# Docker ML Deployment on Flask App – Hands-on Instructions

## Prerequisites:

Ensure Docker Desktop is installed and running on your system.  
Have a Flask application ready with the required files (like app.py, model files, and a Dockerfile) in a designated folder.

## Explanation of Dockerfile Contents

A Dockerfile is a script containing a series of instructions on how to build a Docker image. Below is a typical layout and purpose for each instruction in a Dockerfile for a Flask-based ML application.

### Sample Dockerfile

# Base image: Python 3.8-slim, a lightweight Python image  
**FROM python:3.8-slim**  
  
# Set the working directory in the container  
**WORKDIR /app**  
  
# Copy requirements file to container and install dependencies  
**COPY requirements.txt .  
RUN pip install -r requirements.txt**  
  
# Copy the rest of the application code to the container  
**COPY . .**  
  
# Expose the port that the Flask app will run on  
**EXPOSE 5000**  
  
# Run the Flask application  
**CMD ["python", "app.py"]**

### Explanation of Each Command

• FROM python:3.8-slim: Specifies the base image, a minimal version of Python 3.8, reducing the final image size.  
• WORKDIR /app: Sets the working directory inside the container to /app.  
• COPY requirements.txt .: Copies requirements.txt to the container’s working directory.  
• RUN pip install -r requirements.txt: Installs the Python dependencies.  
• COPY . .: Copies all local files to the container’s /app directory.  
• EXPOSE 5000: Exposes port 5000, where Flask listens by default.  
• CMD ["python", "app.py"]: Starts the Flask app when the container is run.

## Exercise Instructions

### 1. Open Docker and Log In (if required):

Launch Docker Desktop and ensure you are logged in, especially if you plan to push the image to a private container registry.

### 2. Navigate to the Flask App Folder:

Go to the folder where your Flask application files are stored (e.g., exercise\_folder/flaskapp).  
Right-click inside this folder and select “Open in Terminal” or manually navigate there using the command line:

cd path\_to\_your\_folder/exercise\_folder/flaskapp

### 3. Build the Docker Image:

Run the following command to build the Docker image. This command tags the image as pycaret.azurecr.io/pycaret-insurance:latest.

docker build -t pycaret.azurecr.io/pycaret-insurance:latest .

Note: Ensure that the Dockerfile is correctly set up in this directory, with instructions to copy application files and install dependencies.

### 4. Run the Docker Container:

Once the Docker build completes, start the container using:

docker run -d -p 5000:5000 pycaret.azurecr.io/pycaret-insurance

This command runs the container in detached mode (-d), binding port 5000 of your local machine to port 5000 in the container, where the Flask app is listening.

### 5. Access the Application:

Open your web browser and navigate to http://localhost:5000 (or http://127.0.0.1:5000).  
Test your Flask application running within the Docker container.