

DevOps Lab

Program - 1

Build a Docker Container from a Custom Dockerfile

Project Structure

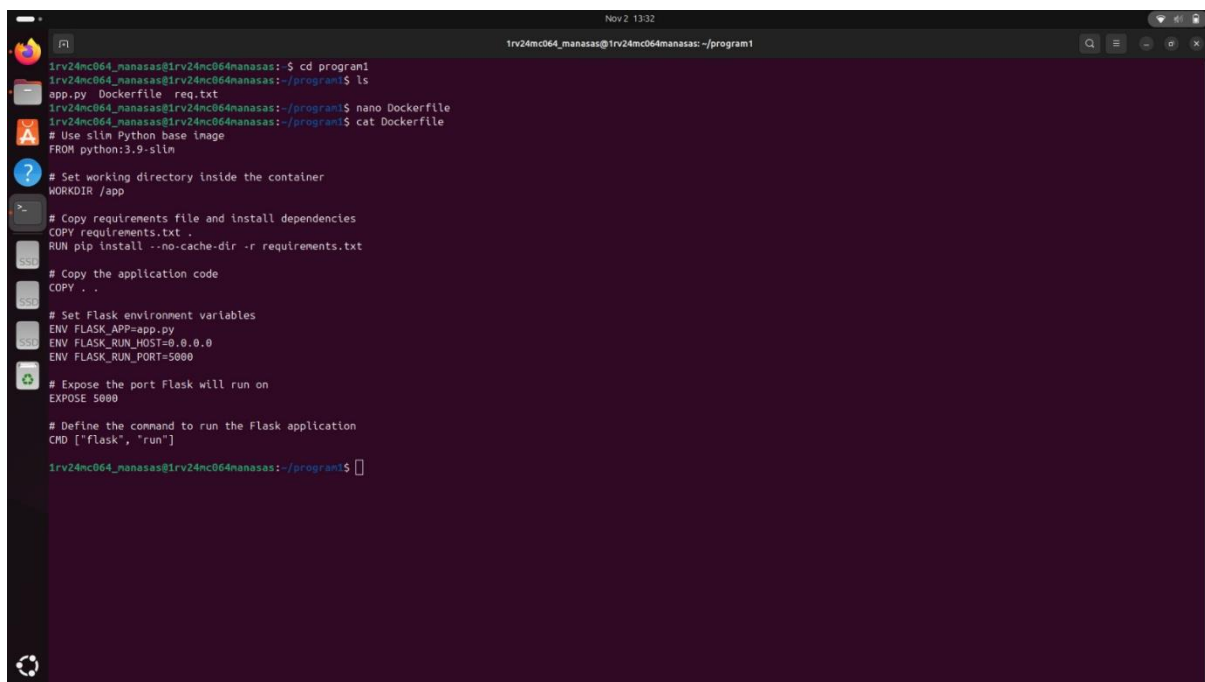
program1/

```
|
|
├── Dockerfile
├── app.py
└── requirements.txt
```

