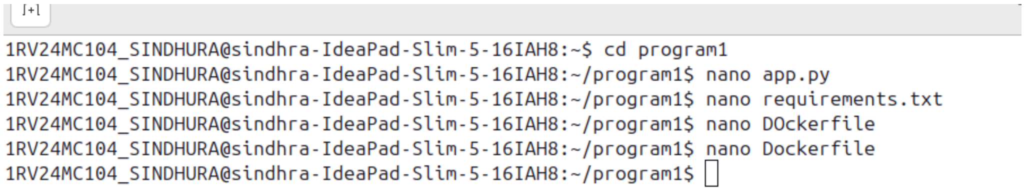
Program 1: Build a Docker Container from a Customer Dockerfile

We have to create a single folder called program 1 so that in that we can create three files called

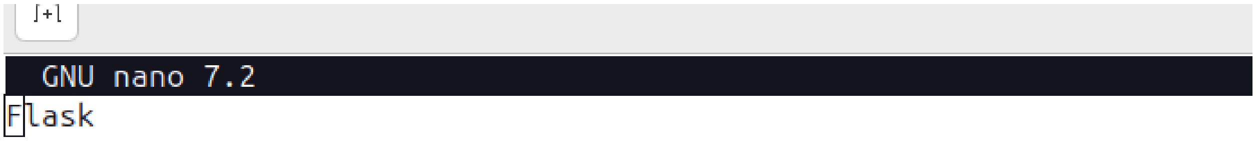
1. Dockerfile
2. app.py
3. requirements.txt



```
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~$ cd program1
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ nano app.py
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ nano requirements.txt
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ nano DOckerfile
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ nano Dockerfile
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ 
```

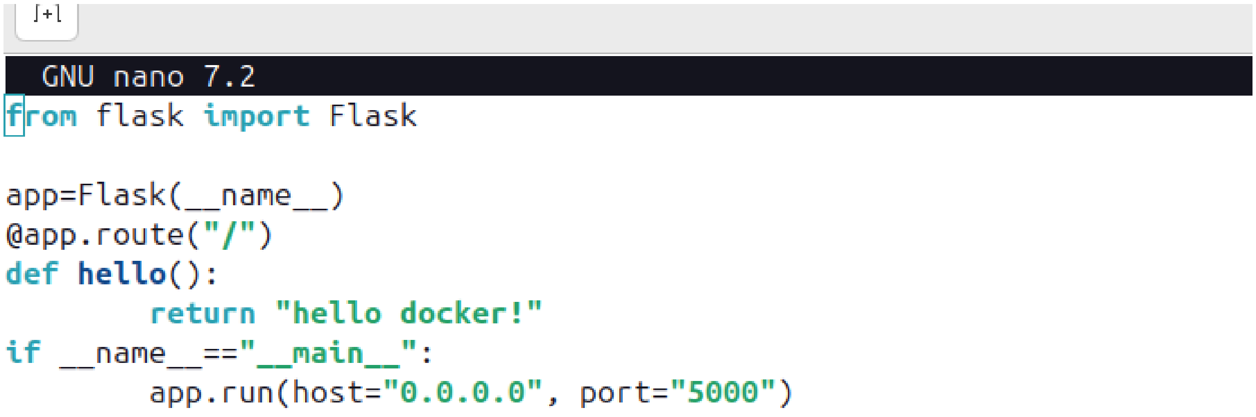
Next we have to add the code in the created files

