

Devops lab program – 3

Code a Dockerized Python Flask or Node.js Application

Project structure

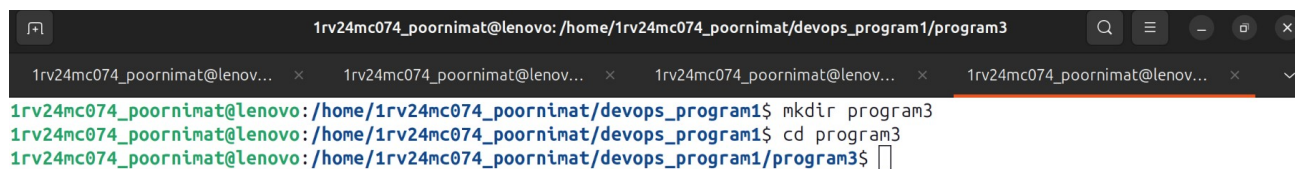
devops_program3

|__Dockerfile

|__app.py

|__requirements.txt

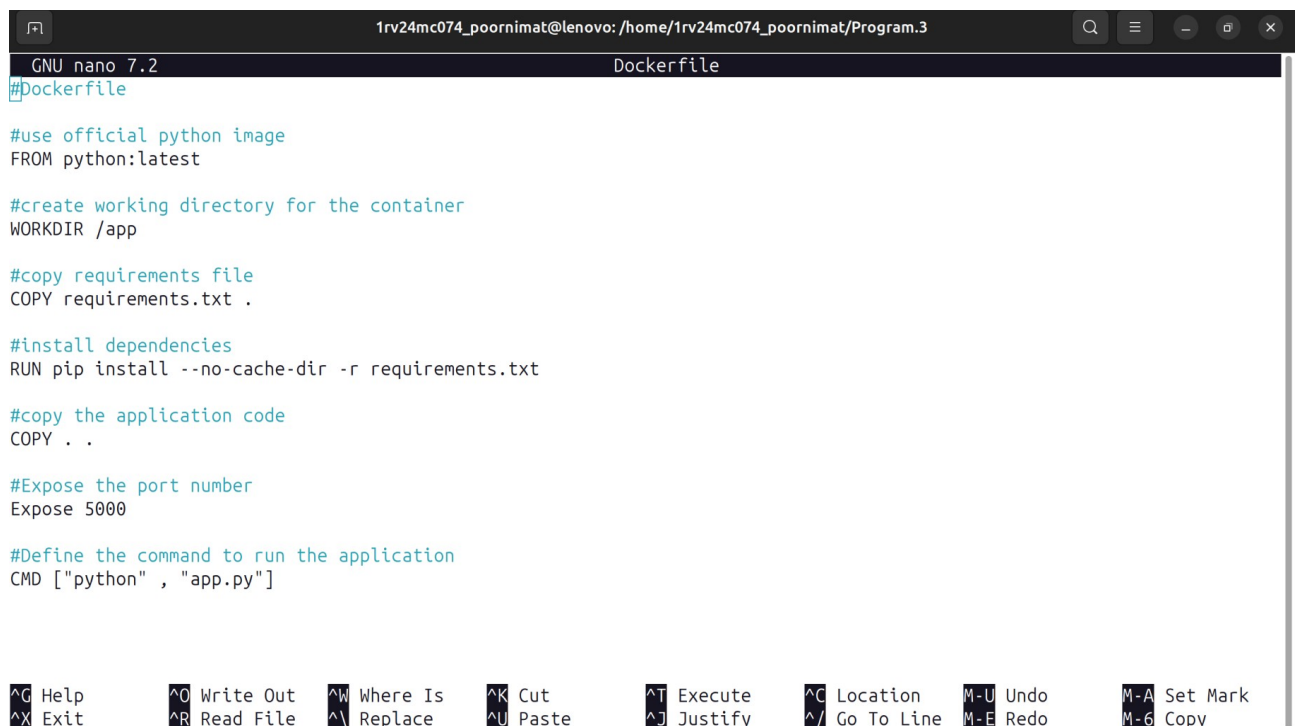
step1:create project folder



```
1rv24mc074_poornimat@lenovo: /home/1rv24mc074_poornimat/devops_program1/program3
1rv24mc074_poornimat@lenovo: /home/1rv24mc074_poornimat/devops_program1$ mkdir program3
1rv24mc074_poornimat@lenovo: /home/1rv24mc074_poornimat/devops_program1$ cd program3
1rv24mc074_poornimat@lenovo: /home/1rv24mc074_poornimat/devops_program1/program3$
```

step2: create docker file

nano Dockerfile



```
GNU nano 7.2 Dockerfile
#Dockerfile

#use official python image
FROM python:latest

#create working directory for the container
WORKDIR /app

#copy requirements file
COPY requirements.txt .

#install dependencies
RUN pip install --no-cache-dir -r requirements.txt

#copy the application code
COPY . .

#Expose the port number
Expose 5000

#Define the command to run the application
CMD ["python" , "app.py"]

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^\ Replace  ^J Paste     ^_ Justify   ^/ Go To Line M-E Redo     M-6 Copy
```

step3:create python application file

nano app.py



```
1rv24mc074_poornimat@lenovo: /home/1rv24mc074_poornimat/Program.3
GNU nano 7.2 app.py
from flask import Flask

app = Flask(__name__)


@app.route('/')
def home():
    return "Hello from flask Docker !"

if __name__ == '__main__':
    app.run(host='0.0.0.0' , port=5000)
```

Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark
Exit Read File Replace Paste Justify Go To Line M-E Redo M-6 Copy

step4:create requirements.txt file

nano requirements.txt



```
1rv24mc074_poornimat@lenovo: /home/1rv24mc074_poornimat/Program.3
GNU nano 7.2 requirements.txt
flask==2.3.3
```

step5:build docker image

sudo docker build -t p1 .

```
1rv24mc074_poornimat@lenovo: /home/1rv24mc074_poornimat/Program.3$ sudo docker build -t p3 .
[+] Building 155.1s (10/10) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 423B                                0.0s
=> WARN: ConsistentInstructionCasing: Command 'Expose' should match the case of the command majority (uppercase) 0.0s
=> [internal] load metadata for docker.io/library/python:latest    5.7s
=> [internal] load .dockerignore                                    0.0s
=> => transferring context: 2B                                       0.0s
=> [internal] load build context                                    0.0s
=> => transferring context: 93B                                       0.0s
=> [1/5] FROM docker.io/library/python:latest@sha256:1ad1a43b5e2478e62056bbc28028afd858185d73bf4d6a439cbb058b6 137.1s
=> => resolve docker.io/library/python:latest@sha256:1ad1a43b5e2478e62056bbc28028afd858185d73bf4d6a439cbb058b680 0.0s
=> => sha256:1ad1a43b5e2478e62056bbc28028afd858185d73bf4d6a439cbb058b6800a96d 10.95kB / 10.95kB 0.0s
=> => sha256:0134b02fcae7fd813645ca13d87412024d3c4324460c0b55e84015951284ad58 6.48kB / 6.48kB 0.0s
=> => sha256:13cc39f8244ac66bf1dd9149e1da421ab1bbc80d612dc14fe368753e7be17b33 49.29MB / 49.29MB 37.3s
=> => sha256:e3143549f2b8b3ad8d79efdc47824641c6771796b3770f3c637a38aabd2b3462 25.62MB / 25.62MB 27.4s
=> => sha256:72e8e93b0d018b135d833207c6feaba205653ac52e0a80d214477ec0de239dee 67.78MB / 67.78MB 64.4s
=> => sha256:c4a87c25190f591b29fb62b4ee424ea9907b4a1997357ed47e82d6db98c80b12 2.32kB / 2.32kB 0.0s
=> => sha256:23d55a674b2e3641a37b7f2053f00af6fe4f2a89576ffb917d4b7b4deaf24591 235.94MB / 235.94MB 127.4s
=> => extracting sha256:13cc39f8244ac66bf1dd9149e1da421ab1bbc80d612dc14fe368753e7be17b33 1.8s
=> => sha256:8f5fb10be9d4525dc6941af4e2a587d1d3edbbb7442cdad0eebb1a3806b183dd 6.09MB / 6.09MB 41.6s
=> => extracting sha256:e3143549f2b8b3ad8d79efdc47824641c6771796b3770f3c637a38aabd2b3462 0.8s
=> => sha256:2fbf02c04055ee28543076a502ce51fa41895e386e4462739fb1526dc371425e 29.23MB / 29.23MB 62.7s
=> => sha256:149075fcccc1b52406789bff7063813f1c14baf10b165df0e1e00fcb0ca415e5 248B / 248B 63.5s
=> => extracting sha256:72e8e93b0d018b135d833207c6feaba205653ac52e0a80d214477ec0de239dee 2.6s
=> => extracting sha256:23d55a674b2e3641a37b7f2053f00af6fe4f2a89576ffb917d4b7b4deaf24591 8.0s
=> => extracting sha256:8f5fb10be9d4525dc6941af4e2a587d1d3edbbb7442cdad0eebb1a3806b183dd 0.3s
=> => extracting sha256:2fbf02c04055ee28543076a502ce51fa41895e386e4462739fb1526dc371425e 1.0s
=> => extracting sha256:149075fcccc1b52406789bff7063813f1c14baf10b165df0e1e00fcb0ca415e5 0.0s
```

step6: run the container
sudo docker run -p 5000:5000 p1

```
1rv24mc074_poornimat@lenovo: /home/1rv24mc074_poornimat/Program.3$ sudo docker run -p 5000:5000 p3
* 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
□
```

step7:verify the application
open your browser and go to <http://localhost:5000>

