Lab 4 - Dockerfile

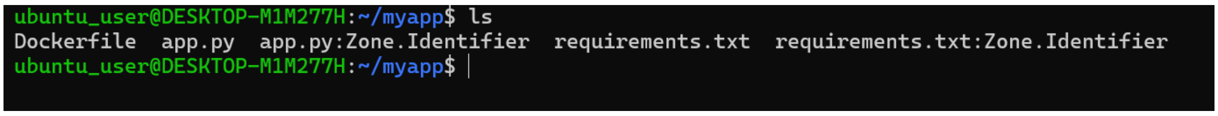
Create your docker image

Create a dockerfile for a Flask application (python).

Tips

Create your dockerfile

1. There are two file
   1. `requirement.txt`, contain python dependencies
   2. `app.py`, contain our flask app that listen on port `9090`
2. Create a new directory named `myapp`
3. Copy `requirement.txt` and `app.py` in `myapp`
4. Run `cd myapp`
5. Create a file name `Dockerfile`



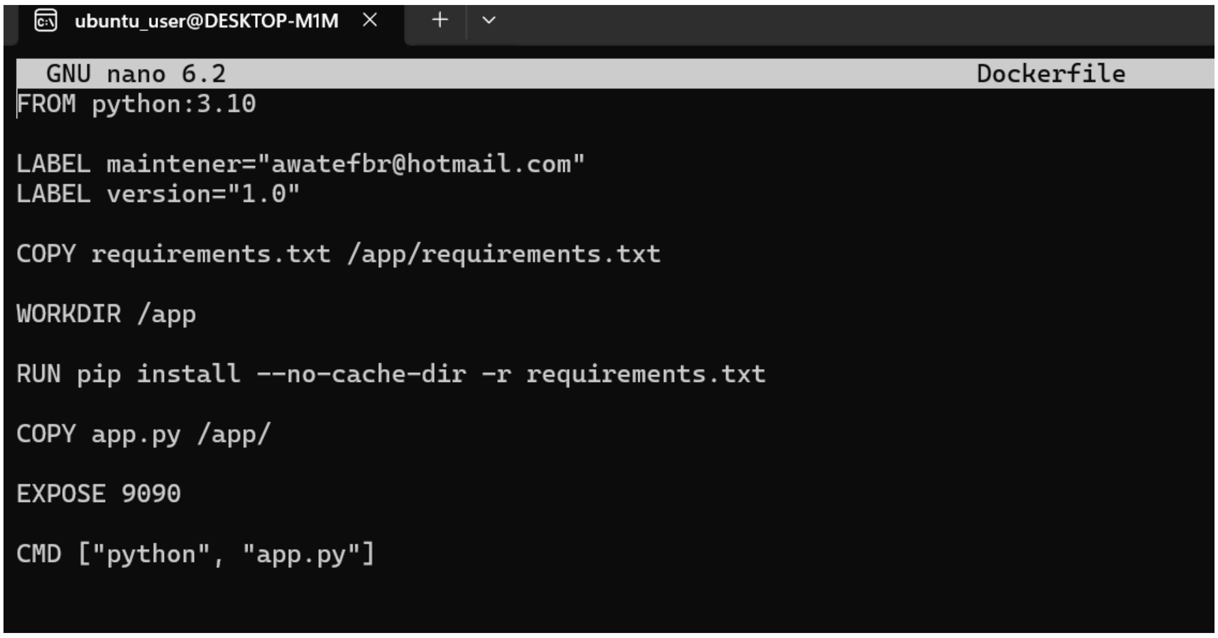
```
ubuntu_user@DESKTOP-M1M277H:~/myapp$ ls
Dockerfile  app.py  app.py:Zone.Identifier  requirements.txt  requirements.txt:Zone.Identifier
ubuntu_user@DESKTOP-M1M277H:~/myapp$ |
```

Modify the dockerfile

1. Use a python image as base
2. Copy `requirement.txt` in `/app/requirements.txt`
3. Define `/app` as working directory
4. Install python dependencies using `pip install -r <file>`
5. Copy `app.py` inside `/app`
6. Specify that the container use the port `9090`

