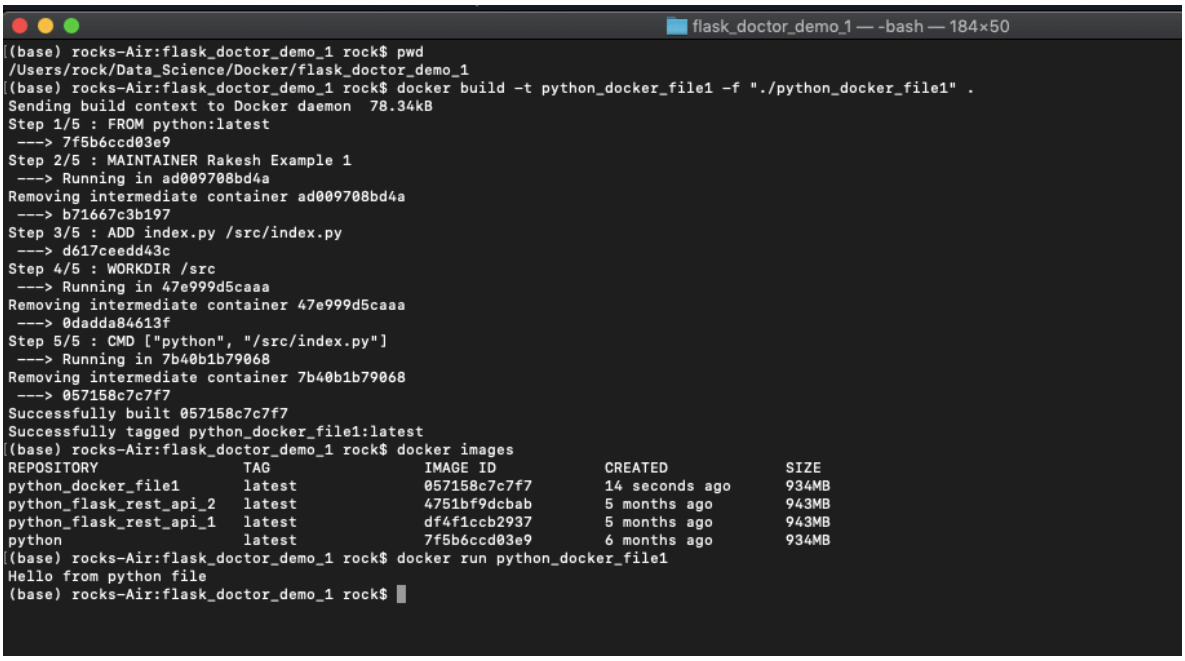
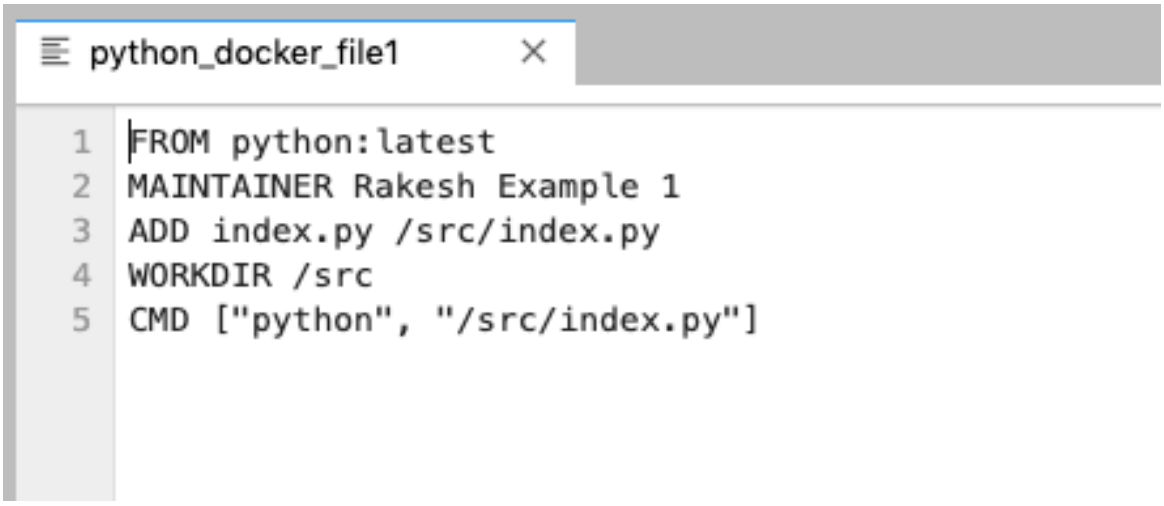
- Create a Docker directory, where Docker Images can be created for each application.
- Build Docker Image.
- Executing python file in that Docker image.



