

Practical-6

Deployment of ML project using Flask.

Task 1: Ensure that the required libraries are installed

pip install Flask

pip install gunicorn

Task 2: Create the docker file using the steps described in theory material.

a) Create a Dockerfile :

```
FROM python:3.8-slim
WORKDIR /app
COPY . /app
RUN pip install --no-cache-dir -r requirements.txt
EXPOSE 80
ENV NAME World
CMD ["gunicorn", "--bind", "0.0.0.0:80", "app:app"]
```

b) Create a requirement.txt file :

```
scikit-learn==0.24.2
pandas==1.3.3
numpy==1.21.2
flask==2.1.0
gunicorn==20.1.0
```

c) Create a Docker Image :

```
[+] Building 65.7s (8/9)
=> => sha256:1fb7efcf9eab7803298874aca4438f97958ccef72e9d62bf6c7654b5d9c92c40 3.51MB / 3.51MB
=> => sha256:ec9a8be8d55c26df0ad6648b4a2cf81563a89cd042b0d16f0ab58eef2cf0e4ac 13.75MB / 13.75MB
=> => sha256:0b0ea7fc90f399b2ca372776ea4b1b7ce28c725e86a2b96066262846942c68fd 245B / 245B
=> => sha256:8b9b67f59e57ed7961ac441a98c5e7481c9ddb658dc2df313fe14931f032f1c3 3.14MB / 3.14MB
=> => extracting sha256:1f7ce2fa46ab3942feabee654933948821303a5a821789dddab2d8c3df59e227
=> => extracting sha256:1fb7efcf9eab7803298874aca4438f97958ccef72e9d62bf6c7654b5d9c92c40
=> => extracting sha256:ec9a8be8d55c26df0ad6648b4a2cf81563a89cd042b0d16f0ab58eef2cf0e4ac
=> => extracting sha256:0b0ea7fc90f399b2ca372776ea4b1b7ce28c725e86a2b96066262846942c68fd
=> => extracting sha256:8b9b67f59e57ed7961ac441a98c5e7481c9ddb658dc2df313fe14931f032f1c3
=> [internal] load build context
=> => transferring context: 752.28MB
=> [2/4] WORKDIR /app
=> [3/4] COPY . /app
=> [4/4] RUN pip install --no-cache-dir -r requirements.txt
=> => # Collecting six>=1.5
=> => # Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
```

- Check the image is created or not :

```
PS D:\Capstone Project-1> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
project	latest	75ebfac9ab69	5 minutes ago	1.23GB
dockerfile	latest	ee193e6cc1a7	12 days ago	509MB
ubuntu	latest	e4c58958181a	7 weeks ago	77.8MB
hello-world	latest	9c7a54a9a43c	6 months ago	13.3kB

```
PS D:\Capstone Project-1> ls
```

Mode	LastWriteTime	Length	Name
d----	30-12-2022 11:43 PM		.idea
d----	28-06-2023 12:40 PM		Capstone Project-1
d----	12-03-2023 12:11 PM		Car Price Prediction
d----	30-06-2023 06:57 PM		Datasets
d----	30-12-2022 07:43 PM		Group Members
d----	07-05-2023 06:36 AM		Laptop_Price_Prediction
d----	30-12-2022 08:13 PM		model
d----	27-12-2022 02:54 PM		PPT
d----	01-07-2023 07:21 PM		README
d----	02-05-2023 12:48 PM		Report
d----	28-06-2023 02:38 PM		UI
-a----	13-04-2023 01:36 PM	108	updated

- Locate the file app.py and start build of a project

```
PS D:\Capstone Project-1\UI> cd '.\New UI\'
PS D:\Capstone Project-1\UI\New UI> ls
```

Directory: D:\Capstone Project-1\UI\New UI

Mode	LastWriteTime	Length	Name
d----	23-04-2023 02:03 PM		static
d----	07-05-2023 04:02 AM		templates
-a----	23-11-2023 04:10 PM	4701	app_final.py
-a----	23-11-2023 04:24 PM	550	Dockerfile
-a----	23-11-2023 04:24 PM	94	requirements.txt

```
PS D:\Capstone Project-1\UI\New UI> docker build -t projecta .
[+] Building 0.0s (0/0)
[+] Building 39.8s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 589B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3.8-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [1/4] FROM docker.io/library/python:3.8-slim@sha256:19e07fa24813e88b04e606772213bd03ba044637cc939a211e28ccf997a9162a
=> [internal] load build context
=> => transferring context: 13.14MB
```

Task 4: Run the docker container to execute the docker image and host the machine learning model using gunicorn wsgi server.

```
D:\Capstone Project-1\UI\New UI>
D:\Capstone Project-1\UI\New UI>docker run -p 4000:80 projecta
[2023-11-23 11:17:21 +0000] [1] [INFO] Starting gunicorn 20.1.0
[2023-11-23 11:17:21 +0000] [1] [INFO] Listening at: http://0.0.0.0:80 (1)
[2023-11-23 11:17:21 +0000] [1] [INFO] Using worker: sync
[2023-11-23 11:17:21 +0000] [8] [INFO] Booting worker with pid: 8
```

Task 5: Compare the performance of the model in docker container and flask script deployment.

Local :

Time : 9.92 sec

Docker :

Time : 8.18 sec

The performance of Docker compared to local hosting can vary depending on several factors, and it's not always straightforward to determine whether Docker is faster or slower. Here are some considerations

- **Container Overhead**
- **Resource Allocation**
- **Host System Resources**

Reasoning:

Faster in Docker: Docker might be faster due to optimized resource allocation, efficient caching, and consistent environments, especially if the Docker container is well-tailored for the application.

Slower in Docker: Docker might be slower if the overhead introduced by containerization is significant, or if there are misconfigurations impacting resource utilization.

Conclusion:

The decision on whether Docker hosting is faster or slower depends on the specific characteristics of your application and how well it is configured for Docker. It's recommended to perform detailed benchmarking and resource monitoring to draw accurate conclusions for your particular use case.