Step 1: Create Project Folder

```
mkdir program1
```

```
cd program1
```

A terminal window with a dark background and light text. The title bar shows 'Nov 2, 13:32' and the user is '1rv24mc064_manasas@1rv24mc064manasas: ~/program1'. The terminal shows the following commands and output:

```
1rv24mc064_manasas@1rv24mc064manasas: $ cd program1
1rv24mc064_manasas@1rv24mc064manasas:~/program1$ ls
app.py  Dockerfile  req.txt
1rv24mc064_manasas@1rv24mc064manasas:~/program1$ nano Dockerfile
1rv24mc064_manasas@1rv24mc064manasas:~/program1$ cat Dockerfile
# Use slim Python base image
FROM python:3.9-slim

# Set working directory inside the container
WORKDIR /app

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

# Copy the application code
COPY . .

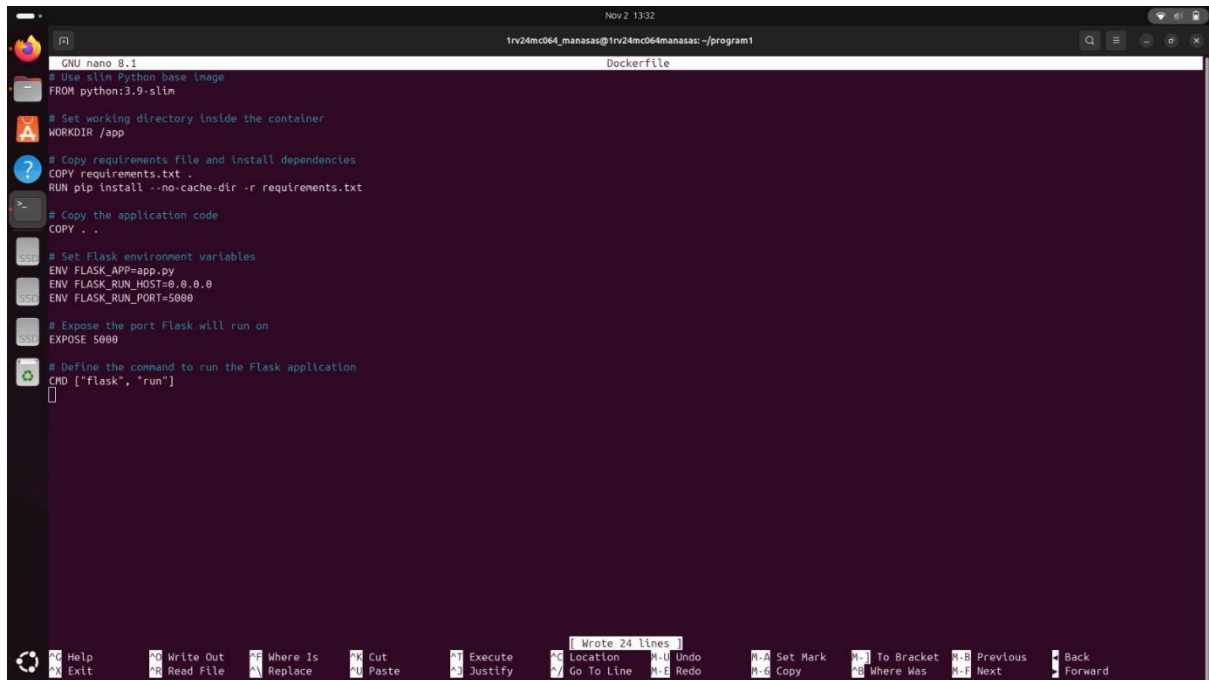
# Set Flask environment variables
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
ENV FLASK_RUN_PORT=5000

# Expose the port Flask will run on
EXPOSE 5000

# Define the command to run the Flask application
CMD ["flask", "run"]
1rv24mc064_manasas@1rv24mc064manasas:~/program1$
```