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

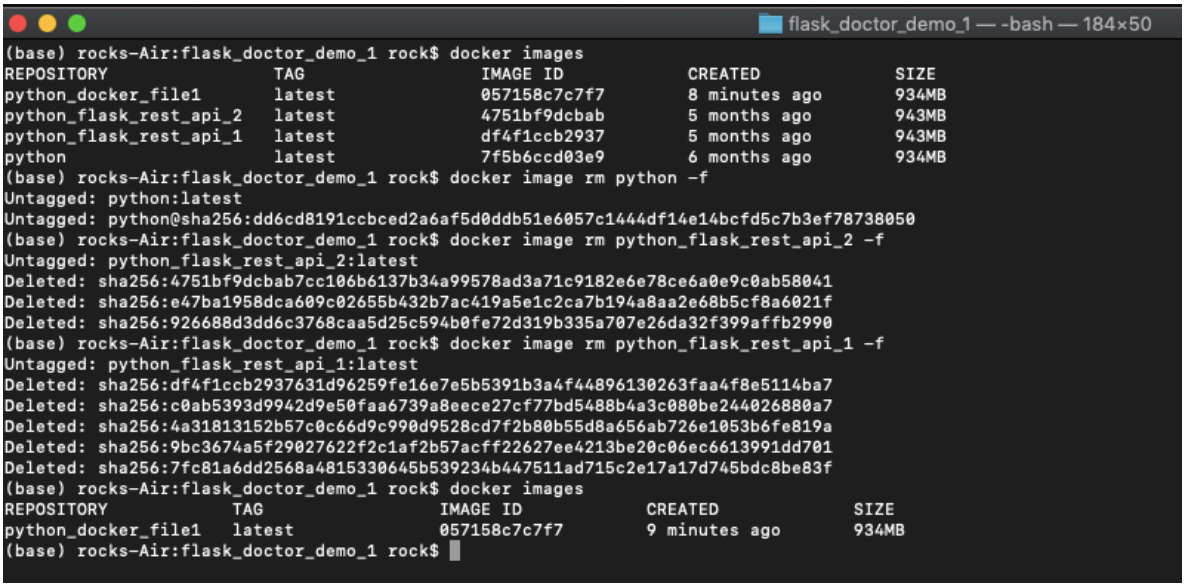


- Python File which run in the Docker Image.

