Date: 18-02-2026
Example

From Python
1. app.py
2. Requirements.txt
3. dockerfile

docker build -t python .
docker images
docker run -p 5002:5000 <image name>

Then to check: localhost:5002

To Run this example we will take Python as a base layer, for this we will choose “python:3.11-slim” or according to your requirement you can choose.

We have three files:
1. app.py → This is where our python code is written and stored, I’ve used flask for server side

```
from flask import Flask, jsonify
app = Flask(__name__)
@app.route("/")
def home():
    return jsonify({
        "message": "Hello from Flask inside Docker 🚀",
        "status": "success"
    })
@app.route("/health")
def health():
    return jsonify({
        "health": "ok"
    })
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

2. Requirements.txt → is a file that contains a list of all Python dependencies required for the project. It ensures that the same versions of libraries are installed inside the Docker container.

```
Flask==3.0.0
```

3. dockerfile → is a text-based document instruction for docker to make image.

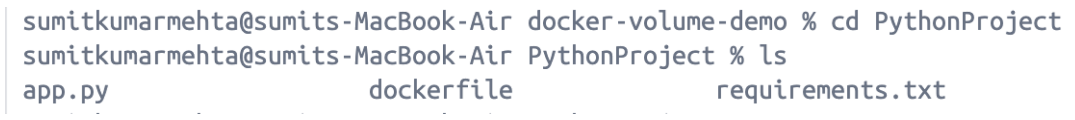
```

FROM python:3.11-slim
```

```
WORKDIR /app
COPY requirements.txt .
RUN pip install -r  requirements.txt
COPY . .
EXPOSE 5000
CMD ["python","app.py"]
```

Now after making dockerfile go to the directory where dockerfile is stored. Using:

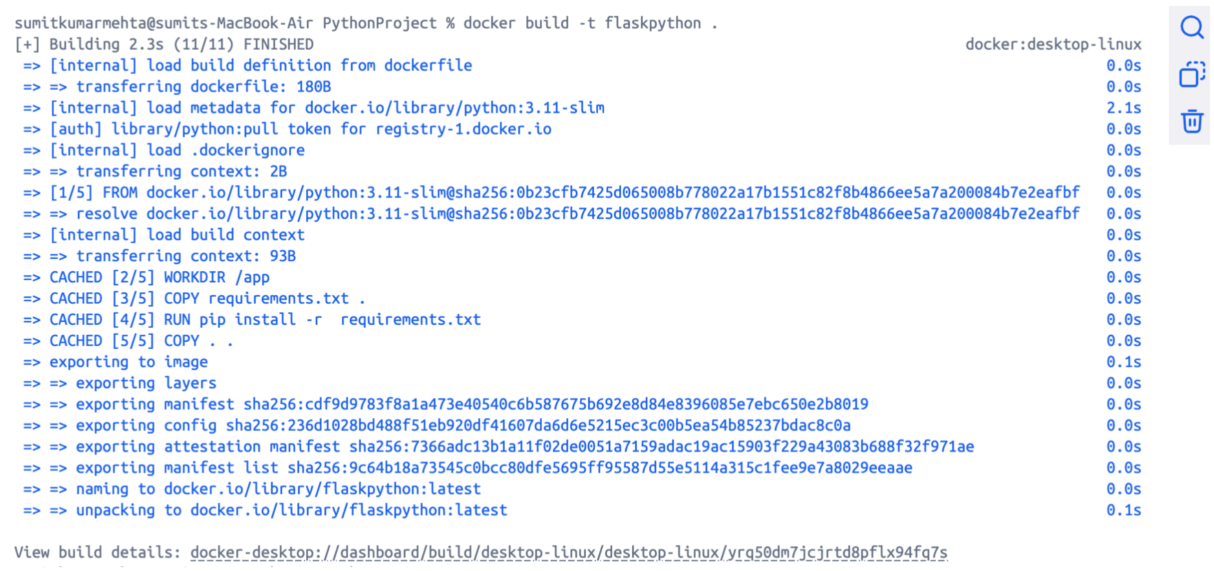
```
cd <path>
```



```
sumitkumarmehta@sumits-MacBook-Air docker-volume-demo % cd PythonProject
sumitkumarmehta@sumits-MacBook-Air PythonProject % ls
app.py                    dockerfile                requirements.txt
```

Now we will build image using:

