

- Create a Docker directory, where Docker Images can be created for each application.
- Build Docker Image.
- Executing python file in that Docker image.

```

(base) rocks-Air:flask_doctor_demo_1 rock$ pwd
/Users/rock/Data_Science/Docker/flask_doctor_demo_1
(base) rocks-Air:flask_doctor_demo_1 rock$ docker build -t python_docker_file1 -f "./python_docker_file1" .
Sending build context to Docker daemon 78.34kB
Step 1/5 : FROM python:latest
--> 7f5b6ccd03e9
Step 2/5 : MAINTAINER Rakesh Example 1
--> Running in ad009708bd4a
Removing intermediate container ad009708bd4a
--> b71667c3b197
Step 3/5 : ADD index.py /src/index.py
--> d617ceedd43c
Step 4/5 : WORKDIR /src
--> Running in 47e999d5caaa
Removing intermediate container 47e999d5caaa
--> 0dadda84613f
Step 5/5 : CMD ["python", "/src/index.py"]
--> Running in 7b40b1b79068
Removing intermediate container 7b40b1b79068
--> 057158c7c7f7
Successfully built 057158c7c7f7
Successfully tagged python_docker_file1:latest
(base) rocks-Air:flask_doctor_demo_1 rock$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
python_docker_file1  latest             057158c7c7f7       14 seconds ago     934MB
python_flask_rest_api_2  latest             4751bf9dcbab       5 months ago       943MB
python_flask_rest_api_1  latest             df4f1ccb2937       5 months ago       943MB
python                latest             7f5b6ccd03e9       6 months ago       934MB
(base) rocks-Air:flask_doctor_demo_1 rock$ docker run python_docker_file1
Hello from python file
(base) rocks-Air:flask_doctor_demo_1 rock$

```

- Docker First Image build/setup file(i.e., which creates Docker Image).

```

python_docker_file1
1 FROM python:latest
2 MAINTAINER Rakesh Example 1
3 ADD index.py /src/index.py
4 WORKDIR /src
5 CMD ["python", "/src/index.py"]

```

- Python File which run in the Docker Image.

```
index.py X
1 print("Hello from python file");
```

- Showing the Docker images and removing the unwanted previous docker images.

```
flask_doctor_demo_1 — -bash — 184x50
(base) rocks-Air:flask_doctor_demo_1 rock$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
python_docker_file1  latest             057158c7c7f7       8 minutes ago      934MB
python_flask_rest_api_2  latest            4751bf9dcbab       5 months ago       943MB
python_flask_rest_api_1  latest            df4f1ccb2937       5 months ago       943MB
python               latest            7f5b6ccd03e9       6 months ago       934MB
(base) rocks-Air:flask_doctor_demo_1 rock$ docker image rm python -f
Untagged: python:latest
Untagged: python@sha256:dd6cd8191ccbc2a6af5d0ddb51e6057c144df14e14bafd5c7b3ef78738050
(base) rocks-Air:flask_doctor_demo_1 rock$ docker image rm python_flask_rest_api_2 -f
Untagged: python_flask_rest_api_2:latest
Deleted: sha256:4751bf9dcbab7cc106b6137b34a99578ad3a71c9182e6e78ce6a0e9c0ab58041
Deleted: sha256:e47ba1958dca609c02655b432b7ac419a5e1c2ca7b194a8aa2e68b5cf8a6021f
Deleted: sha256:926688d3dd6c3768caa5d25c594b0fe72d319b335a707e26da32f399affb2990
(base) rocks-Air:flask_doctor_demo_1 rock$ docker image rm python_flask_rest_api_1 -f
Untagged: python_flask_rest_api_1:latest
Deleted: sha256:df4f1ccb2937631d96259fe16e7e5b5391b3a4f44896130263faa4f8e5114ba7
Deleted: sha256:c0ab5393d9942d9e50faa6739a8eece27cf77bd5488b4a3c080be244026880a7
Deleted: sha256:4a31813152b57c0c66d9c990d9528cd7f2b80b55d8a656ab726e1053b6fe819a
Deleted: sha256:9bc3674a5f29027622f2c1af2b57acff22627ee4213be20c06ec6613991dd701
Deleted: sha256:7fc81a6dd2568a4815330645b539234b447511ad715c2e17a17d745bdc8be83f
(base) rocks-Air:flask_doctor_demo_1 rock$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
python_docker_file1  latest             057158c7c7f7       9 minutes ago      934MB
(base) rocks-Air:flask_doctor_demo_1 rock$
```