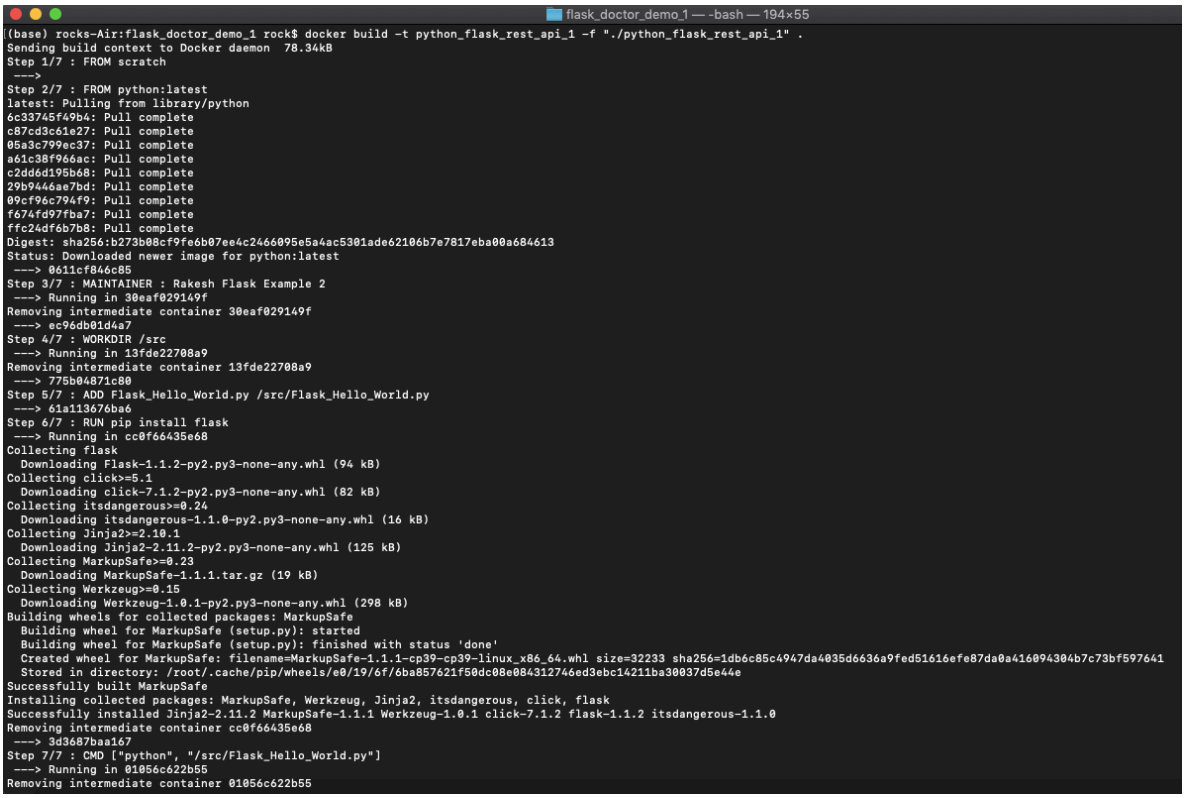
index.py

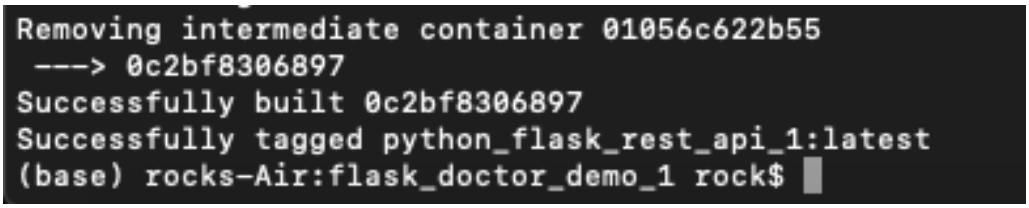
```
1 print("Hello from python file");
```

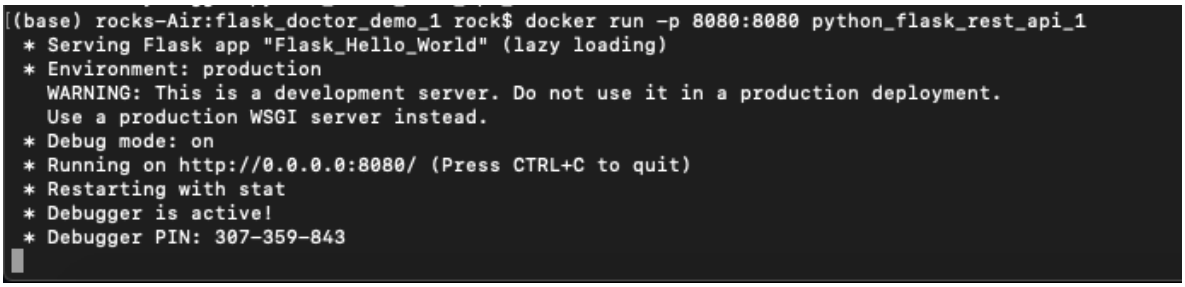
- Showing the Docker images and removing the unwanted previous docker images.



