

DevOps Lab – Program 1

CREATING A DOCKER FILE

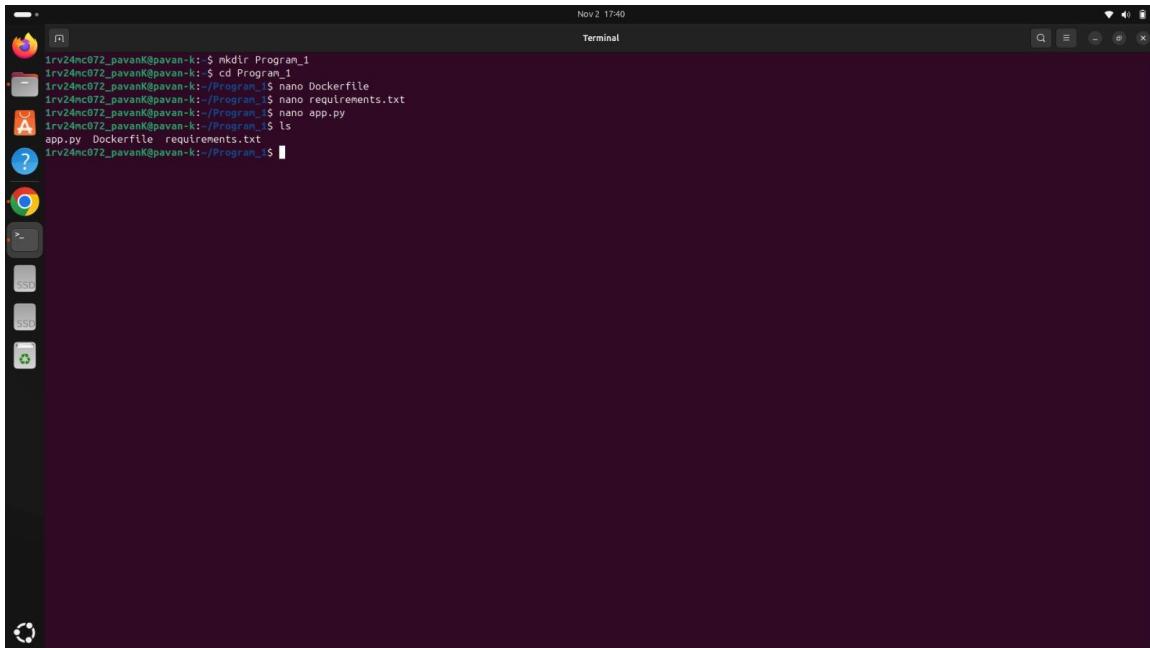
Project Structure

```
program_1/  
|  
+-- Dockerfile  
+-- app.py  
+-- requirements.txt
```

Step 1 – Set Up Project Directory

```
mkdir program_1
```

```
cd program_1
```



The screenshot shows a terminal window on a Linux desktop environment. The terminal output is as follows:

```
Nov 2 17:40  
Terminal  
irv24nc072_pavanK@pavan-k: ~$ mkdir Program_1  
irv24nc072_pavanK@pavan-k: ~/Program_1$ cd Program_1  
irv24nc072_pavanK@pavan-k: ~/Program_1$ nano Dockerfile  
irv24nc072_pavanK@pavan-k: ~/Program_1$ nano requirements.txt  
irv24nc072_pavanK@pavan-k: ~/Program_1$ nano app.py  
irv24nc072_pavanK@pavan-k: ~/Program_1$ ls  
app.py Dockerfile requirements.txt  
irv24nc072_pavanK@pavan-k: ~/Program_1$
```

Step 2 – Create the Dockerfile

Open a new Dockerfile:

```
sudo nano Dockerfile
```

```
Nov 2 17:40
Terminal
irv24nc072_pavan@pavan-k: ~
irv24nc072_pavan@pavan-k: $ mkdir Program_1
irv24nc072_pavan@pavan-k: $ cd Program_1
irv24nc072_pavan@pavan-k:~/Program_1$ nano Dockerfile
irv24nc072_pavan@pavan-k:~/Program_1$ nano requirements.txt
irv24nc072_pavan@pavan-k:~/Program_1$ nano app.py
irv24nc072_pavan@pavan-k:~/Program_1$ ls
app.py Dockerfile requirements.txt
irv24nc072_pavan@pavan-k:~/Program_1$ cat Dockerfile
FROM python:3.9-slim

WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
ENV FLASK_RUN_PORT=5000
EXPOSE 5000
CMD ["flask", "run"]
irv24nc072_pavan@pavan-k:~/Program_1$
```

Step 3 – Write the Flask Application

Create the main Python file:

nano app.py

```
Nov 2 17:41
Terminal
GNU nano 7.2
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)
[ Read 11 lines ]
irv24nc072_pavan@pavan-k:~/Program_1$
```

