

Practical-4

Deploy the Machine Learning Model using Flask and Docker.

Task 1: Install the required libraries

```
pip install Flask  
pip install gunicorn
```

Task 2: Follow the steps described in theory material to deploy the model using Flask. Run the flask application to execute the deployed model.

Flask Code :

```
from flask import Flask, jsonify, request  
from your_model import predict # Import your model's prediction function  
  
app = Flask(__name__)  
  
@app.route('/predict', methods=['POST'])  
def prediction():  
    data = request.get_json(force=True)  
    result = predict(data) # Use your model to make predictions  
    return jsonify(result)  
  
if __name__ == '__main__':  
    app.run(port=5000)
```

Task 3: Create the docker file using the steps described in theory material.

Docker File Code :

```
FROM python:3.8-slim  
WORKDIR /app  
COPY . /app
```

RUN pip install --trusted-host pypi.python.org -r requirements.txt

EXPOSE 80

ENV NAME World

CMD ["python", "app.py"]

Task 4 : Create the Docker Image

docker build -t dockerfile .

```
PS D:\SEM 7\ML-OPS\Practical\practical> docker build -t dockerfile .
[+] Building 25.5s (9/9) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build definition from dockerfile
```

Task 5 : Create the Docker File

What's Next?

View summary of image vulnerabilities and recommendations → `docker scout quickview`
 PS D:\SEM 7\ML-OPS\Practical\practical> `docker run -p 4000:80 dockerfile`

Task 6 : Check Performance

```
PS D:\SEM 7\ML-OPS\Practical\practical> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
dockerfile    latest   ee193e6cc1a7   2 minutes ago  509MB
hello-world    latest   9c7a54a9a43c   6 months ago  13.3kB
```

```
PS D:\SEM 7\ML-OPS\Practical\practical> docker images
CONTAINER ID   NAME                CPU %     MEM USAGE / LIMIT   MEM %     NET I/O   BLOCK I/O   PIDS
785e4a62c222   quizzical bardeen   0.00%     0B / 0B              0.00%     0B / 0B    0B / 0B     0
```

Task 7 : Hands-on on docker commands:

1. docker pull ubuntu:latest

```
PS D:\SEM 7\ML-OPS\Practical\practical> docker pull ubuntu:latest
latest: Pulling from library/ubuntu
aece8493d397: Downloading [=====> ] 26.84MB/29.54MB
```

2. docker ps

```
PS D:\SEM 7\ML-OPS\Practical\practical> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

3. docker ps -a

```
PS D:\SEM 7\ML-OPS\Practical\practical> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
785e4a62c222	dockerfile	"python app.py"	7 minutes ago	Exited (0) 7 minutes ago		quizzical_bardeen
523f21a1dd21	dockerfile	"python app.py"	8 minutes ago	Exited (0) 8 minutes ago		xenodochial_moser
98032478cfe5	hello-world:latest	"/hello"	2 months ago	Exited (0) 25 minutes ago		mystifying_fermi

4. docker inspect container_name or id

```
PS D:\SEM 7\ML-OPS\Practical\practical> docker inspect 785e4a62c222
```

```
[
  {
    "Id": "785e4a62c2221af87166077e7d40adb41cac19f42a69ec6c4200dc583e5eb5ab",
    "Created": "2023-11-10T14:03:56.854950025Z",
    "Path": "python",
    "Args": [
      "app.py"
    ],
    "State": {
      "Status": "exited",
      "Running": false,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 0,
```