- Creating 2nd Docker Image.

```

flask_doctor_demo_1 --- bash -- 194x55
(base) rocks-Air:flask_doctor_demo_1 rock$ docker build -t python_flask_rest_api_1 -f ./python_flask_rest_api_1 .
Sending build context to Docker daemon  78.34kB
Step 1/7 : FROM scratch
---->
Step 2/7 : FROM python:latest
latest: Pulling from library/python
6c33745f49b4: Pull complete
c87cd3c61e27: Pull complete
95a3c799ac37: Pull complete
a61c38f966ac: Pull complete
c2dd6d195b68: Pull complete
29b9446ae7bd: Pull complete
09cf96c794f9: Pull complete
f674fd97fba7: Pull complete
ffc24df6b7b8: Pull complete
Digest: sha256:b273b08cf9fe6b07ee4c2466895e5a4ac5381ade62106b7e7817eba0a684613
Status: Downloaded newer image for python:latest
----> 0611cf846c85
Step 3/7 : MAINTAINER : Rakesh Flask Example 2
----> Running in 30eaf029149f
Removing intermediate container 30eaf029149f
----> ec96db01d4a7
Step 4/7 : WORKDIR /src
----> Running in i3fde22708a9
Removing intermediate container 13fde22708a9
----> 775b04871c80
Step 5/7 : ADD Flask_Hello_World.py /src/Flask_Hello_World.py
----> 61a113676ba6
Step 6/7 : RUN pip install flask
----> Running in cc0f66435e68
Collecting flask
  Downloading flask-1.1.2-py2.py3-none-any.whl (94 kB)
Collecting click>=5.1
  Downloading click-7.1.2-py3-none-any.whl (82 kB)
Collecting itsdangerous>=0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2>=2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
Collecting MarkupSafe>=0.23
  Downloading MarkupSafe-1.1.1.tar.gz (19 kB)
Collecting Werkzeug>=0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
Building wheels for collected packages: MarkupSafe
  Building wheel for MarkupSafe (setup.py): started
  Building wheel for MarkupSafe (setup.py): finished with status 'done'
  Created wheel for MarkupSafe: filename=MarkupSafe-1.1.1-cp39-cp39-linux_x86_64.whl size=32233 sha256=1db6c85c4947da4035d6636a9fed51616efe87da0a416094304b7c73bf597641
  Stored in directory: /root/.cache/pip/wheels/e0/19/6f/6ba857621f50dc08e084312746ed3ebc14211ba30037d5e44e
Successfully built MarkupSafe
Installing collected packages: MarkupSafe, Werkzeug, Jinja2, itsdangerous, click, flask
Successfully installed Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 flask-1.1.2 itsdangerous-1.1.0
Removing intermediate container cc0f66435e68
----> 3d3687baa167
Step 7/7 : CMD ["python", "/src/Flask_Hello_World.py"]
----> Running in 01056c622b55
Removing intermediate container 01056c622b55

```

```

Removing intermediate container 01056c622b55
----> 0c2bf8306897
Successfully built 0c2bf8306897
Successfully tagged python_flask_rest_api_1:latest
(base) rocks-Air:flask_doctor_demo_1 rock$

```

```

(base) rocks-Air:flask_doctor_demo_1 rock$ docker run -p 8080:8080 python_flask_rest_api_1
* Serving Flask app "Flask_Hello_World" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 307-359-843