7. Specify the maintainer and the version of the dockerfile

8. Make sure the container will run the command ` python app.py`



```
  GNU nano 6.2                                                  Dockerfile
FROM python:3.10

LABEL maintener="awatefbr@hotmail.com"
LABEL version="1.0"

COPY requirements.txt /app/requirements.txt

WORKDIR /app

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

COPY app.py /app/

EXPOSE 9090

CMD ["python", "app.py"]
```

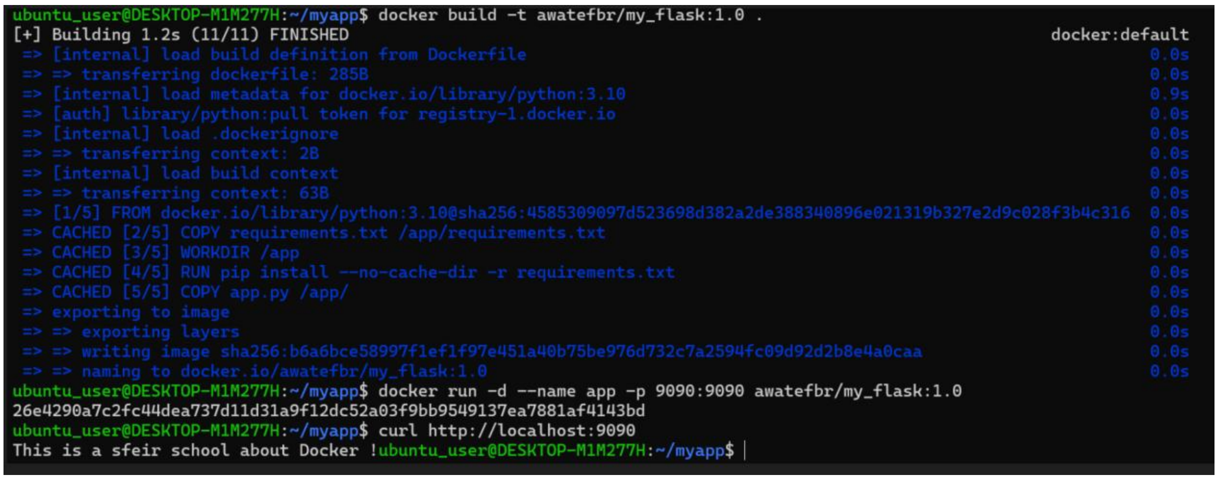
Build the image

1. Build the docker image and name it <dockerHubId>/my_flask:1.0
2. Push it to the docker hub

Run it

1. Run your application as ` app`

2. curl localhost:9090



```
ubuntu_user@DESKTOP-M1M277H:~/myapp$ docker build -t awatefbr/my_flask:1.0 .
[+] Building 1.2s (11/11) FINISHED                                                                 docker:default
 => [internal] load build definition from Dockerfile                                                          0.0s
 => => transferring dockerfile: 285B                                                                          0.0s
 => [internal] load metadata for docker.io/library/python:3.10                                                0.9s
 => [auth] library/python:pull token for registry-1.docker.io                                                 0.0s
 => [internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                               0.0s
 => [internal] load build context                                                                             0.0s
 => => transferring context: 63B                                                                              0.0s
 => [1/5] FROM docker.io/library/python:3.10@sha256:4585309097d523698d382a2de388340896e021319b327e2d9c028f3b4c316  0.0s
 => CACHED [2/5] COPY requirements.txt /app/requirements.txt                                                  0.0s
 => CACHED [3/5] WORKDIR /app                                                                                 0.0s
 => CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt                                           0.0s
 => CACHED [5/5] COPY app.py /app/                                                                            0.0s
 => exporting to image                                                                                        0.0s
 => => exporting layers                                                                                       0.0s
 => => writing image sha256:b6a6bce58997f1ef1f97e451a40b75be976d732c7a2594fc09d92d2b8e4a0caa                  0.0s
 => => naming to docker.io/awatefbr/my_flask:1.0                                                              0.0s
ubuntu_user@DESKTOP-M1M277H:~/myapp$ docker run -d --name app -p 9090:9090 awatefbr/my_flask:1.0
26e4290a7c2fc44dea737d11d31a9f12dc52a03f9bb9549137ea7881af4143bd
ubuntu_user@DESKTOP-M1M277H:~/myapp$ curl http://localhost:9090
This is a sfeir school about Docker !ubuntu_user@DESKTOP-M1M277H:~/myapp$
```