```
docker build -t <image_name> .
```

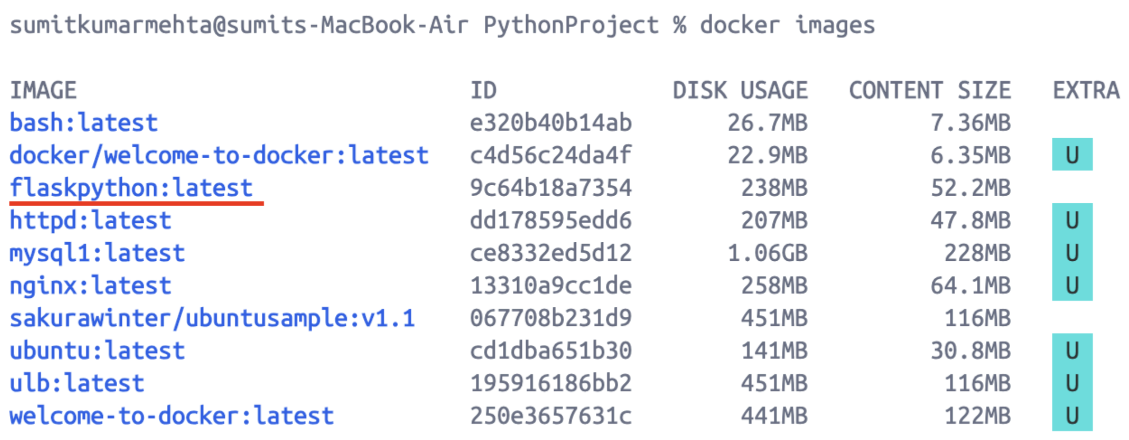


```
sumitkumarmehta@sumits-MacBook-Air PythonProject % docker build -t flaskpython .
[+] Building 2.3s (11/11) FINISHED                                                                docker:desktop-linux
 => [internal] load build definition from dockerfile                                                            0.0s
 => => transferring dockerfile: 180B                                                                            0.0s
 => [internal] load metadata for docker.io/library/python:3.11-slim                                             2.1s
 => [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:3.11-slim@sha256:0b23cfb7425d065008b778022a17b1551c82f8b4866ee5a7a200084b7e2eafbf  0.0s
 => => resolve docker.io/library/python:3.11-slim@sha256:0b23cfb7425d065008b778022a17b1551c82f8b4866ee5a7a200084b7e2eafbf  0.0s
 => [internal] load build context                                                                               0.0s
 => => transferring context: 93B                                                                                0.0s
 => CACHED [2/5] WORKDIR /app                                                                                   0.0s
 => CACHED [3/5] COPY requirements.txt .                                                                        0.0s
 => CACHED [4/5] RUN pip install -r  requirements.txt                                                           0.0s
 => CACHED [5/5] COPY . .                                                                                       0.0s
 => exporting to image                                                                                          0.1s
 => => exporting layers                                                                                         0.0s
 => => exporting manifest sha256:cdf9d9783f8a1a473e40540c6b587675b692e8d84e8396085e7ebc650e2b8019               0.0s
 => => exporting config sha256:236d1028bd488f51eb920df41607da6d6e5215ec3c00b5ea54b85237bdac8c0a                 0.0s
 => => exporting attestation manifest sha256:7366adc13b1a11f02de0051a7159adac19ac15903f229a43083b688f32f971ae   0.0s
 => => exporting manifest list sha256:9c64b18a73545c0bcc80dfe5695ff95587d55e5114a315c1fee9e7a8029eeaae          0.0s
 => => naming to docker.io/library/flaskpython:latest                                                           0.0s
 => => unpacking to docker.io/library/flaskpython:latest                                                        0.1s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/yrq50dm7jcjrtd8pflx94fq7s
```

To check built image:

```
docker images
```

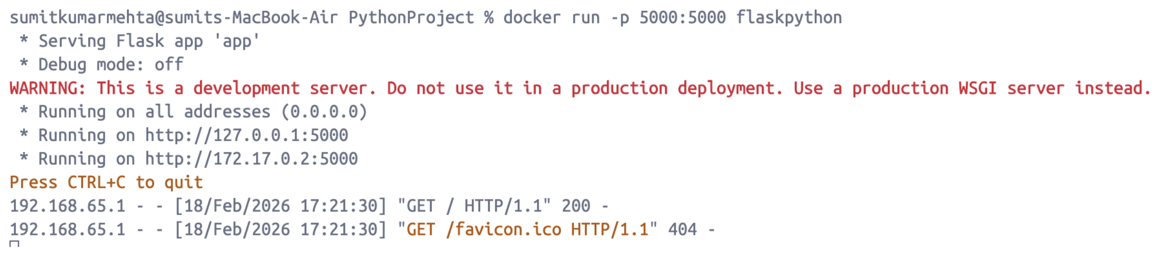


```
sumitkumarmehta@sumits-MacBook-Air PythonProject % docker images

IMAGE                             ID             DISK USAGE   CONTENT SIZE   EXTRA
bash:latest                       e320b40b14ab       26.7MB         7.36MB
docker/welcome-to-docker:latest   c4d56c24da4f       22.9MB         6.35MB    U
flaskpython:latest                9c64b18a7354        238MB         52.2MB
httpd:latest                      dd178595edd6        207MB         47.8MB    U
mysql1:latest                     ce8332ed5d12       1.06GB          228MB    U
nginx:latest                      13310a9cc1de        258MB         64.1MB    U
sakurawinter/ubuntusample:v1.1    067708b231d9        451MB          116MB
ubuntu:latest                     cd1dba651b30        141MB         30.8MB    U
ulb:latest                        195916186bb2        451MB          116MB    U
welcome-to-docker:latest          250e3657631c        441MB          122MB    U
```

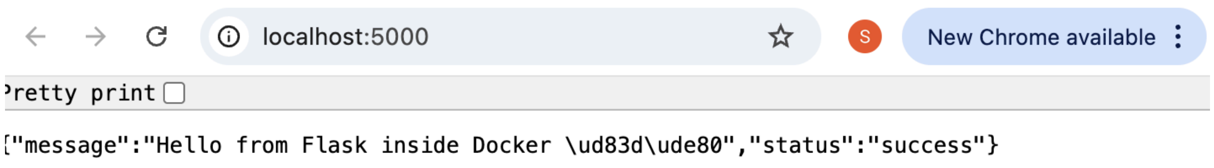
Result: Now run using:

```
docker run -p <host_port>:<container_port> <image_name>
```



```
sumitkumarmehta@sumits-MacBook-Air PythonProject % docker run -p 5000:5000 flaskpython
 * 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://172.17.0.2:5000
Press CTRL+C to quit
192.168.65.1 - - [18/Feb/2026 17:21:30] "GET / HTTP/1.1" 200 -
192.168.65.1 - - [18/Feb/2026 17:21:30] "GET /favicon.ico HTTP/1.1" 404 -
```

Now go to localhost:<host_port>:



```
{"message":"Hello from Flask inside Docker \ud83d\ude80","status":"success"}
```