

1. Build a Docker container from a custom Docker-file

Step-1: Create a Dockerfile using nano

—> **Dockerfile**

```
FROM python:3.9-slim
WORKDIR /app
COPY req.txt .
RUN pip install --no-cache-dir -r req.txt
COPY . .
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
ENV FLASK_RUN_PORT=5000
EXPOSE 5000
CMD ["flask","run"]
```

Step-2: Create a python file app.py

→**app.py**

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello, Dockerrr...!"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

Step-3: Create a requirements file req.txt

Flask

```
Nov 2 13:48
chandanr@ubuntu: ~
chandanr@ubuntu:~$ nano dockerfile
chandanr@ubuntu:~$ nano app.py
chandanr@ubuntu:~$ ls
app.py  dockerfile  Documents  firstfile  Pictures  Public  Templates
Desktop  Dockerfiles  Downloads  Music      program1  snap    Videos
chandanr@ubuntu:~$ nano app.py
chandanr@ubuntu:~$ nano req.txt
chandanr@ubuntu:~$ sudo docker images
[sudo] password for chandanr:
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
firstfile     latest   8fc72fa2b285   5 days ago    17.8MB
<none>        <none>   8ba1428db01c   6 days ago    143MB
<none>        <none>   27e21c97a2aa   6 days ago    147MB
hello-world   latest   1b44b5a3e06a   2 months ago   10.1kB
chandanr@ubuntu:~$ sudo docker build -t p1 .
[+] Building 30.7s (10/10) FINISHED
=> [internal] load build definition from dockerfile                                0.0s
=> => transferring dockerfile: 243B                                              0.0s
=> [internal] load metadata for docker.io/library/python:3.9-slim                9.2s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                    0.0s
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acd1e13cf1731b1b  5.6s
=> => resolve docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acd1e13cf1731b1b  0.0s
=> => sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acd1e13cf1731b1b 10.36kB / 10.36kB 0.0s
=> => sha256:dad5b29e3506c35e0fd222736f4d4ef25d21b219acd73f7bb41d59996ca8e0d 1.74kB / 1.74kB 0.0s
=> => sha256:085da638e1b8a449514c3fda83ff50a3bffa4418b050cfacd87e5722071f497 5.40kB / 5.40kB 0.0s
=> => sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881d 1.29MB / 1.29MB 1.2s
=> => sha256:fc74430849022d13b0d44b0969a953f842f59c6e9d1a0c2c83d710affa286c08 13.88MB / 13.88MB 4.7s
=> => sha256:ea56f685404adf81680322f152d2cfec62115b30dda481c2c450078315beb508 251B / 251B 2.3s
=> => extracting sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881d 0.1s
=> => extracting sha256:fc74430849022d13b0d44b0969a953f842f59c6e9d1a0c2c83d710affa286c08 0.7s
=> => extracting sha256:ea56f685404adf81680322f152d2cfec62115b30dda481c2c450078315beb508 0.0s
=> [internal] load build context                                                  9.3s
=> => transferring context: 1.34GB                                               9.3s
=> [2/5] WORKDIR /app                                                            0.5s
=> [3/5] COPY req.txt .                                                          0.3s
=> [4/5] RUN pip install --no-cache-dir -r req.txt                             6.2s
=> [5/5] COPY . .                                                                2.1s
=> exporting to image                                                            3.5s
=> => exporting layers                                                            3.5s
=> => writing image sha256:52ee4529d164f3012c3b142163f7b9fe1d4b42d43437b745f229cc76bec774e1 0.0s
=> => naming to docker.io/library/p1                                            0.0s
chandanr@ubuntu:~$
```

Step-4: Build the docker image using the below command

Sudo docker build -t p1 .

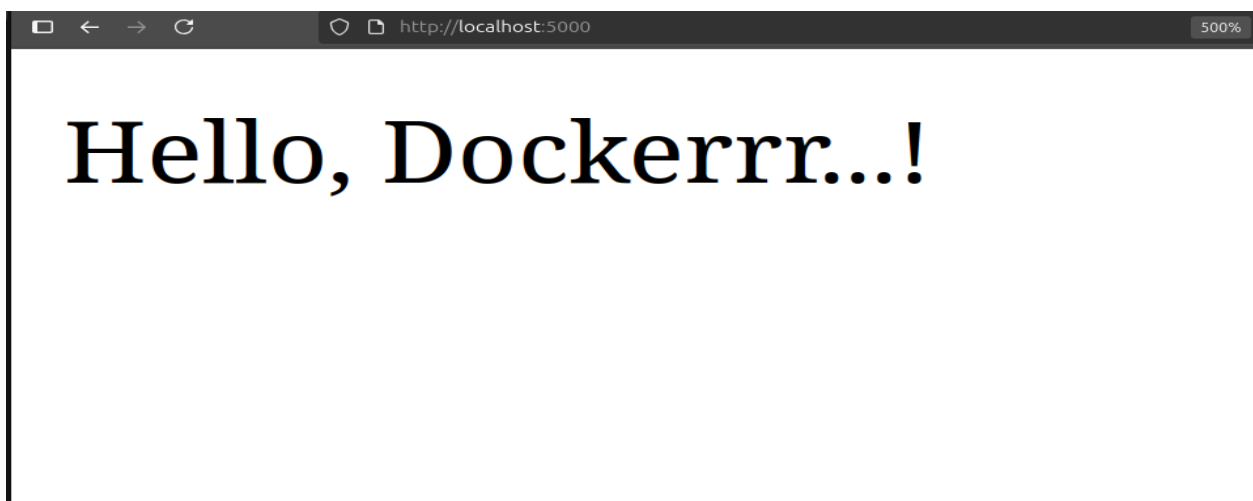
Step-5: Run the Docker Run command to create a container specifying the port number and name for the container

```
sudo docker run -d -p 5000:5000 --name PRG1 p1
```

```
Run 'docker run --help' for more information
chandanr@ubuntu:~$ docker run -d -p 5000:5000 --name PRG1 p1
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied

Run 'docker run --help' for more information
chandanr@ubuntu:~$ sudo docker run -d -p 5000:5000 --name PRG1 p1
[sudo] password for chandanr:
\\fe0d145056a8ab58bcb0e43156a80ff14f21da25b3f02a6de1390c2f2c4a5f28
chandanr@ubuntu:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
                NAMES
fe0d145056a8   p1        "flask run"              30 seconds ago Up 29 seconds 0.0.0.0:5000->5000/tcp, [::]:5000->5000/tcp   PRG1
chandanr@ubuntu:~$
```

Step-6: Test the container by verifying whether its showing some results on <http://localhost:5000> stating “hello Dockerrr!”



Step-7: Stop the container, remove the container using container number

```
sudo docker container ls -a    or    docker ps -a
sudo docker container stop <container-id>
sudo docker container rm <container-id>
sudo docker image rm <image-id>    or    docker rmi <image-id>
```

```
S referenced image 52ee4529d164
chandanr@ubuntu:~$
chandanr@ubuntu:~$ sudo docker rm fe0
fe0
chandanr@ubuntu:~$ sudo docker rmi p1
Untagged: p1:latest
Deleted: sha256:52ee4529d164f3012c3b142163f7b9fe1d4b42d43437b745f
229cc76bec774e1
chandanr@ubuntu:~$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
firstfile            latest             8fc72fa2b285       5 days ago         17.8MB
<none>               <none>             8ba1428db01c       6 days ago         143MB
<none>               <none>             27e21c97a2aa       6 days ago         147MB
hello-world          latest             1b44b5a3e06a       2 months ago       10.1kB
chandanr@ubuntu:~$
```