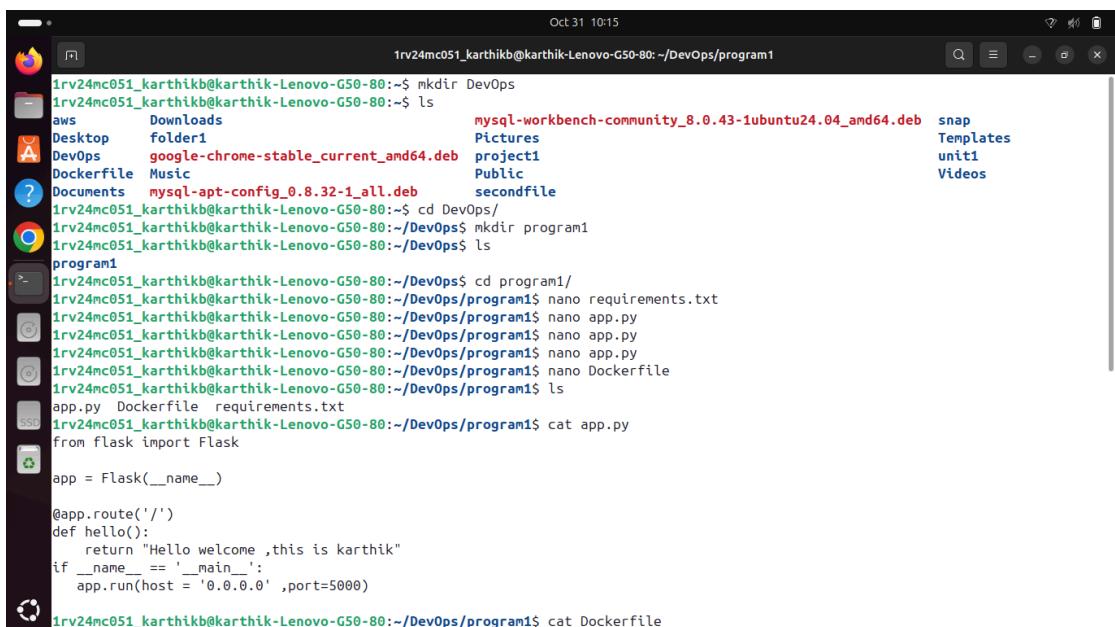


# Program 1

## Build a docker container from a custom Dockerfile

Step1:Created a folder named "program1"

