

Name: Jayesh Kumar R

USN: 1RV24MC048

Program 1: Build a Docker Container from a Custom Dockerfile

Project Structure:

Program-1/

-----> Dockerfile

-----> requirements.txt

-----> app.py

STEP 1: Create a **Dockerfile** and add the following content.

```
flask1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab1$ cat Dockerfile
FROM python:3.9-slim

WORKDIR /app

LABEL maintainer="jayeshkumarr"

COPY requirements.txt .

RUN pip install -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"]1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab1$
```

STEP 2: Create a Python file -**app.py**

```
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab1$ cat app.py
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "Hello Deepkia Ma'am"

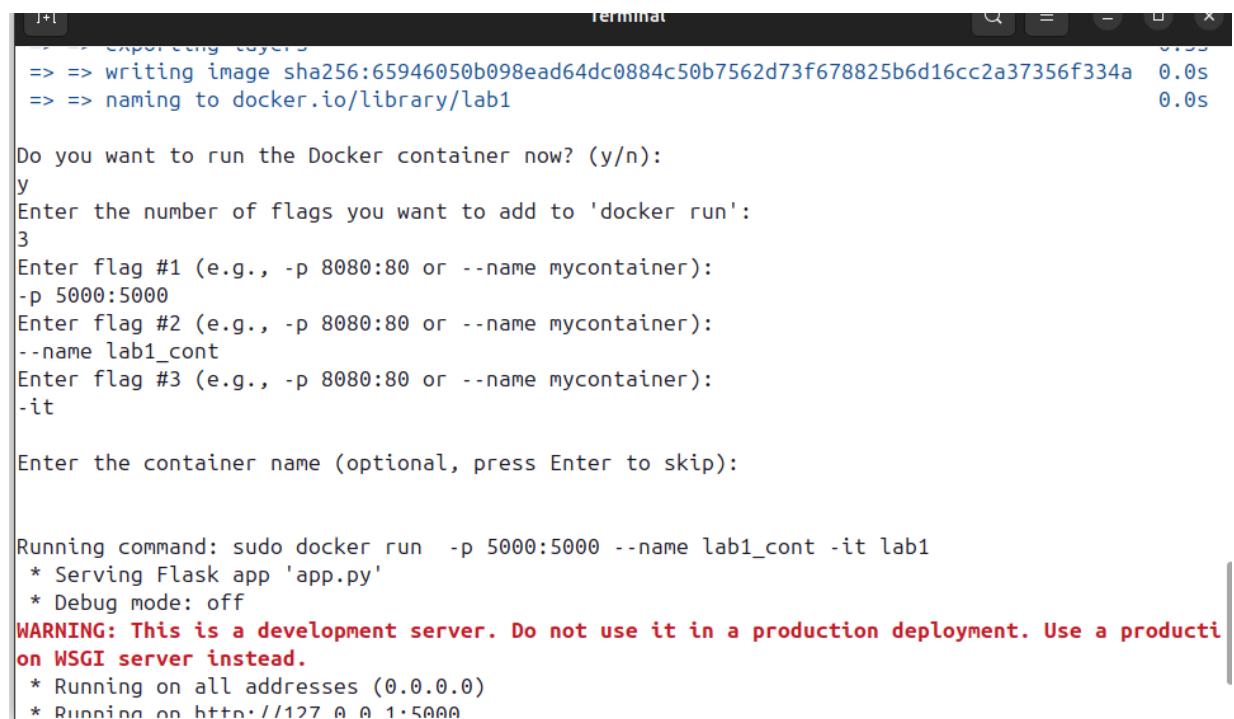
if __name__ == '__main__':
    app.run(host="0.0.0.0", port=5000)
```

STEP 3: Create a simple text file – **requirements.txt** & add the following

```
flask
```

STEP 4: Execute the Docker Build command on the Terminal to build the Docker image and then execute the Docker run command specifying the Port Number for the container.

```
1rv24mc048_Jayesh@Jayesh:~/Desktop/DEVOPS_Lab/lab1$ dockerscript
Enter the Name of the Image you want to create:
lab1
Image lab1 does not exist. Building it now...
[+] Building 13.7s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 286B
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd338d3060f261f53f14
=> => resolve docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd338d3060f261f53f14
=> => sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3d 1.29MB / 1.29MB 0.9s
=> => sha256:fc74430849022d13b0d44b8969a953f842f59c6e9d1a0c2c83d710af 13.88MB / 13.88MB 1.9s
=> => sha256:ea56f685404adf81680322f152d2cfec62115b30dda481c2c450078315beb5 251B / 251B 1.3s
=> => sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13c 10.36kB / 10.36kB 0.0s
=> => sha256:dad5b29e3506c35e0fd222736f4d4ef25d21b219acdd73f7bb41d59996 1.74kB / 1.74kB 0.0s
=> => sha256:085da638e1b8a449514c3fda83ff50a3bffaee4418b050cfacd87e57220 5.40kB / 5.40kB 0.0s
=> => extracting sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881 0.1s
=> => extracting sha256:fc74430849022d13b0d44b8969a953f842f59c6e9d1a0c2c83d710affa286c0 0.9s
=> => extracting sha256:ea56f685404adf81680322f152d2cfec62115b30dda481c2c450078315beb50 0.0s
=> [internal] load build context
=> => transferring context: 93B
```