- Creating 2nd Docker Image.







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

python_flask_rest_api_1

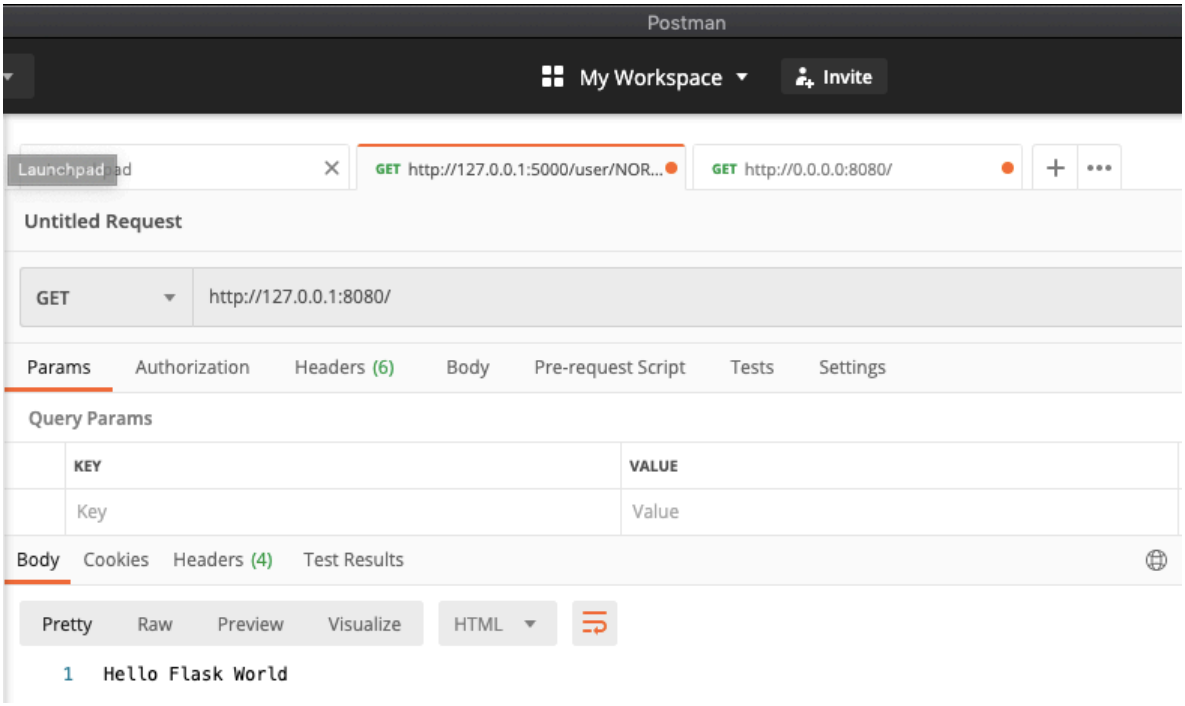
```
#Everytime Creates New Docker Image
FROM scratch

#Loading Python Environment To Execute
FROM python:latest

#Owner Of The Docker Image
MAINTAINER : Rakesh Flask Example 2

#Workdir path for execution
WORKDIR /src

#Adding executable files to work directory
ADD Flask_Hello_World.py /src/Flask_Hello_World.py

#Installing required libraries
RUN pip install flask


##Installing Required Libraries - Generally will be mentioned in requirements.txt
#RUN mkdir -p /src
#RUN pip install --upgrade pip
#RUN pip install -r requirements.txt
#COPY Flask_Hello_World.py /src
#COPY requirements.txt /src


#Run app.py when the container launches
CMD ["python", "/src/Flask_Hello_World.py"]
```