Step 2: Create Dockerfile

sudo nano Dockerfile



The screenshot shows a terminal window with the nano text editor open to a file named 'Dockerfile'. The editor's status bar at the top indicates 'GNU nano 8.1' and 'Dockerfile'. The file content is as follows:

```
# Use slim Python base image
FROM python:3.9-slim

# Set working directory inside the container
WORKDIR /app

# Copy 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
ENV FLASK_RUN_PORT=5000

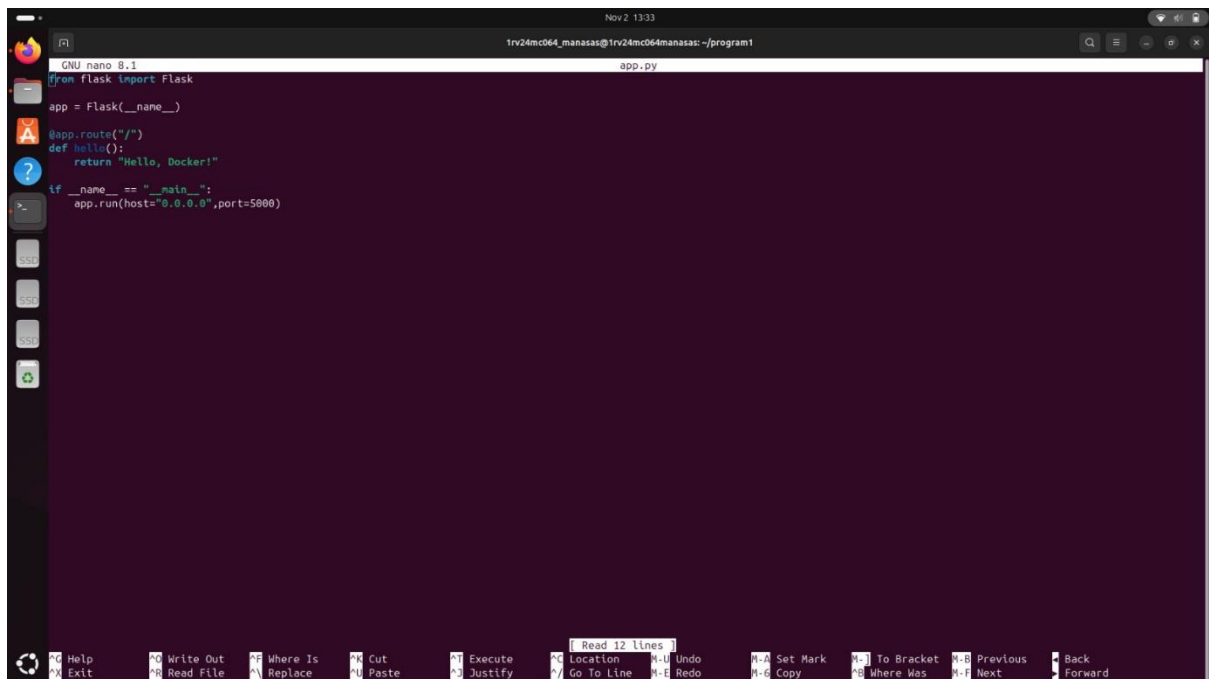
# Expose the port Flask will run on
EXPOSE 5000

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

The nano editor's bottom status bar shows various keyboard shortcuts and indicates 'Wrote 24 lines'.

Step 3: Create Python Application File

nano app.py



The screenshot shows a terminal window with the nano text editor open to a file named 'app.py'. The editor's status bar at the top indicates 'GNU nano 8.1' and 'app.py'. The file content is as follows:

```
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)
```

The nano editor's bottom status bar shows various keyboard shortcuts and indicates 'Read 12 lines'.

nano requirements.txt

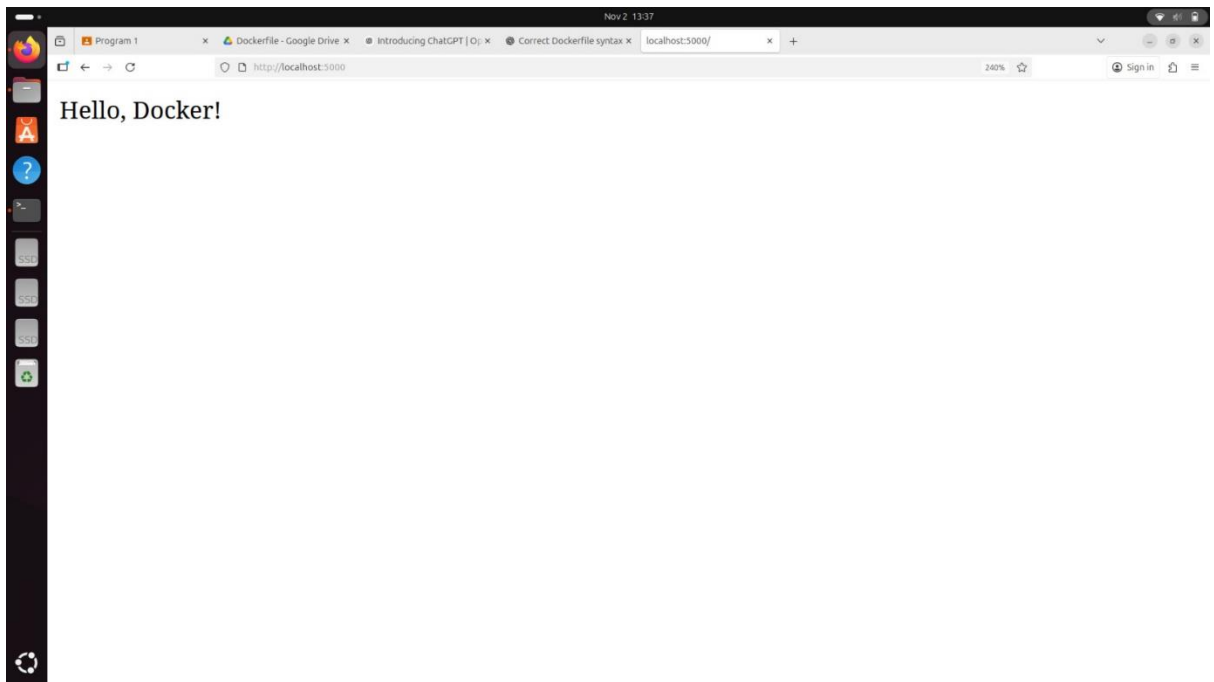
Step 5: Build the Docker Image