In requirements.trx



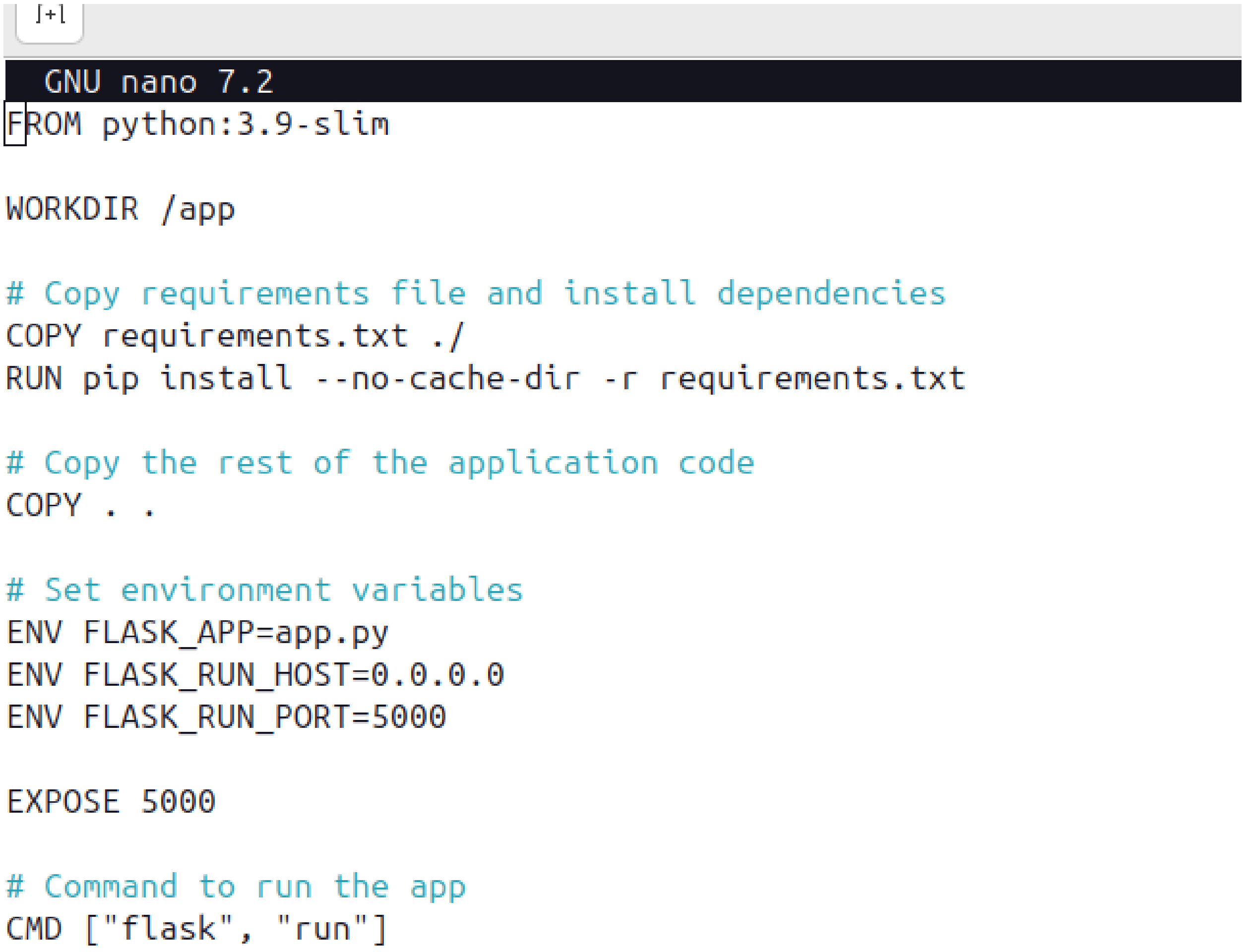
```
  GNU nano 7.2
Flask
```

In appy.py we have to type a code as below



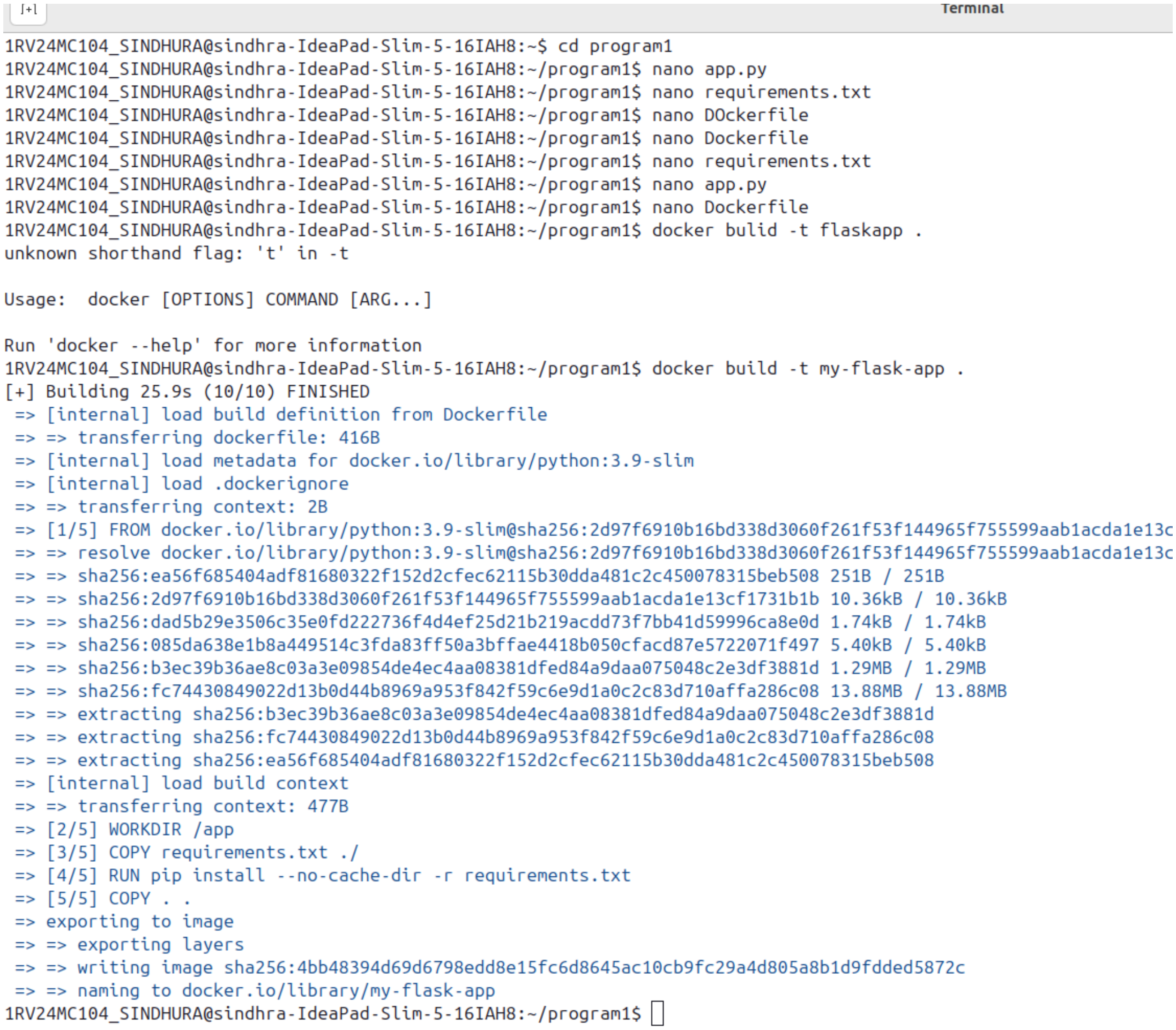
```
from flask import Flask

app=Flask(__name__)
@app.route("/")
def hello():
        return "hello docker!"
if __name__=="__main__":
        app.run(host="0.0.0.0", port="5000")
```

