

Srusti G C
1RV24MC106

Program 1: Build a Docker container from custom Docker file

Step 1: mkdir -p ~/DevOps/Program1

cd DevOps/Program1

Step 2: nano requirements.txt

Flask

Step 3: nano app.py

from flask import Flask

```
app = Flask(__name__)
```

```
@app.route("/")
```

```
def hello():
```

```
    return "Hello, Docker!"
```

```
if __name__ == "__main__":
```

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

Step 4: nano Dockerfile

```
# This is a Dockerfile
```

```
# It builds a container image for a simple Flask app
```

```
# Use a slim Python base image
```

```
FROM python:3.9-slim
```

```
# Set the working directory inside the container
```

```
WORKDIR /app
```

```
# Copy the requirements file and install dependencies
```

```
COPY requirements.txt .
```

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

```

# Copy the application code
COPY . .

# Set Flask environment variables
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
ENV FLASK_RUN_PORT=5000

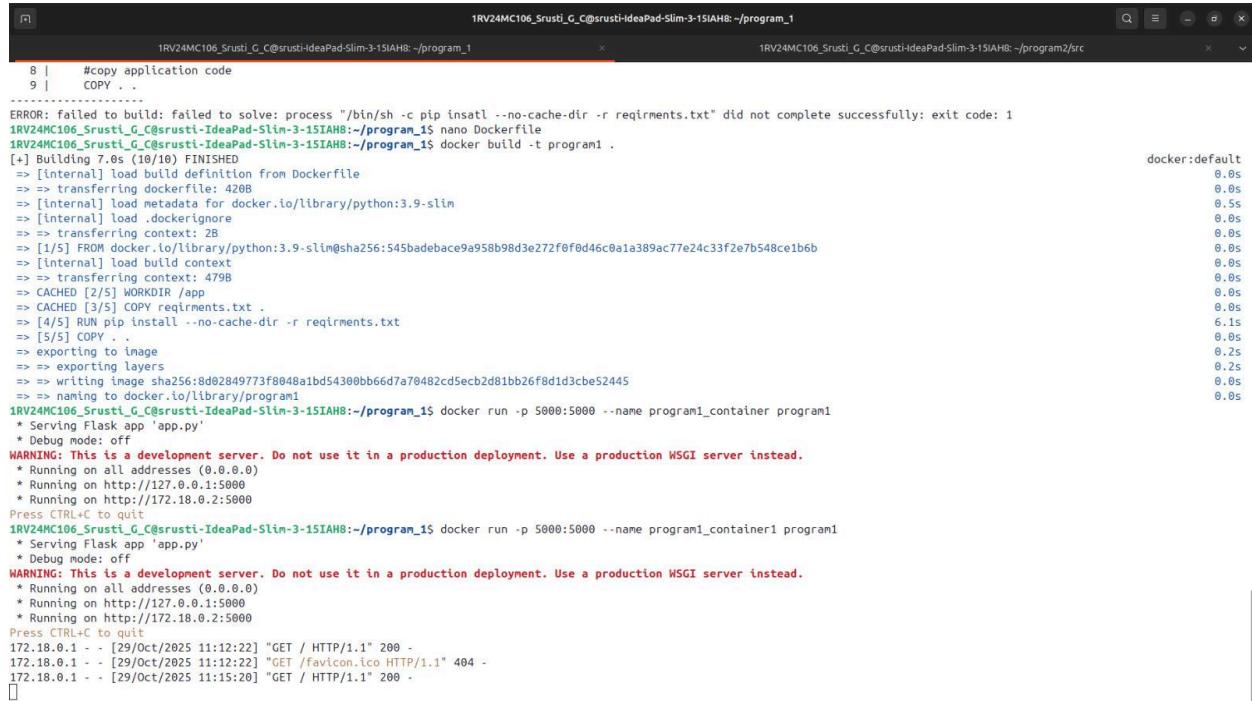
# Expose the port the Flask app will run on
EXPOSE 5000

# Define the command to run the Flask application
CMD ["flask", "run"]

```

Step 5: docker build -t flask-app .

Step 6: docker run -p 5000:5000 flask-app



```

1RV24MC106_Srusti_G_C@srusti-IdeaPad-Slim-3-15IAH8:~/program_1
1RV24MC106_Srusti_G_C@srusti-IdeaPad-Slim-3-15IAH8:~/program_1$ nano Dockerfile
1RV24MC106_Srusti_G_C@srusti-IdeaPad-Slim-3-15IAH8:~/program_1$ docker build -t program1 .
[+] Building 7.0s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring dockerfile: 420B
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [internal] load .dockerignore
=> transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:545badbeace9a958b98d3e272f0f0d46c0a1a389ac77e24c33f2e7b548ce1b6b
=> [internal] load build context
=> transferring context: 479B
=> CACHED [2/5] WORKDIR /app
=> CACHED [3/5] COPY requirements.txt .
=> [4/5] RUN pip install --no-cache-dir -r requirements.txt
=> [5/5] COPY .
=> exporting to image
=> exporting layers
=> writing image sha256:8d02849773f8048a1bd54300bb66d7a70482cd5ecb2d81bb26f8d1d3cbe52445
=> naming to docker.io/library/program1
1RV24MC106_Srusti_G_C@srusti-IdeaPad-Slim-3-15IAH8:~/program_1$ docker run -p 5000:5000 --name program1_container program1
* Serving Flask app 'app.py'
* 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://172.18.0.2:5000
Press CTRL+C to quit
1RV24MC106_Srusti_G_C@srusti-IdeaPad-Slim-3-15IAH8:~/program_1$ docker run -p 5000:5000 --name program1_container1 program1
* Serving Flask app 'app.py'
* 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://172.18.0.2:5000
Press CTRL+C to quit
172.18.0.1 - - [29/Oct/2025 11:12:22] "GET / HTTP/1.1" 200 -
172.18.0.1 - - [29/Oct/2025 11:12:22] "GET /favicon.ico HTTP/1.1" 404 -
172.18.0.1 - - [29/Oct/2025 11:15:20] "GET / HTTP/1.1" 200 -

```