```
Nov 23 13:06
t1nv24mc064_manasas@t1nv24mc064manasas: ~/program1
==> ERROR [3/5] COPY require.txt .
0.0%
> [3/5] COPY require.txt .:
-----
Dockerfile:8
-----
6 |
7 | # Copy requirements file and install dependencies
8 | >>> COPY require.txt .
9 | RUN pip install --no-cache-dir -r req.txt
10 |
-----
ERROR: failed to build: failed to solve: failed to compute cache key: failed to calculate checksum of ref 7dbe3070-c6bf-41fc-af61-9a7457b39e44: osq00b8l0st4zica1kfk1ja5w: "/require.txt": not found
t1nv24mc064_manasas@t1nv24mc064manasas: ~/program1$ nano Dockerfile
t1nv24mc064_manasas@t1nv24mc064manasas: ~/program1$ sudo docker build -t firstexe .
[+] Building 12.4s (10/10) FINISHED
==> [internal] load build definition from Dockerfile
0.0%
==> transferring dockerfile: 523B
0.0%
==> [internal] load metadata for docker.io/library/python:3.9-slim
0.4%
==> [internal] load .dockerignore
0.0%
==> transferring context: 2B
0.0%
==> [internal] load build context
0.0%
==> transferring context: 570B
0.0%
==> [1/5] FROM docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd33bd3060f261f53f144965f755599aeb1acd4e13cf173101b
5.6%
==> resolve docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd33bd3060f261f53f144965f755599aeb1acd4e13cf173101b
0.0%
==> sha256:cf74430849022d130bd44b0969a953f842f5c6e6d1a8c2c83d73ba9fa286c08 13.00MB / 13.00MB
4.8%
==> sha256:e56f68540adaf81680322f152d2cfec2115b30dda481c2c45007b315beb508 251B / 251B
1.4%
==> sha256:2d97f6910b16bd33bd3060f261f53f144965f755599aeb1acd4e13cf173101b 10.36KB / 10.36KB
0.0%
==> sha256:50c96e30b10ba049514c3fda83ff50a303ffaa431b05b8cacc087c5722071f497 5.40KB / 5.40KB
0.0%
==> sha256:8050a630b10ba049514c3fda83ff50a303ffaa431b05b8cacc087c5722071f497 5.40KB / 5.40KB
0.0%
==> sha256:30c39b16a0c03a309854d4e4ecaa0831dfed84b0da075040c2a3d3381d 1.29MB / 1.29MB
1.6%
==> extracting sha256:30c39b16a0c03a309854d4e4ecaa0831dfed84b0da075040c2a3d3381d
0.1%
==> extracting sha256:cf74430849022d130bd44b0969a953f842f5c6e6d1a8c2c83d73ba9fa286c08
0.7%
==> extracting sha256:e56f68540adaf81680322f152d2cfec2115b30dda481c2c45007b315beb508
0.0%
==> [2/5] WORKDIR /app
0.1%
==> [3/5] COPY req.txt
0.0%
==> [4/5] RUN pip install --no-cache-dir -r req.txt
0.1%
==> [5/5] COPY .
0.0%
==> exporting to image
0.1%
==> exporting layers
0.1%
==> writing image sha256:d0b53e9ad246ac7315b0db265f1a48330ccaab4f0cab6ff51a2be0822329ee
0.0%
==> pushing to docker.io/library/firstexe
0.0%
t1nv24mc064_manasas@t1nv24mc064manasas: ~/program1
```

