**Lab Exercise 1- Docker Image for Metaflow Workflow**

This exercise will help you containerize a simple Metaflow workflow, allowing for easy distribution and portability.

**Objective**

* Create a simple Metaflow workflow.
* Write a Dockerfile to build a Docker image containing the workflow and dependencies.
* Run the workflow in a Docker container.

**Prerequisites**

Docker: Installed on your machine.

Metaflow: Installed on your local machine (pip install metaflow).

**Step-by-Step Lab Exercise**

**Step 1: Set Up the Directory Structure**

Create a project directory named **metaflow\_docker\_lab** and open it in your IDE.

Inside the directory, create the following structure:

metaflow\_docker\_lab/

├── flows/

│ └── simple\_flow.py # Simple Metaflow workflow

├── Dockerfile # Dockerfile to containerize the Metaflow workflow

└── requirements.txt # Python dependencies

**Step 2: Write the Simple Metaflow Workflow**

Create **simple\_flow.py** inside the flows/ directory with the following code:

from metaflow import FlowSpec, step

class SimpleFlow(FlowSpec):

@step

def start(self):

print("Starting the Metaflow Docker workflow.")

self.next(self.hello)

@step

def hello(self):

print("Hello from inside a Docker container!")

self.next(self.end)

@step

def end(self):

print("Flow completed inside Docker.")

if \_\_name\_\_ == '\_\_main\_\_':

SimpleFlow()

This is a simple workflow with three steps that print messages to demonstrate that the workflow is running.

**Step 3: Create a requirements.txt File**

Create a requirements.txt file in the root directory and add the following line:

metaflow

This will ensure that Metaflow is installed when the Docker container is built.

**Step 4: Write the Dockerfile**

Create a **Dockerfile** in the root directory with the following content:

# Use a Python base image

FROM python:3.8-slim

# Set up a working directory

WORKDIR /app

# Set METAFLOW\_USER environment variable

ENV METAFLOW\_USER="docker\_user"

# Copy requirements file and install dependencies

COPY requirements.txt .

RUN pip install -r requirements.txt

# Copy the Metaflow flow script into the container

COPY flows/simple\_flow.py /app/simple\_flow.py

# Set the entrypoint to run the Metaflow workflow

CMD ["python", "/app/simple\_flow.py", "run"]

**Step 5: Build the Docker Image**

From the root of the metaflow\_docker\_lab directory, build the Docker image with the following command:

docker build -t metaflow-docker-lab .

This command creates a Docker image named metaflow-docker-lab based on the Dockerfile.

**Step 6: Run the Metaflow Workflow in a Docker Container**

Once the Docker image is built, you can run it in a Docker container:

docker run metaflow-docker-lab

This command starts a container and runs the simple\_flow.py Metaflow workflow inside it.

You should see output similar to the following:

Starting the Metaflow Docker workflow.

Hello from inside a Docker container!

Flow completed inside Docker.

**Step 7: Verify and Clean Up**

* Verify: Check that the workflow has executed as expected by reading the output.
* Cleanup: Remove the Docker container and image if desired:

docker rm $(docker ps -a -q) # Remove all containers

docker rmi metaflow-docker-lab # Remove the Metaflow image

This exercise provides a foundation for creating Dockerized Metaflow workflows, which can then be integrated with container orchestration tools like Kubernetes.