In Dockerfile we have to add the code as below



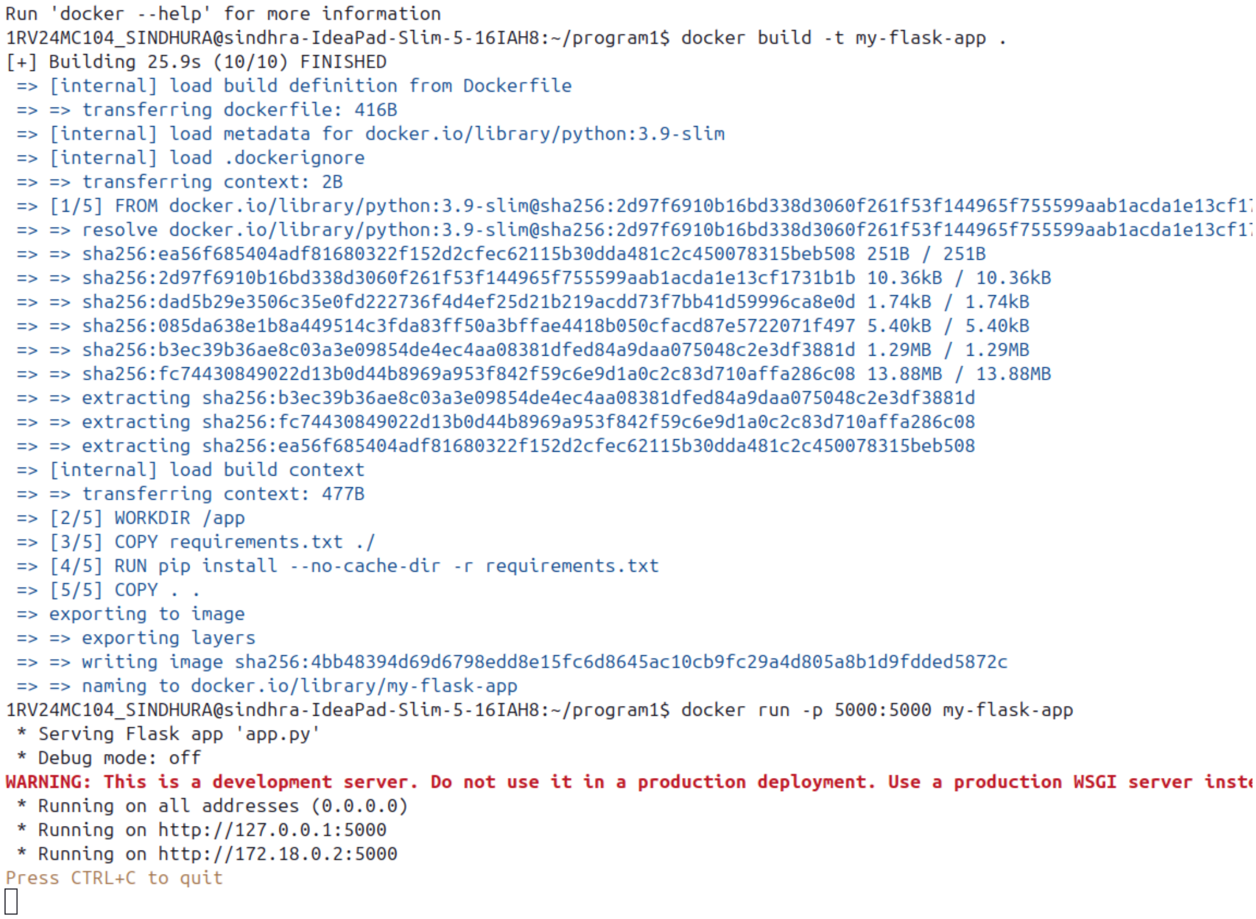
```
FROM python:3.9-slim

WORKDIR /app

# Copy requirements file and install dependencies
COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt

# Copy the rest of the application code
COPY . .

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

EXPOSE 5000

# Command to run the app
CMD ["flask", "run"]
```

