps

- List all containers (including stopped ones):

    docker ps -a

- Stop a container:

    docker stop <container_id>

- Remove a container:

    docker rm <container_id>

- Build a Docker image from a Dockerfile:

    docker build -t my-image-name .

- Run a container from an image:

    docker run -d my-image-name

6. Creating a Dockerfile

A Dockerfile is a script that contains instructions on how to build a Docker image.

Here's an example for a simple Python app:

Create a file called Dockerfile in your project directory:

```
FROM python:3.9-slim
WORKDIR /app
COPY . /app
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
```

Build the Docker image:

```
docker build -t my-python-app .
```

Run the Docker container:

```
docker run -d -p 5000:5000 my-python-app
```

7. Docker Compose (for multi-container applications)

Docker Compose allows you to define and run multi-container applications.

Create a docker-compose.yml file:

Example docker-compose.yml:

```
version: "3.8"
services:
  web:
    image: my-python-app
    build: .
    ports:
      - "5000:5000"
```

Run the application with:

```
docker-compose up
```

8. Managing Docker Images

- List images:

```
docker images
```

- Remove an image:

```
docker rmi <image_id>
```

9. Stopping & Removing Containers and Images

- Stop all running containers:

```
docker stop $(docker ps -q)
```

- Remove all stopped containers:

```
docker rm $(docker ps -a -q)
```

- Remove all unused images:

```
docker rmi $(docker images -q)
```