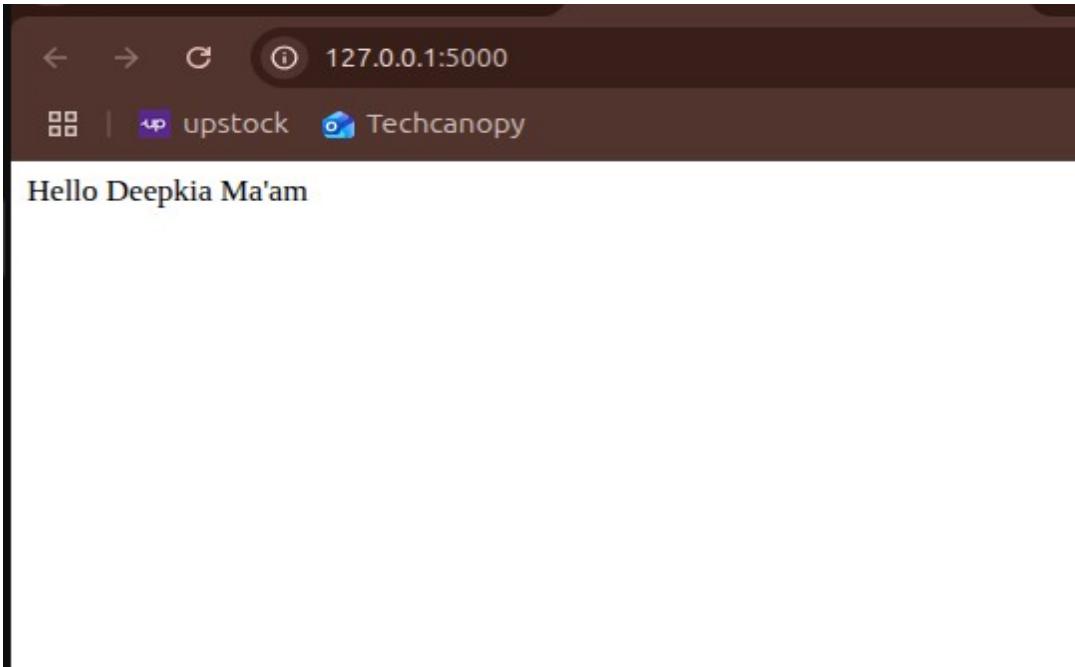
```
J+1 Terminal
--> Exporting layers
=> => writing image sha256:65946050b098ead64dc0884c50b7562d73f678825b6d16cc2a37356f334a 0.0s
=> => naming to docker.io/library/lab1 0.0s

Do you want to run the Docker container now? (y/n):
y
Enter the number of flags you want to add to 'docker run':
3
Enter flag #1 (e.g., -p 8080:80 or --name mycontainer):
-p 5000:5000
Enter flag #2 (e.g., -p 8080:80 or --name mycontainer):
--name lab1_cont
Enter flag #3 (e.g., -p 8080:80 or --name mycontainer):
-it

Enter the container name (optional, press Enter to skip):

Running command: sudo docker run -p 5000:5000 --name lab1_cont -it lab1
 * Serving Flask app 'app.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a producti
on WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
```

STEP 5: Test the Container by verifying the localhost details on web browser, the text Hello Docker! Will be displayed.



STEP 6: Stop the container with the container number, remove the docker container and delete the Docker Image with the name program-1

```
sudo docker container stop lab1_cont
```

```
sudo docker container rm lab1_cont
```

```
sudo docker image rm lab1
```

Dockerscript-to build and run an image

```
echo "Enter the Name of the Image you want to create: "
read -r image

# Check if image exists
if ! docker images | grep -q "$image"; then
    echo "Image $image does not exist. Building it now..."
    sudo docker build -t "$image" .
else
    echo "Image $image already exists."
    echo "Re-running the script..."
fi

echo
echo "Do you want to run the Docker container now? (y/n): "
read -r run_choice

if [[ "$run_choice" =~ ^[Yy]$ ]]; then
    echo "Enter the number of flags you want to add to 'docker run': "
    read -r num_flags

    flags=""

    for ((i=1; i<=num_flags; i++)); do
        echo "Enter flag #\$i (e.g., -p 8080:80 or --name mycontainer): "
        read -r flag
        flags="\$flags \$flag"
    done

    echo
    echo "Enter the container name (optional, press Enter to skip): "
    read -r container_name

    if [[ -n "$container_name" ]]; then
        run_cmd="sudo docker run \$flags --name \$container_name \$image"
    else
        run_cmd="sudo docker run \$flags \$image"
    fi

    echo
    echo "Running command: \$run_cmd"
    eval "\$run_cmd"
else
    echo "Exiting without running container."
fi
```