51. Create a simple Hello-world python flask application and create the docker image of that Flask application

To create a simple "Hello, World" Python Flask application and package it into a Docker image, follow these steps:

1. Set Up the Project Directory

Create a directory for the project:

bash

Copy code

```bash
mkdir hello-flask-app

cd hello-flask-app
```

2. Create the Flask Application

Inside the project directory, create a file named `app.py` for the Flask application:

app.py

python

Copy code

```python
from flask import Flask


app = Flask(__name__)


@app.route('/')

def hello():

    return "Hello, World!"
```

```python
if __name__ == '__main__':

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

3. Create a Requirements File

To specify the dependencies, create a `requirements.txt` file:

requirements.txt

plaintext

Copy code

```
flask
```

4. Create the Dockerfile

Now, create a Dockerfile to containerize the Flask application.

Dockerfile

dockerfile

Copy code

```dockerfile
# Use the official Python image from Docker Hub

FROM python:3.9-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 packages specified in requirements.txt

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


# Make port 5000 available to the world outside this container

EXPOSE 5000


# Define environment variable

ENV FLASK_APP=app.py


# Run the application

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

5. Build the Docker Image

Run the following command to build the Docker image:

bash

Copy code

```
docker build -t hello-flask-app .
```

6. Run the Docker Container

Run the container from the newly created Docker image:

bash

Copy code

```
docker run -p 5000:5000 hello-flask-app
```

7. Test the Flask Application

Open your browser and go to `http://localhost:5000`. You should see the message "Hello, World!" from your Flask application.

With this setup, you've created a simple Flask application, containerized it with Docker, and can access it on your local machine!