

Lab Program 1:

Build a docker container from custom dockerfile

Name: Harshavardhan B R

USN: 1RV24MC042

Steps:

Step 1: Create the following files in the project directory

> mkdir lab1

> cd lab1

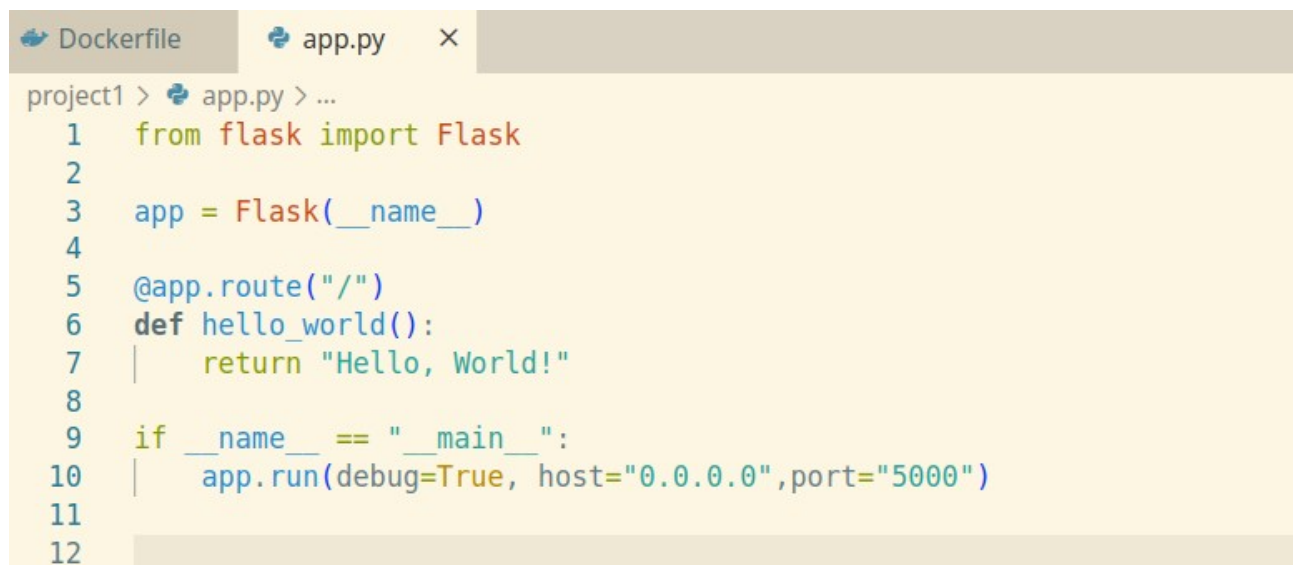
> touch Dockerfile app.py requirements.txt

Expected project structure

```
• > tree project1
project1
├── app.py
├── Dockerfile
└── requirements.txt
```

Step 2: Add the content to the created files

App.py

A screenshot of a code editor with two tabs: 'Dockerfile' and 'app.py'. The 'app.py' tab is active, showing a Python script for a Flask web application. The script imports Flask, creates an app, defines a route for the root path, and runs the app on port 5000. The code is as follows:

```
project1 > app.py > ...
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  @app.route("/")
6  def hello_world():
7      return "Hello, World!"
8
9  if __name__ == "__main__":
10     app.run(debug=True, host="0.0.0.0", port="5000")
11
12
```

requirements.txt

Flask

Dockerfile

```
Dockerfile X
project1 > Dockerfile > ...
1 FROM python:3.9-slim
2 WORKDIR /app
3 COPY /requirements.txt .
4 RUN pip install --no-cache-dir -r requirements.txt
5 COPY . .
6 ENV ENVHOST="0.0.0.0"
7 ENV ENVPORT=5000
8 EXPOSE 5000
9 CMD [ "python", "app.py" ]
10
11
```

Step 3: Build the image

```
• > docker build . -t harsha/lab1
[+] Building 2.3s (11/11) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 238B                                0.0s
=> [internal] load metadata for docker.io/library/python:3.9-slim 2.2s
=> [auth] library/python:pull token for registry-1.docker.io      0.0s
=> [internal] load .dockerignore                                   0.0s
=> => transferring context: 2B                                       0.0s
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:545badebace9a9 0.0s
=> [internal] load build context                                   0.0s
=> => transferring context: 272B                                     0.0s
=> CACHED [2/5] WORKDIR /app                                       0.0s
=> CACHED [3/5] COPY /requirements.txt .                           0.0s
=> CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt 0.0s
=> [5/5] COPY . .                                                 0.0s
=> exporting to image                                             0.0s
=> => exporting layers                                             0.0s
=> => writing image sha256:54e3369d0d5c88a3773a9cc70399df2ea71423a049 0.0s
=> => naming to docker.io/harsha/lab1                             0.0s
```

Step 4: Run the container

```
~ /Desktop/RVCE/devopslab/project1
> docker run -p 5000:5000 harsha/lab1
* 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 http://127.0.0.1:5000
Press CTRL+C to quit
```

Step 5: Check if the application is running

```
> curl -X GET http://127.0.0.1:5000
Hello, World!%
```