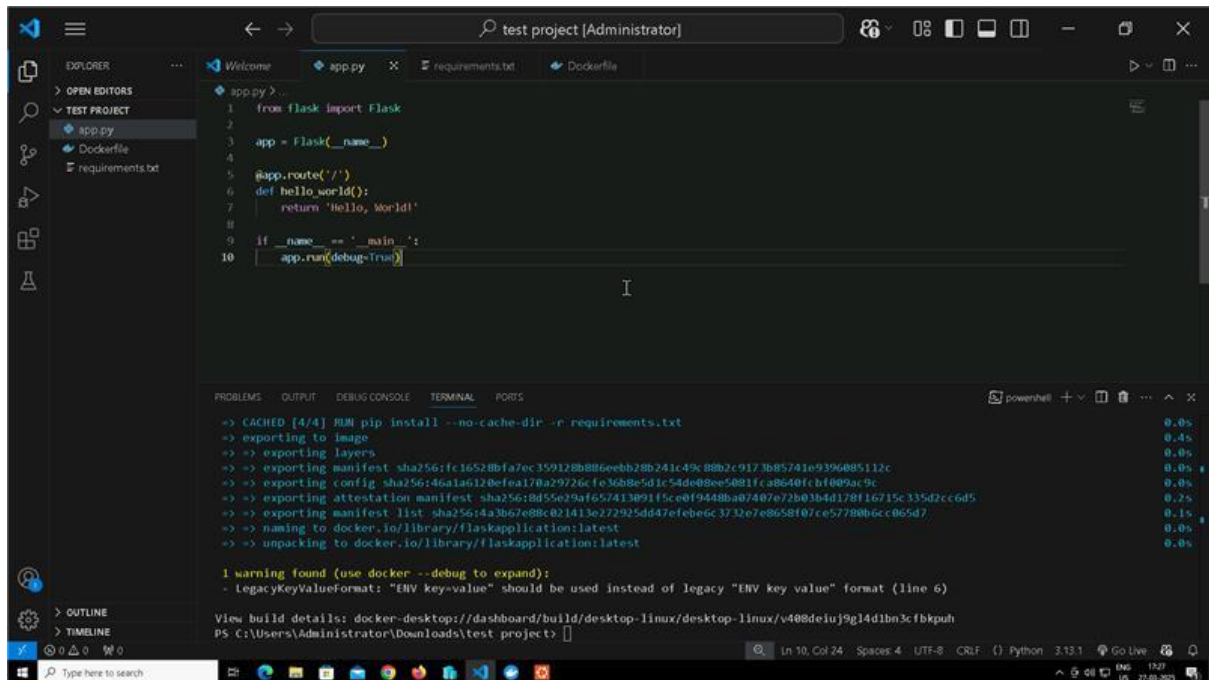


Deploy a Python Flask App using Docker

1)create app.py



The screenshot shows the Visual Studio Code editor with a file named `app.py` open. The code in `app.py` is as follows:

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 @app.route('/')
6 def hello_world():
7     return 'Hello, World!'
8
9 if __name__ == '__main__':
10     app.run(debug=True)
```

The terminal at the bottom shows the output of the `docker build` command. It indicates that the requirements.txt file has been installed and the application is being exported to a Docker image. The output includes the following lines:

```
=> CACHED [4/4] RUN pip install --no-cache-dir -r requirements.txt 0.0s
=> exporting to image 0.4s
=> exporting layers 0.0s
=> exporting manifest sha256:fc16528bfa7ec359128b886ebb28b241c49c88b2c9173b85741e9396085112c 0.0s
=> exporting config sha256:46a1a6120feaf170a29726cfe36b8e5d1c54de08ee5881fca8640fcbf009ac9c 0.0s
=> exporting attestation manifest sha256:8d55e29af657413091f5ce0f9448ba07407e72b03b4d178f16715c335d2cc6d5 0.2s
=> exporting manifest list sha256:4a3b67e88c021413e272925dd47efebec3732e7e8658f07ce57780b6cc065d7 0.1s
=> naming to docker.io/library/flaskapplication:latest 0.0s
=> unpacking to docker.io/library/flaskapplication:latest 0.0s
```

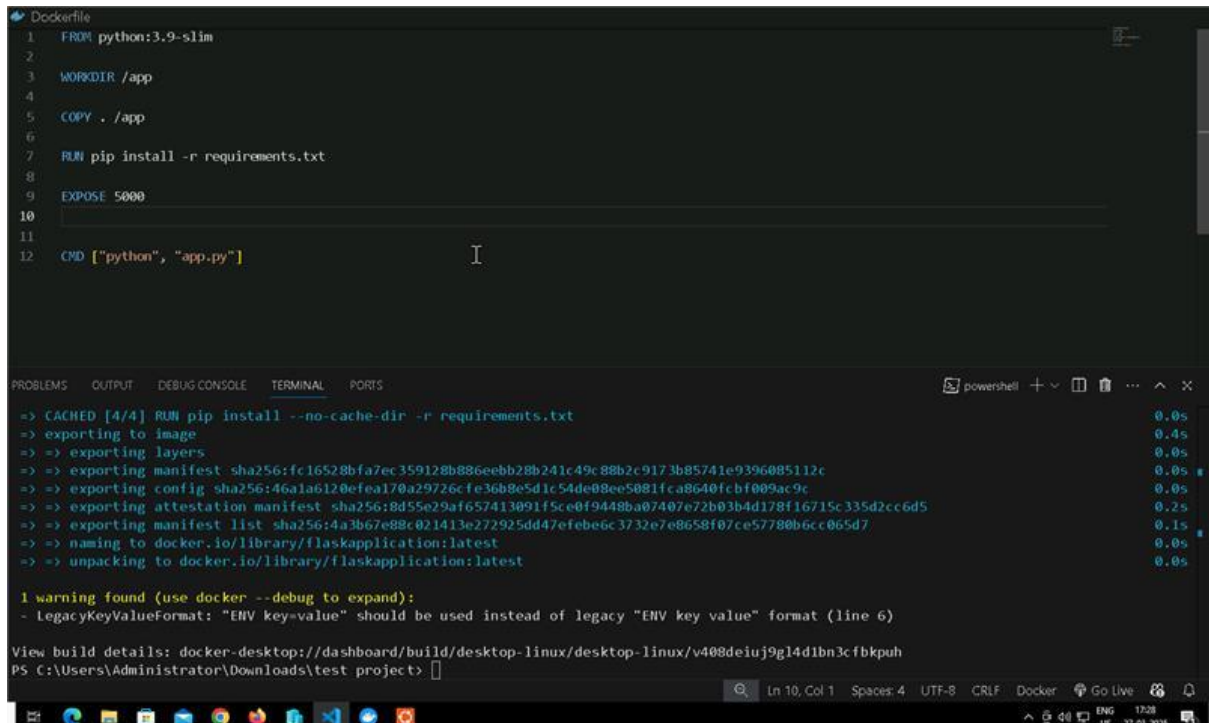
A warning is also shown in the terminal:

```
1 warning found (use docker --debug to expand):
- LegacyKeyValueFormat: "ENV key=value" should be used instead of legacy "ENV key value" format (line 6)
```

The terminal also shows the build details and the current directory:

```
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/v408deiu9g14d1bn3cfbkpuh
PS C:\Users\Administrator\Downloads\test project>
```

2)create the docker file and requirements.txt file



The screenshot shows the Visual Studio Code editor with a file named `Dockerfile` open. The content of the `Dockerfile` is as follows:

```
1 FROM python:3.9-slim
2
3 WORKDIR /app
4
5 COPY . /app
6
7 RUN pip install -r requirements.txt
8
9 EXPOSE 5000
10
11
12 CMD ["python", "app.py"]
```

The terminal at the bottom shows the output of the `docker build` command. It indicates that the Dockerfile has been built and the application is being exported to a Docker image. The output includes the following lines:

```
=> CACHED [4/4] RUN pip install --no-cache-dir -r requirements.txt 0.0s
=> exporting to image 0.4s
=> exporting layers 0.0s
=> exporting manifest sha256:fc16528bfa7ec359128b886ebb28b241c49c88b2c9173b85741e9396085112c 0.0s
=> exporting config sha256:46a1a6120feaf170a29726cfe36b8e5d1c54de08ee5881fca8640fcbf009ac9c 0.0s
=> exporting attestation manifest sha256:8d55e29af657413091f5ce0f9448ba07407e72b03b4d178f16715c335d2cc6d5 0.2s
=> exporting manifest list sha256:4a3b67e88c021413e272925dd47efebec3732e7e8658f07ce57780b6cc065d7 0.1s
=> naming to docker.io/library/flaskapplication:latest 0.0s
=> unpacking to docker.io/library/flaskapplication:latest 0.0s
```

A warning is also shown in the terminal:

```
1 warning found (use docker --debug to expand):
- LegacyKeyValueFormat: "ENV key=value" should be used instead of legacy "ENV key value" format (line 6)
```

The terminal also shows the build details and the current directory:

```
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/v408deiu9g14d1bn3cfbkpuh
PS C:\Users\Administrator\Downloads\test project>
```

3)Build the docker image by using docker build command

```
PS C:\Users\Administrator\Downloads\test project> docker build -t flaskapplication .
[+] Building 3.1s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 201B
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/python:3.9-slim@sha256:e52ca5f579cc58fed41efcbb55a0ed5dccf6c7a156cba76acfb4ab42fc19dd00
=> => resolve docker.io/library/python:3.9-slim@sha256:e52ca5f579cc58fed41efcbb55a0ed5dccf6c7a156cba76acfb4ab42fc19dd00
=> [internal] load build context
=> => transferring context: 93B
=> => resolve docker.io/library/python:3.9-slim@sha256:e52ca5f579cc58fed41efcbb55a0ed5dccf6c7a156cba76acfb4ab42fc19dd00
=> [internal] load build context
=> => transferring context: 93B
=> CACHED [2/4] WORKDIR /app
=> CACHED [2/4] WORKDIR /app
=> CACHED [3/4] COPY . /app
=> CACHED [4/4] RUN pip install --no-cache-dir -r requirements.txt
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:fc16528bfa7ec359128b886eebb28b241c49c88b2c9173b85741e9396085112c
```

4)Now run it using docker run

```
Select ubuntu@77d2e6b99e1b5c6: ~
ubuntu@77d2e6b99e1b5c6:~$ docker run -it -p 5001:5000 flaskapplication
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 291-072-221
```

5)So in the below output we can see that it is running on port 5001