```
docker run -d -p 5000:5000 program 1
```

Open your browser and go to: <http://localhost:5000>

```
Nov 2 13:37
1rv24mc064_manasas@1rv24mc064manasas: ~/program1
-----
Dockerfile:8
-----
6 |
7 | # Copy requirements file and install dependencies
8 | >>> COPY requirreq.txt .
9 | RUN pip install --no-cache-dir -r req.txt
10 |
-----
ERROR: failed to build: failed to solve: failed to compute cache key: failed to calculate checksum of ref 7dbe3b70-c6bf-41fc-af61-9a7457b39e44::osq00b8l0st4zica1kflja5w: "/requirreq.tx
t": not found
1rv24mc064_manasas@1rv24mc064manasas:~/program1$ nano Dockerfile
1rv24mc064_manasas@1rv24mc064manasas:~/program1$ sudo docker build -t firstexte .
[+] Building 12.4s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [internal] load .dockerignore
=> [internal] load build context
=> [internal] load build context
=> [internal] load build context
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd33bd366f261f53f144965f75559aab1acda1e13cf1731b1b
=> resolve docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd33bd366f261f53f144965f75559aab1acda1e13cf1731b1b
=> sha256:fc7d43864902d13b0d4b0b99a951f0a7459c4e0d18c2c83d71ba7fa28c00: 13.80MB / 13.80MB
=> sha256:ee56f685494df61680322f152d2cfec62115b30dda401c2c450070315beb500: 251B / 251B
=> sha256:2d97f6910b16bd33bd366f261f53f144965f75559aab1acda1e13cf1731b1b: 10.36KB / 10.36KB
=> sha256:dad5b29e350ec35e0fd22736f4d4e725d21b219acdd73f7bb41d5999ecaeed: 1.74KB / 1.74KB
=> sha256:085d938e1b0a449514c3fda83ff50a3bffa4418b059cfac087e5722071f497: 5.40KB / 5.40KB
=> sha256:b3ec30b3a6ec03a3e09854de4ec4a08381dfed04a9da075048c2e3df30810: 1.29MB / 1.29MB
=> extracting sha256:b3ec30b3a6ec03a3e09854de4ec4a08381dfed04a9da075048c2e3df30810: 0.15
=> extracting sha256:fc7d43864902d13b0d4b0b99a951f0a7459c4e0d18c2c83d71ba7fa28c00: 0.75
=> extracting sha256:ee56f685494df61680322f152d2cfec62115b30dda401c2c450070315beb500: 0.05
=> [2/5] WORKDIR /app
=> [3/5] COPY req.txt .
=> [4/5] RUN pip install --no-cache-dir -r req.txt
=> [5/5] COPY . .
=> exporting to image
=> exporting layers
=> writing image sha256:d0b53e9ad246ach731b08b62a5f1a48330cca0b4f0ca86ff51a2be0022229ee
=> naming to docker.io/library/firstexte
1rv24mc064_manasas@1rv24mc064manasas:~/program1$ sudo docker run -d -p 5000:5000 firstexte
09f239aa207f036c1a0fde61fb65429689c95684be4ab065e8674f6384b4f71e
1rv24mc064_manasas@1rv24mc064manasas:~/program1$
```



Step 8: Verify Running Container

docker ps