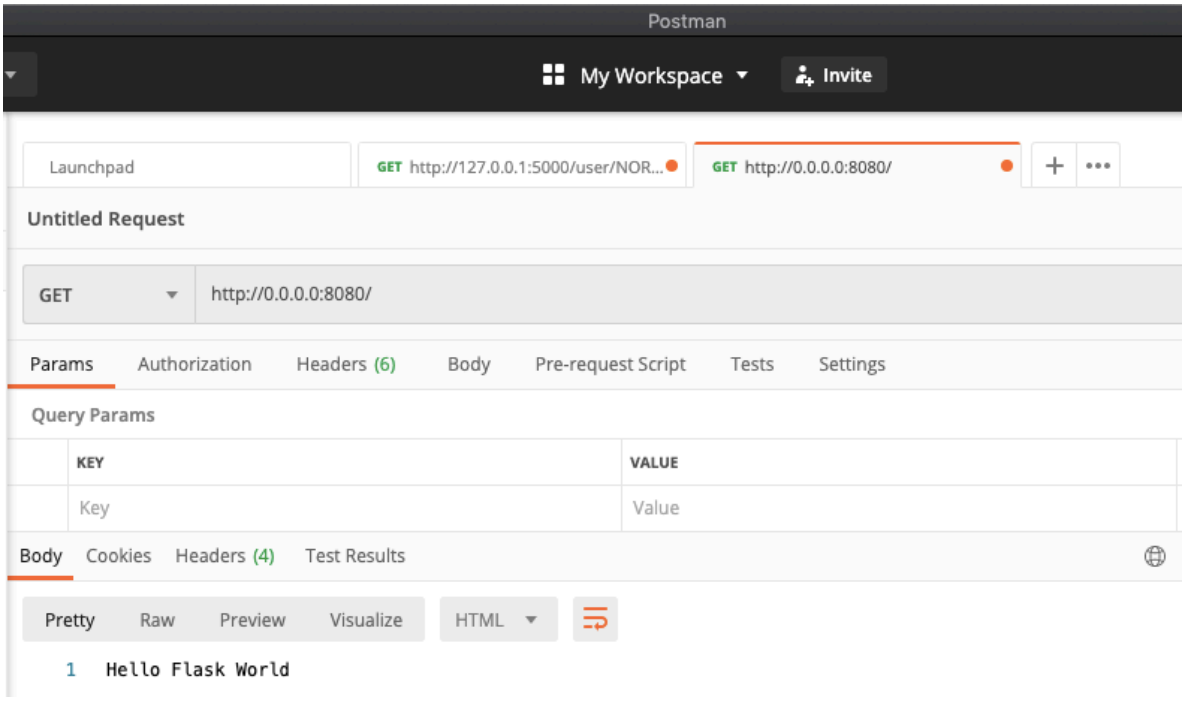
- Python File which run in the Docker Image.

Flask_Hello_World.py

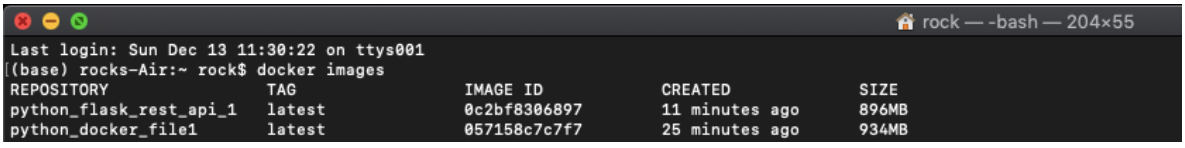
```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello Flask World'


if __name__ == '__main__':
    #app.run()
    app.run(host="0.0.0.0", debug=True, port=8080)
```