After this we have to run the Docker container using docker build command which tells the docker to create a new image using Dockerfile

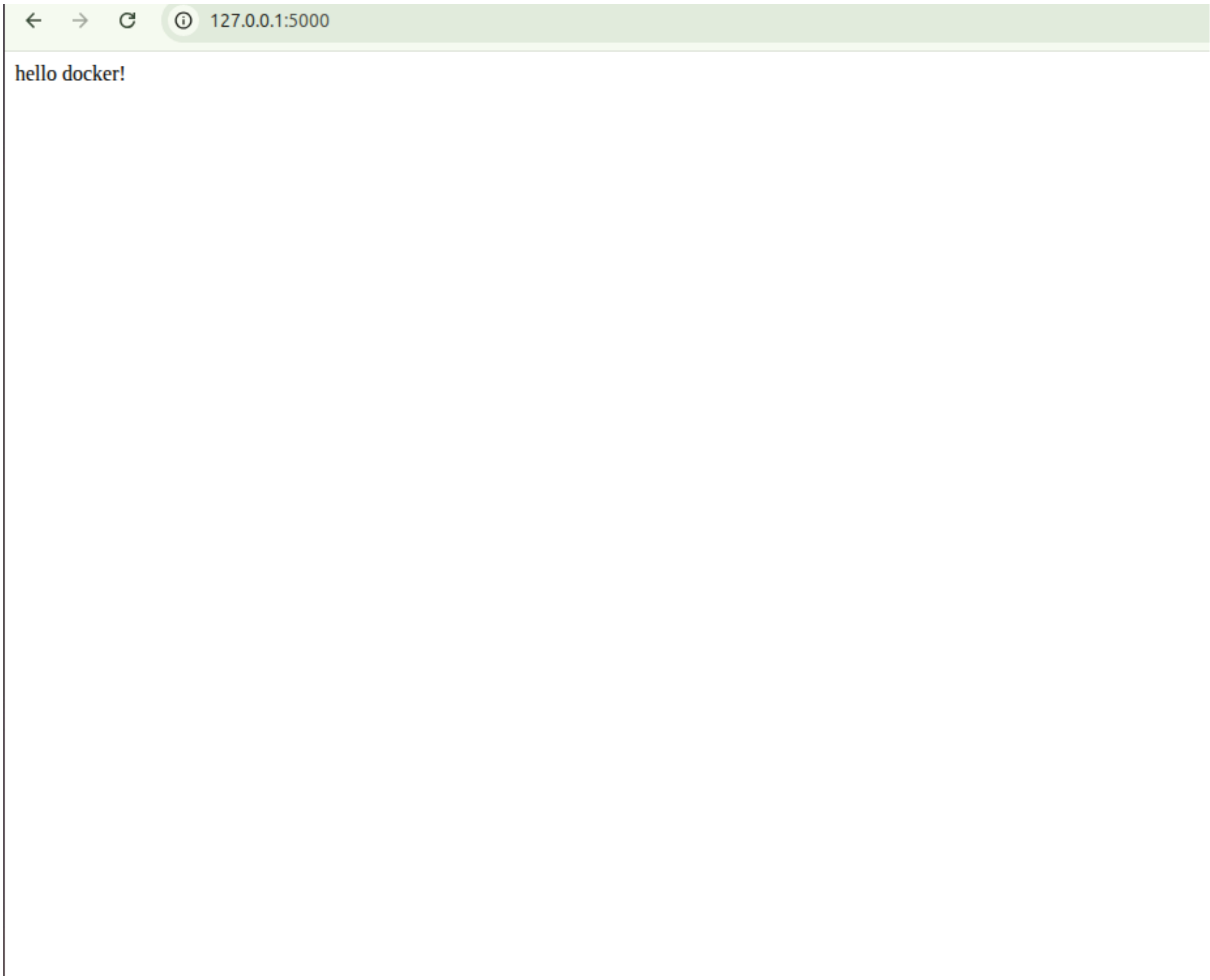


```
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~$ cd program1
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ nano app.py
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ nano requirements.txt
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ nano DOckerfile
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ nano Dockerfile
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ nano requirements.txt
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ nano app.py
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ nano Dockerfile
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ docker bulid -t flaskapp .
unknown shorthand flag: 't' in -t

Usage:  docker [OPTIONS] COMMAND [ARG...]

Run 'docker --help' for more information
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ docker build -t my-flask-app .
[+] Building 25.9s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 416B
 => [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@sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13c
 => => resolve docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13c
 => => sha256:ea56f685404adf81680322f152d2cfec62115b30dda481c2c450078315beb508 251B / 251B
 => => sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13cf1731b1b 10.36kB / 10.36kB
 => => sha256:dad5b29e3506c35e0fd222736f4d4ef25d21b219acdd73f7bb41d59996ca8e0d 1.74kB / 1.74kB
 => => sha256:085da638e1b8a449514c3fda83ff50a3bffae4418b050cfacd87e5722071f497 5.40kB / 5.40kB
 => => sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881d 1.29MB / 1.29MB
 => => sha256:fc74430849022d13b0d44b8969a953f842f59c6e9d1a0c2c83d710affa286c08 13.88MB / 13.88MB
 => => extracting sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881d
 => => extracting sha256:fc74430849022d13b0d44b8969a953f842f59c6e9d1a0c2c83d710affa286c08
 => => extracting sha256:ea56f685404adf81680322f152d2cfec62115b30dda481c2c450078315beb508
 => [internal] load build context