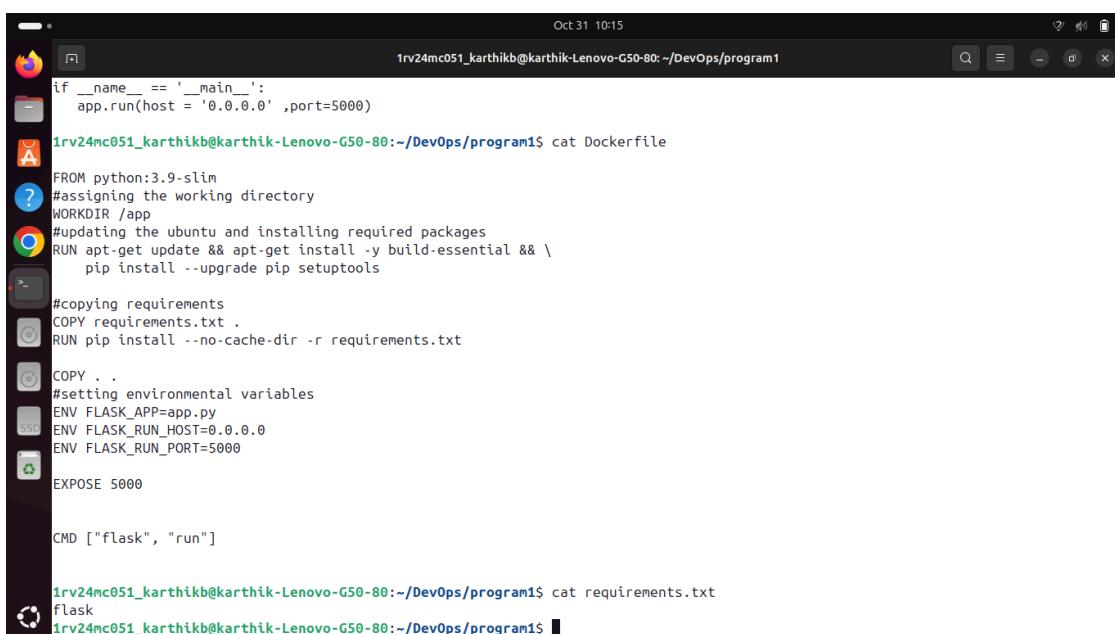
Step 2: Created 3 files and wrote the necessary code in requirements.txt, app.py, and Dockerfile



```
Oct 31 10:15
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~$ mkdir DevOps
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~$ ls
aws      Downloads          mysql-workbench-community_8.0.43-1ubuntu24.04_amd64.deb    snap
Desktop   folder1           Pictures             Templates
DevOps    google-chrome-stable_current_amd64.deb  project1        unit1
Dockerfile Music            Public              Videos
Documents  mysql-apt-config_0.8.32-1_all.deb    secondfile
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~$ cd DevOps/
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps$ mkdir program1
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps$ ls
program1
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps$ cd program1/
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1$ nano requirements.txt
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1$ nano app.py
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1$ nano Dockerfile
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1$ ls
app.py  Dockerfile  requirements.txt
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1$ cat app.py
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello():
    return "Hello welcome ,this is karthik"
if __name__ == '__main__':
    app.run(host = '0.0.0.0' ,port=5000)
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1$ cat Dockerfile
```

```
Oct 31 10:15
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~$ cat Dockerfile
if __name__ == '__main__':
    app.run(host = '0.0.0.0' ,port=5000)

1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1$ cat Dockerfile
FROM python:3.9-slim
#assigning the working directory
WORKDIR /app
#updating the ubuntu and installing required packages
RUN apt-get update && apt-get install -y build-essential && \
    pip install --upgrade pip setuptools

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

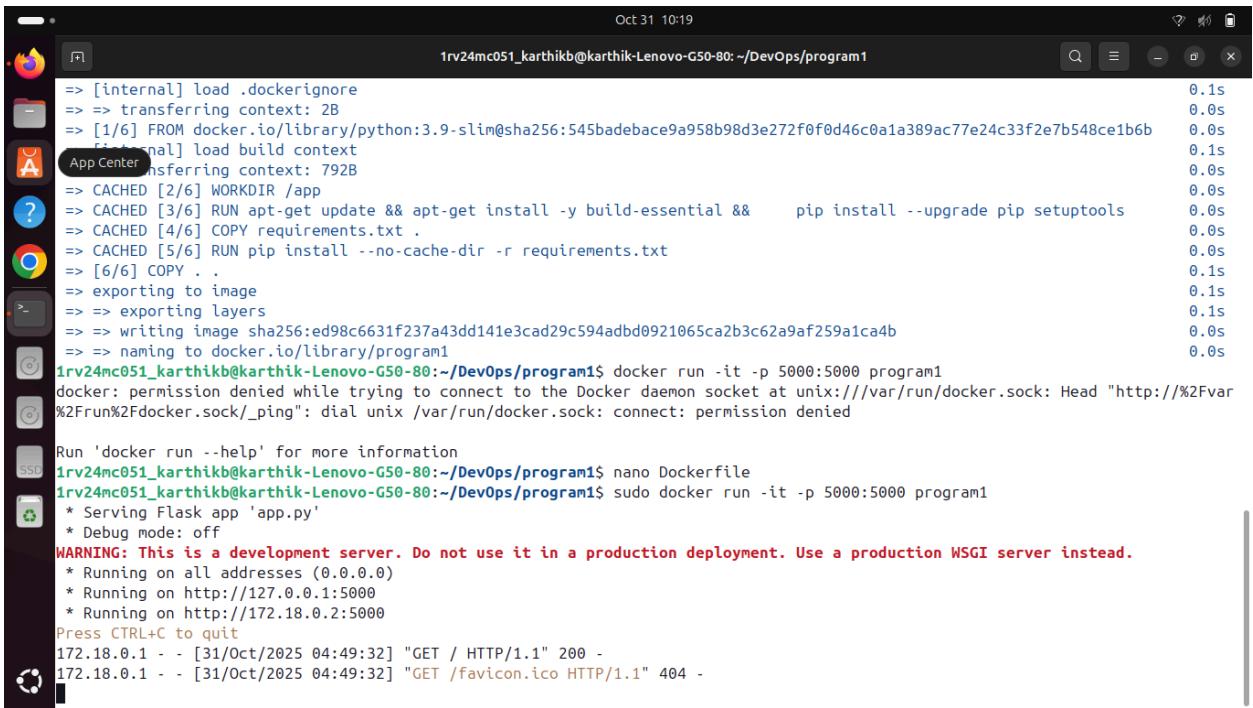
#setting environmental variables
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
ENV FLASK_RUN_PORT=5000

EXPOSE 5000

CMD ["flask", "run"]

1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1$ cat requirements.txt
Flask
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1$
```

Step3: Now Built the Docker image using  
sudo docker build -t program1 .



```
Oct 31 10:19
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/6] FROM docker.io/library/python:3.9-slim@sha256:545badebace9a958b98d3e272f0f0d46c0a1a389ac77e24c33f2e7b548ce1b6b
=> [internal] load build context
=> App Center transferring context: 792B
=> [2/6] WORKDIR /app
=> [3/6] RUN apt-get update && apt-get install -y build-essential &&     pip install --upgrade pip setuptools
=> [4/6] COPY requirements.txt .
=> [5/6] RUN pip install --no-cache-dir -r requirements.txt
=> [6/6] COPY . .
=> exporting to image
=> => exporting layers
=> => writing image sha256:ed98c6631f237a43dd141e3cad29c594adbd0921065ca2b3c62a9af259a1ca4b
=> => naming to docker.io/library/program1
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1$ docker run -it -p 5000:5000 program1
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied

Run 'docker run --help' for more information
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1$ nano Dockerfile
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~/DevOps/program1$ sudo docker run -it -p 5000:5000 program1
 * 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 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 04:49:32] "GET / HTTP/1.1" 200 -
172.18.0.1 - - [31/Oct/2025 04:49:32] "GET /favicon.ico HTTP/1.1" 404 -
```

Step 4: Open the application in the server using

