Practical: Containerise a Python application

Practical: Containerise an application (Python)

This version of the practical assumes you are working within the EC2 instance with Docker installed you created in the previous practical. This is the Python version of demonstrating how to containerise an application using Docker.

1. Clone repository from Github that has a barebones python api application using FastAPI.

```
git clone https://github.com/estebanx64/python-docker-example && cd p
ython-docker-example
```

2. We will need to create the DockerFile. In this instance, we will create a file named `Dockerfile` in the root of the project with the following content:

```
# Use the official Python image from the Docker Hub
FROM python:3.11-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
EXPOSE 8080
CMD ["python3", "-m", "uvicorn", "app:app", "--host=0.0.0.0", "--port
=8080"]
```

Here is a brief explanation of the Dockerfile:

- `FROM python:3.11-slim`: This line specifies the base image to use, which is a slim version of Python 3.11.
- `WORKDIR /app`: This sets the working directory inside the container to `/app`.
- `COPY requirements.txt .`: This copies the `requirements.txt` file into the container.
- `RUN pip install --no-cache-dir -r requirements.txt`: This installs the Python dependencies specified in the `requirements.txt` file.
- `COPY . .`: This copies the rest of the application code into the container.
- `EXPOSE 8080`: This exposes port 8080, which is the port the FastAPI application will run on.
- `CMD ["python3", "-m", "uvicorn", "app:app", "--host=0.0.0.0", "--port=8080"]`: This specifies the command to run the FastAPI application using uvicorn. The application will be running on localhost on port 8080.

3. We will build the Docker image using the following command:

```
docker build -t python-docker-example .
```

4. After building the image, we can run the container using the following command:

```
docker run -dp 8080:8080 python-docker-example
```

5. To verify that the application is running, you can open your web browser and navigate to `http://<your-ec2-instance-ip>:8080/docs`. You should see an Hello world message displayed in the browser, which indicates that the FastAPI application is running successfully.