**To Containerize A Python Application With Flask And Deploy It Using Docker**

1. Install Docker: Make sure Docker is installed on your system.
2. Create a Flask Application: If you don't already have a Flask app, you can create a simple one.

Example `app.py`:

from flask import Flask app = Flask(\_\_name\_\_)

@app.route('/') def hello\_world():

return 'Hello, Docker World!' if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=5000)

1. Create a `requirements.txt` File: List all the dependencies in a `requirements.txt` file. For a basic Flask app, the file will look like this:

Flask==2.2.3

1. Create a `Dockerfile`: This file defines how the Docker image should be built.

Example `Dockerfile`:

```Dockerfile

# Use an official Python runtime as a parent image

FROM python:3.12-slim

# Set the working directory in the container

WORKDIR /app

# Copy the current directory contents into the container at /app

COPY . /app

# Install any needed dependencies

RUN pip install --no-cache-dir -r requirements.txt

# Make port 5000 available to the world outside the container

EXPOSE 5000

# Define environment variable

ENV FLASK\_APP=app.py

# Run app.py when the container launches

CMD ["flask", "run", "--host=0.0.0.0"]

1. Build the Docker Image: In the directory where your `Dockerfile` is located, build the image by running:

docker build -t flask-app .

1. Run the Docker Container: Once the image is built, you can run the container:

docker run -p 5000:5000 flask-app

1. Access the Application: You can now access your Flask application by navigating to `http://localhost:5000` in your browser.

**Terminal Output:**

PS D:\Creation\Flask Container> docker build -t flask-docker-app .

>>

[+] Building 15.8s (10/10) FINISHED docker:desktop-linux

=> [internal] load build definition from Dockerfile 0.1s

=> => transferring dockerfile: 518B 0.0s

=> [internal] load metadata for docker.io/library/python:3.12 4.2s

=> [auth] library/python:pull token for registry-1.docker.io 0.0s

=> [internal] load .dockerignore 0.0s

=> => transferring context: 2B 0.0s

=> [1/4] FROM docker.io/library/python:3.12-slim@sha256:2a638 6.2s

=> => resolve docker.io/library/python:3.12-slim@sha256:2a638 0.0s

=> => sha256:52a9b53e12bf51c83494d6f4faad4d25 3.51MB / 3.51MB 1.7s

=> => sha256:7fe47536e82b130868383b654c94cf 13.63MB / 13.63MB 2.8s

=> => sha256:2a6386ad2db20e7f55073f69a98d6da2 9.12kB / 9.12kB 0.0s

=> => sha256:c013e32e74b0ed13ac59533a446bcfcc 1.75kB / 1.75kB 0.0s

=> => sha256:668757ec60efb72ddf3c03bc8d452cc7 5.30kB / 5.30kB 0.0s

=> => sha256:2d429b9e73a6cf90a5bb85105c8118 29.13MB / 29.13MB 2.4s

=> => sha256:ae157567daae7ded9ad41f2a2e4206df88d5 250B / 250B 2.2s

=> => extracting sha256:2d429b9e73a6cf90a5bb85105c8118b30a1b2 2.1s

=> => extracting sha256:52a9b53e12bf51c83494d6f4faad4d25ce4d1 0.3s

=> => extracting sha256:7fe47536e82b130868383b654c94cfb966a68 1.0s

=> => extracting sha256:ae157567daae7ded9ad41f2a2e4206df88d54 0.0s

=> [internal] load build context 0.1s

=> => transferring context: 815B 0.1s

=> [2/4] WORKDIR /app 0.1s

=> [3/4] COPY . . 0.0s

=> [4/4] RUN pip install --no-cache-dir -r requirements.txt 4.7s

=> exporting to image 0.2s

=> => exporting layers 0.1s

=> => writing image sha256:ad78346652833a7df3cd4a42f74533a62d 0.0s

=> => naming to docker.io/library/flask-docker-app 0.0s

PS D:\Creation\Flask Container> docker run -p 5000:5000 flask-docker-app

>>

\* Serving Flask app 'app' (lazy loading)

\* Environment: production

WARNING: This is a development server. Do not use it in a production deployment.

Use a production WSGI server instead.

\* Debug mode: off

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

\* Running on all addresses (0.0.0.0)

\* Running on http://127.0.0.1:5000

\* Running on http://172.17.0.2:5000

Press CTRL+C to quit

172.17.0.1 - - [30/Nov/2024 06:46:35] "GET / HTTP/1.1" 200 -

172.17.0.1 - - [30/Nov/2024 06:46:36] "GET /favicon.ico HTTP/1.1" 404 –



