

## 1. Build a Docker container from a custom Dockerfile

**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, Dockerr...!"

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

**Step-3: Create a requirements file req.txt**

Flask

The screenshot shows a terminal window titled "chandanr@ubuntu:~". The terminal displays the following command sequence and its output:

```
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 38.7s (10/10) FINISHED
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 243B
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13cf1731b1b
=> => resolve docker.io/library/python:3.9-slim@sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13cf1731b1b
=> => sha256:2d97f6910b16bd338d3060f261f53f144965f755599aab1acda1e13cf1731b1b 10.36kB / 10.36kB
=> => sha256:dad5b29e3506c35e0fd222736f4d4ef25d21b219acdd73f7bb41d59996ca8e0d 1.74kB / 1.74kB
=> => sha256:085da638e1b8a449514c3fda83ff50a3bffa4e418b050cfaed87e5722071f497 5.40kB / 5.40kB
=> => sha256:b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881d 1.29MB / 1.29MB
=> => sha256:fc74430849022d13b0d44b8969a953f842f59c6e9d1a0c2c83d710affa286c08 13.88MB / 13.88MB
=> => sha256:ea56f685404adff816809322f152d2cfe62115b30dda481c2c450078315beb508 2510 / 2518
=> => extracting b3ec39b36ae8c03a3e09854de4ec4aa08381dfed84a9daa075048c2e3df3881d
=> => extracting sha256:f74430849022d13b0d44b8969a953f842f59c6e9d1a0c2c83d710affa286c08
=> => extracting sha256:ea56f685404adff816808322f152d2cfec62115b30dda481c2c450078315beb508
=> [internal] load build context
=> => transferring context: 1.34GB
=> [2/5] WORKDIR /app
=> [3/5] COPY req.txt .
=> [4/5] RUN pip install --no-cache-dir -r req.txt
=> [5/5] COPY .
=> exporting to image
=> exporting layers
=> => writing image sha256:52ee4529d164f3012c3b142163f7b9fe1d4b42d43437b745f229cc76bec774e1
=> => naming to docker.io/library/p1
chandanr@ubuntu:~$
```

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

Sudo docker build -t p1 .

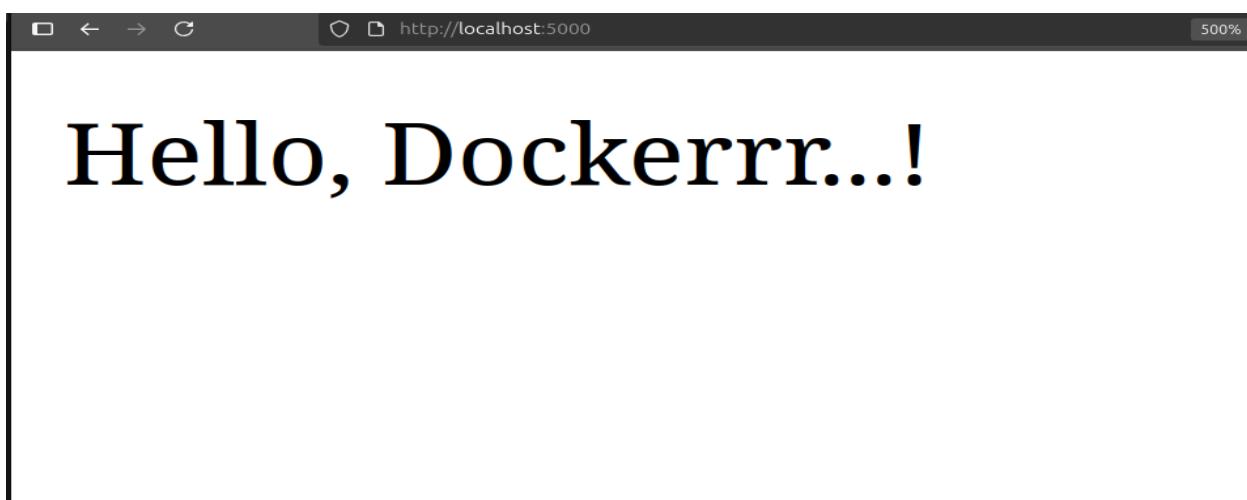
**Step-5: Run the Docker Run command to create a container specifying the port number and name for the container**

```
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://var/run/docker.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 Dockerr!”**



### 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>
```

```
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:~$
```