Step 4 – Specify Dependencies

Create the requirements file:

nano requirements.txt

```

Nov 2 17:40
Terminal

irv24mc072_pavanK@pavan-k: ~$ mkdir Program_1
irv24mc072_pavanK@pavan-k: ~$ cd Program_1
irv24mc072_pavanK@pavan-k: ~/Program_1$ nano Dockerfile
irv24mc072_pavanK@pavan-k: ~/Program_1$ nano requirements.txt
irv24mc072_pavanK@pavan-k: ~/Program_1$ nano app.py
irv24mc072_pavanK@pavan-k: ~/Program_1$ ls
app.py Dockerfile requirements.txt
irv24mc072_pavanK@pavan-k: ~/Program_1$ cat Dockerfile
FROM python:3.9-slim

WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY .

ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
ENV FLASK_RUN_PORT=5000
EXPOSE 5000

CMD ["flask", "run"]
irv24mc072_pavanK@pavan-k: ~/Program_1$ cat requirements.txt
flask
irv24mc072_pavanK@pavan-k: ~/Program_1$ 

```

Step 5 – Build the Docker Image

Execute:

`docker build -t program_1 .`

```

Nov 2 17:44
Terminal

1rv24mc072_pavanK@pavan-k:~/Program_1$ docker build -t flask-demo:1.0 .
[+] Building 17.0s (10/10) FINISHED
   = [internal] load build definition from Dockerfile
   = [internal] load .dockerignore
   = [internal] transfer context to docker:io/library/python:3.9-slim
   = [internal] load , dockerizing...
   = [internal] transfer context: 2B
   = [1/5] FROM docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd33bd3860f261f53f144965f755599aab1cad1e13cf1731b1b
   = [internal] load build definition from Dockerfile
   = [internal] transfer context: 2B
   = [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 sha256:27fd6b579f44bbc40edf2e55c5b410a3c60574039ee7f3adcc847a7a8759837d
   = naming to docker.io/library/flask-demo:1.0
1rv24mc072_pavanK@pavan-k:~/Program_1$ 

```

Step 6 – Run the Container

Launch the container:

`docker run -d -p 5000:5000 program_1`

```
Nov 2 17:45

[+] Building 17.0s (10/10) FINISHED
   => transferring dockerfile: 268B
   => [internal] load dockerfile
   => [internal] load .dockerignore
   => transferring context: 28
  => [1/5] FROM docker.io/library/python:3.9-slim
   => resolve docker.io/library/python:3.9-slim@sha256:209776910b16bd33d3060f261f53f144965f755599aab1cadale1e13cf1731b1b
   => sha256:f7c70430849822d130b044b2698a53f84259c6e9d1ab2c3d1f0a18afaf28ec0 13.88MB / 13.88MB
   => sha256:209776910b16bd33d3060f261f53f144965f755599aab1cadale1e13cf1731b1b
   => sha256:daf5b79e5b06c35ef0d222736f4d4ef25d2b19acdd73f7b041d59996ca8e0d 1.74KB / 1.74KB
   => sha256:085da538e1ba849514c3fcfa8ff50a3bf8e441b0850fcfad87e572207f1497 5.40KB / 5.40KB
   => sha256:3613b0d7256113945cd5db3b30915a633cfca475dc7072ad820191028ca5d 29.78MB / 29.78MB
   => sha256:b3c5e1396aebcb3a3e9854de4ec4aa083b0fe0d8409a0e75048c2e3df3801d 1.29MB / 1.29MB
   => sha256:ea567835494ad1616809322f152d2ccfe621530d0d4a01c2c4509f78319b6508 251B / 251B
   => extracting sha256:385139d7256313495cd083b30915a633cfca474d52a8702a2c0d19182c054
   => extracting sha256:385139d7256313495cd083b30915a633cfca48489310dfe0d49a0a7f584c26d0f3801d
   => extracting sha256:385139d7256313495cd083b30915a633cfca48489310dfe0d49a0a7f584c26d0f3801d
   => extracting sha256:385139d7256313495cd083b30915a633cfca48489310dfe0d49a0a7f584c26d0f3801d
   => extracting sha256:385139d7256313495cd083b30915a633cfca48489310dfe0d49a0a7f584c26d0f3801d
   => [internal] load build context
   => transferring context: 529B
  => [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 sha256:27fd6b579f44bbc40edf2e55c5b410a3c60574639ee7f3addcc847a7a8759837d
  => => writing to docker.io/library/flask:0.10.0

[+] Building 0.0s (0/0)
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.2:5000

Press Ctrl+C to quit
```

Step 7 – Test the Application

Open a web browser and visit:

<http://localhost:5000>