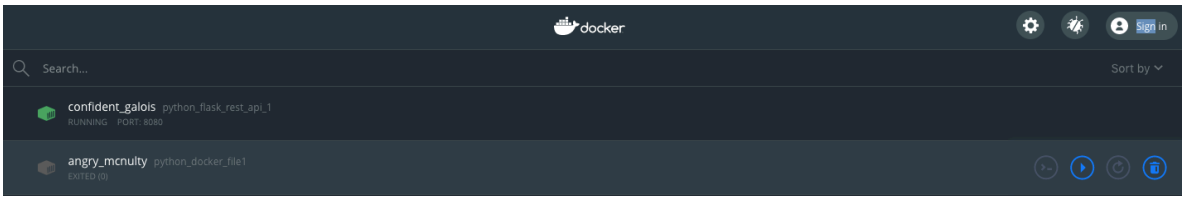
- Check Docker Flask API in Postman.

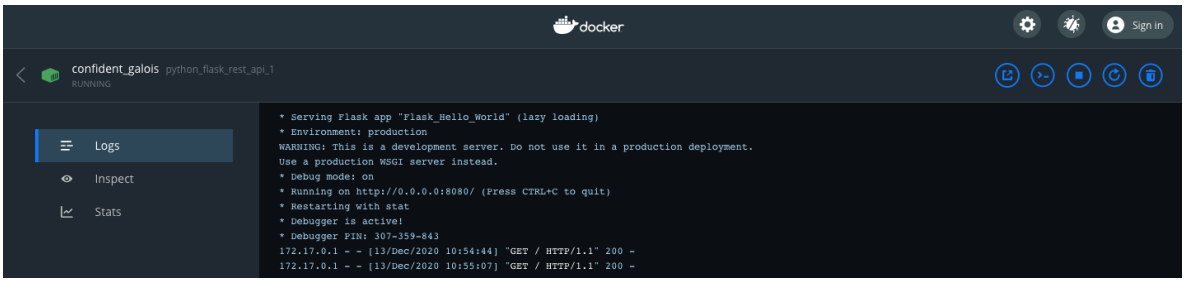


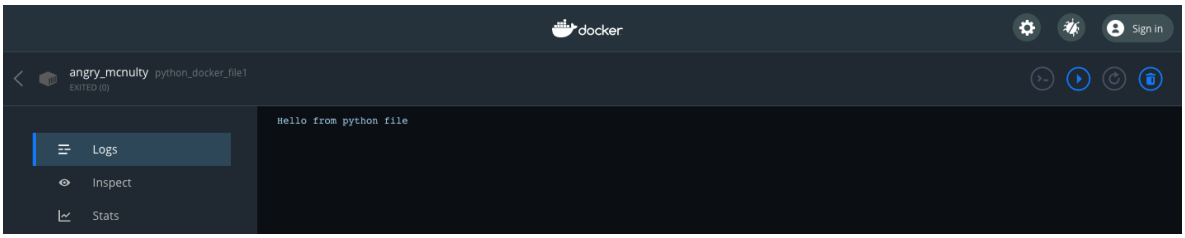


- We can see 2 Docker Images that we have created just now.




docker
Sign in
Search...
Sort by
confident_galois python_flask_rest_api_1
RUNNING PORT: 8080
angry_mcnulty python_docker_file1
EXITED (0)


docker
Sign in
confident_galois python_flask_rest_api_1
RUNNING
Logs
Inspect
Stats
* 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
172.17.0.1 - - [13/Dec/2020 10:54:44] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [13/Dec/2020 10:55:07] "GET / HTTP/1.1" 200 -


docker
Sign in
angry_mcnulty python_docker_file1
EXITED (0)
Logs
Inspect
Stats
Hello from python file