

## Program1: Build a docker container from a Dockerfile

**step1:** create directory also create app.py and Dockerfile inside the directory

```
rv24mc053_kavya@kavya-Victus-by-HP-Gaming-Laptop-15-fa1xxx:~$ mkdir program1
rv24mc053_kavya@kavya-Victus-by-HP-Gaming-Laptop-15-fa1xxx:~$ cd program1
rv24mc053_kavya@kavya-Victus-by-HP-Gaming-Laptop-15-fa1xxx:~/program1$ sudo nano app.py
[sudo] password for rv24mc053_kavya:
rv24mc053_kavya@kavya-Victus-by-HP-Gaming-Laptop-15-fa1xxx:~/program1$ nano Dockerfile
rv24mc053_kavya@kavya-Victus-by-HP-Gaming-Laptop-15-fa1xxx:~/program1$ nano requirements.txt
```

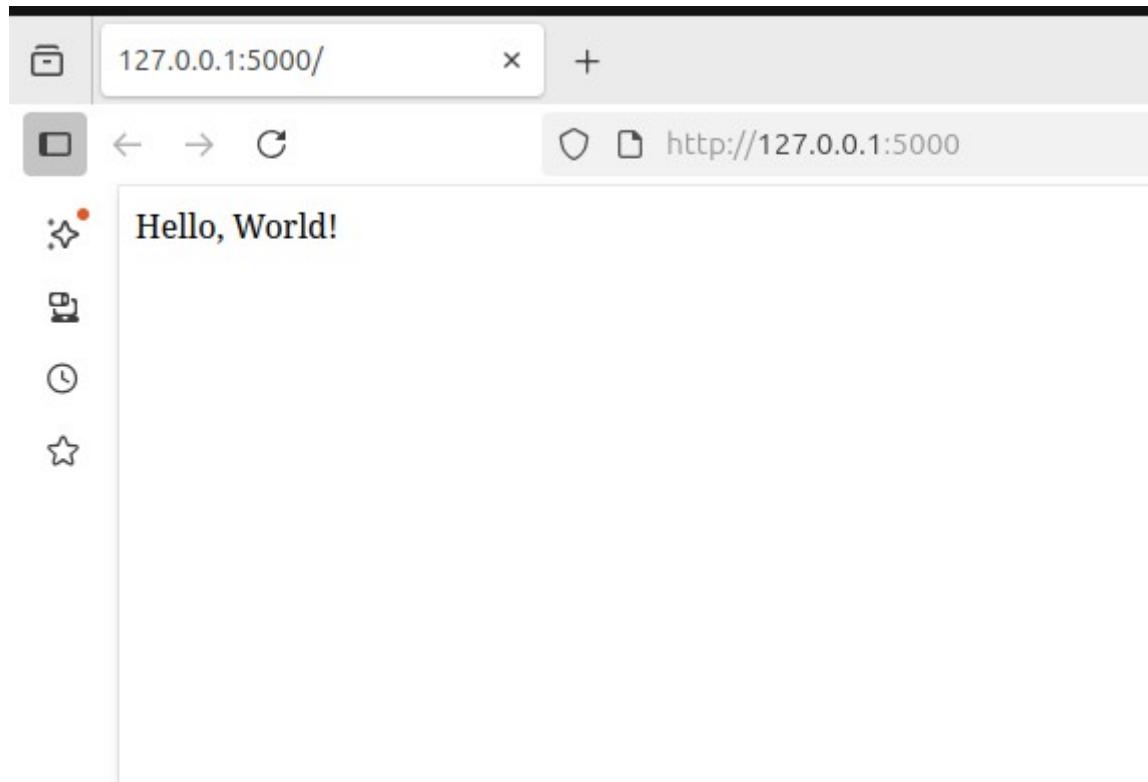
**step2:** build the docker

```
rv24mc053_kavya@kavya-Victus-by-HP-Gaming-Laptop-15-fa1xxx:~/program1$ sudo docker build -t program1 .
[+] Building 19.0s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 554B
=> [internal] load metadata for docker.io/library/python:3.12-slim
=> [internal] load .dockerrcignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/python:3.12-slim@sha256:e97cf9a2e84d604941d9902f00616db7466ff302af4b1c3c67fb7c52
=> => resolve docker.io/library/python:3.12-slim@sha256:e97cf9a2e84d604941d9902f00616db7466ff302af4b1c3c67fb7c52
=> => sha256:e97cf9a2e84d604941d9902f00616db7466ff302af4b1c3c67fb7c522efa8ed9 10.37kB / 10.37kB
=> => sha256:408ad54fa40b7a4fff7cb21ce6e1b74eb5154bfc6c0fb0af98560c1850db88f 1.75kB / 1.75kB
=> => sha256:324231aabbd84383faa67463fb653f94cfab52b71ec3a56f4c49b294aee25968 5.58kB / 5.58kB
=> => sha256:f2a111092025316a53c811d958b8c96faf9e98f6245c9a9cc1b7d3ff8936ac76 1.29MB / 1.29MB
=> => sha256:79f2dc6dd7d858f8963f236b2ab93a0f607fc2e3554e19e244b3c9db593c880b 12.11MB / 12.11MB
=> => sha256:d2876f169c021dd03855501cad6f7f8d06701774c8fbbff1b13e74b46f2ea550 250B / 250B
=> => extracting sha256:f2a111092025316a53c811d958b8c96faf9e98f6245c9a9cc1b7d3ff8936ac76
=> => extracting sha256:79f2dc6dd7d858f8963f236b2ab93a0f607fc2e3554e19e244b3c9db593c880b
=> => extracting sha256:d2876f169c021dd03855501cad6f7f8d06701774c8fbbff1b13e74b46f2ea550
=> [internal] load build context
=> => transferring context: 868B
=> [2/4] WORKDIR /app
=> [3/4] COPY . /app
=> [4/4] RUN pip install --no-cache-dir -r requirements.txt
=> exporting to image
=> => exporting layers
=> => writing image sha256:4c6a52fc24dc195d5b6dd4d03d1cc91b4ed40b90d615294b2c7eb55c7d568e9c
=> => naming to docker.io/library/program1
```

**step3:** connect to port to see the output

```
rv24mc053_kavya@kavya-Victus-by-HP-Gaming-Laptop-15-fa1xxx:~/program1$ sudo docker run -p 5000:5000 program1
 * Serving Flask app 'app'
 * 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 - - [31/Oct/2025 06:33:45] "GET / HTTP/1.1" 200 -
172.18.0.1 - - [31/Oct/2025 06:33:45] "GET /favicon.ico HTTP/1.1" 404 -
[]
```

**step4:** This is the output



**This is app.py file**

```
rv24mc053_kavya@kavya-Victus-by-HP-Gaming-Laptop-15-fa1xxx:~/program1$ cat app.py
from flask import Flask
import os

app = Flask(__name__)

@app.route('/')
def hello():
    name = os.environ.get("NAME", "World")
    return f"Hello, {name}!"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
rv24mc053_kavya@kavya-Victus-by-HP-Gaming-Laptop-15-fa1xxx:~/program1$
```

## This is Dockerfile(no extension)

```
^Crv24mc053_kavya@kavya-Victus-by-HP-Gaming-Laptop-15-fa1xxx:~/program1$ cat Dockerfile
# Use an official Python runtime as a parent image
FROM python:3.12-slim

# Set the working directory inside 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 NAME World

# Run app.py when the container launches
CMD ["python", "app.py"]
```