 => => transferring context: 477B
 => [2/5] WORKDIR /app
 => [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 sha256:4bb48394d69d6798edd8e15fc6d8645ac10cb9fc29a4d805a8b1d9fdded5872c
 => => naming to docker.io/library/my-flask-app
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ 
```

Now we have to run Docker container using docj=ker run command which creates the container from the image we built .



```
Run 'docker --help' for more information
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ docker build -t my-flask-app .
[+] Building 25.9s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 416B
 => [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@sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13cf1
 => => resolve docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13cf1
 => => sha256:ea56f685404adf81680322f152d2cfec62115b30dda481c2c450078315beb508 251B / 251B
 => => sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13cf1731b1b 10.36kB / 10.36kB
 => => sha256:dad5b29e3506c35e0fd222736f4d4ef25d21b219acdd73f7bb41d59996ca8e0d 1.74kB / 1.74kB
 => => sha256:085da638e1b8a449514c3fda83ff50a3bffae4418b050cfacd87e5722071f497 5.40kB / 5.40kB
 => => sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881d 1.29MB / 1.29MB
 => => sha256:fc74430849022d13b0d44b8969a953f842f59c6e9d1a0c2c83d710affa286c08 13.88MB / 13.88MB
 => => extracting sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881d
 => => extracting sha256:fc74430849022d13b0d44b8969a953f842f59c6e9d1a0c2c83d710affa286c08
 => => extracting sha256:ea56f685404adf81680322f152d2cfec62115b30dda481c2c450078315beb508
 => [internal] load build context
 => => transferring context: 477B
 => [2/5] WORKDIR /app
 => [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 sha256:4bb48394d69d6798edd8e15fc6d8645ac10cb9fc29a4d805a8b1d9fdded5872c
 => => naming to docker.io/library/my-flask-app
1RV24MC104_SINDHURA@sindhra-IdeaPad-Slim-5-16IAH8:~/program1$ docker run -p 5000:5000 my-flask-app
 * 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 inst
 * 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
```



hello docker!