```

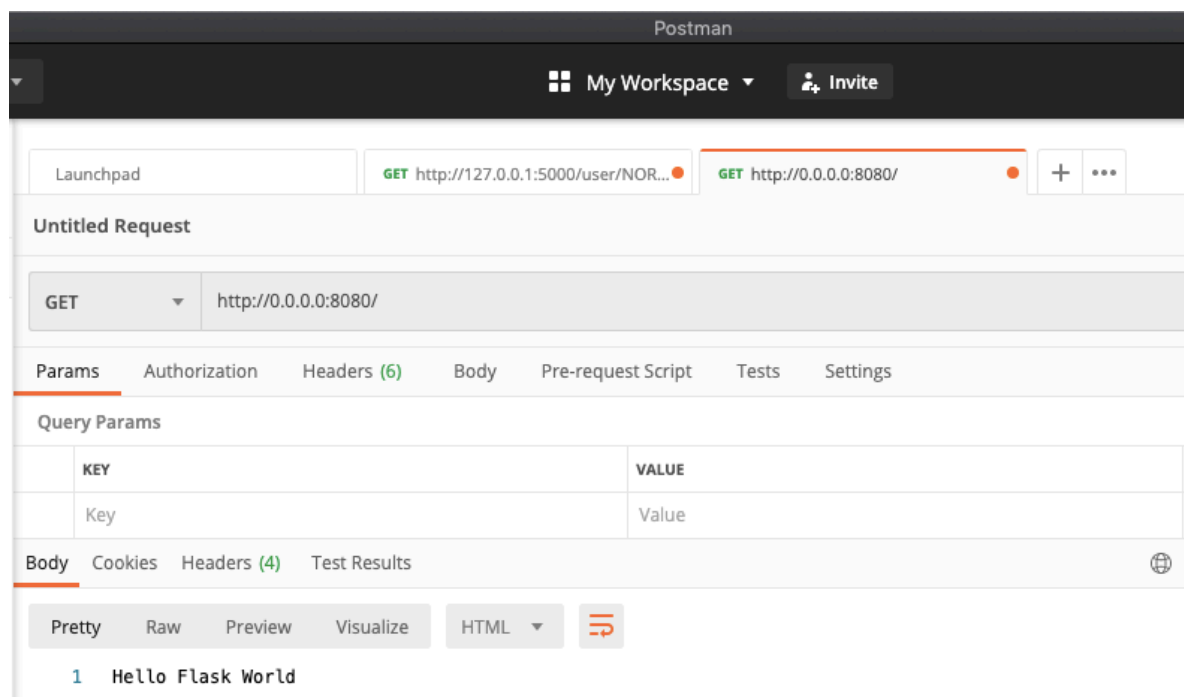
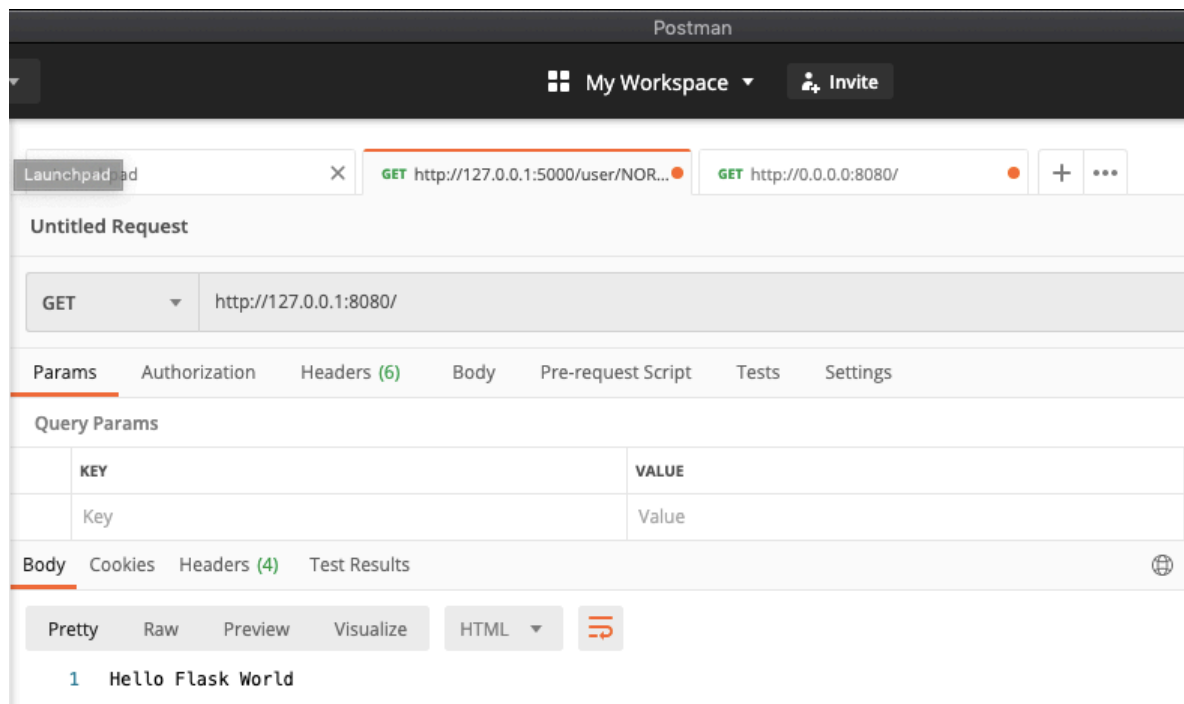
- Docker Second Image build/setup file(i.e., which creates Docker Image).

```
python_flask_rest_api_1 x
1 #Everytime Creates New Docker Image
2 FROM scratch
3
4 #Loading Python Environment To Execute
5 FROM python:latest
6
7 #Owner Of The Docker Image
8 MAINTAINER : Rakesh Flask Example 2
9
10 #Workdir path for execution
11 WORKDIR /src
12
13 #Adding executable files to work directory
14 ADD Flask_Hello_World.py /src/Flask_Hello_World.py
15
16 #Installing required libraries
17 RUN pip install flask
18
19
20 ##Installing Required Libraries - Generally will be mentioned in requirements.txt
21 #RUN mkdir -p /src
22 #RUN pip install --upgrade pip
23 #RUN pip install -r requirements.txt
24 #COPY Flask_Hello_World.py /src
25 #COPY requirements.txt /src
26
27
28 #Run app.py when the container launches
29 CMD ["python", "/src/Flask_Hello_World.py"]
```

- Python File which run in the Docker Image.

```
Flask_Hello_World.py x
1 from flask import Flask
2 app = Flask(__name__)
3
4 @app.route('/')
5 def hello_world():
6     return 'Hello Flask World'
7
8
9 if __name__ == '__main__':
10     #app.run()
11     app.run(host="0.0.0.0", debug=True, port=8080)
```

- Check Docker Flask API in Postman.



- We can see 2 Docker Images that we have created just now.

