

DEPLOYMENT OF APP IN IBM CLOUD

CONTAINERIZE THE APP

TEAM ID	PNT2022TMID02772
PROJECT NAME	Smart Fashion Recommender Application

In your project directory, create a file named "Dockerfile". In the file, paste this code. Open the terminal and type this command to build an image from your Dockerfile: `docker build -t hello-world:latest`.

The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a project structure with files like `_pycache_`, `venv`, `Include`, `Lib`, `Scripts`, `.gitignore`, `Dockerfile`, `pyvenv.cfg`, and `app.py`. The Dockerfile is open in the editor, showing the following content:

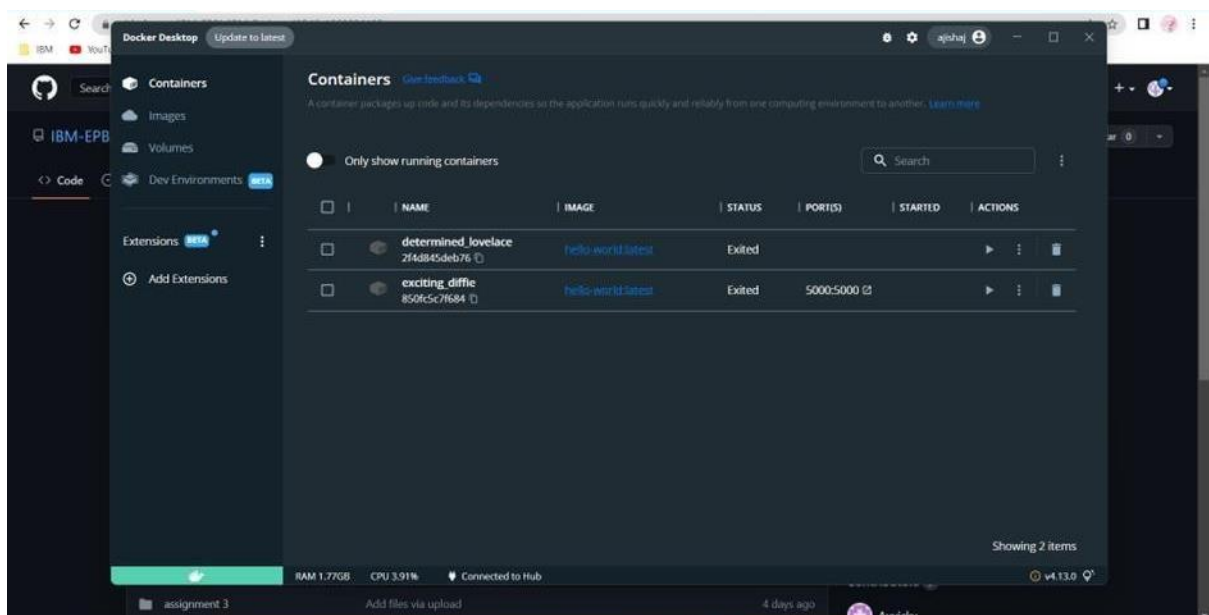
```

1 FROM python:2.7
2 LABEL maintainer="Kunal Malhotra, kunal.malhotra@ibm.com"
3 RUN apt-get update
4 RUN mkdir /app
5 WORKDIR /app
6 COPY . /app
7 RUN pip install -r requirements.txt
8 EXPOSE 5000
9 ENTRYPOINT [ "python" ]
10 CMD [ "app.py" ]

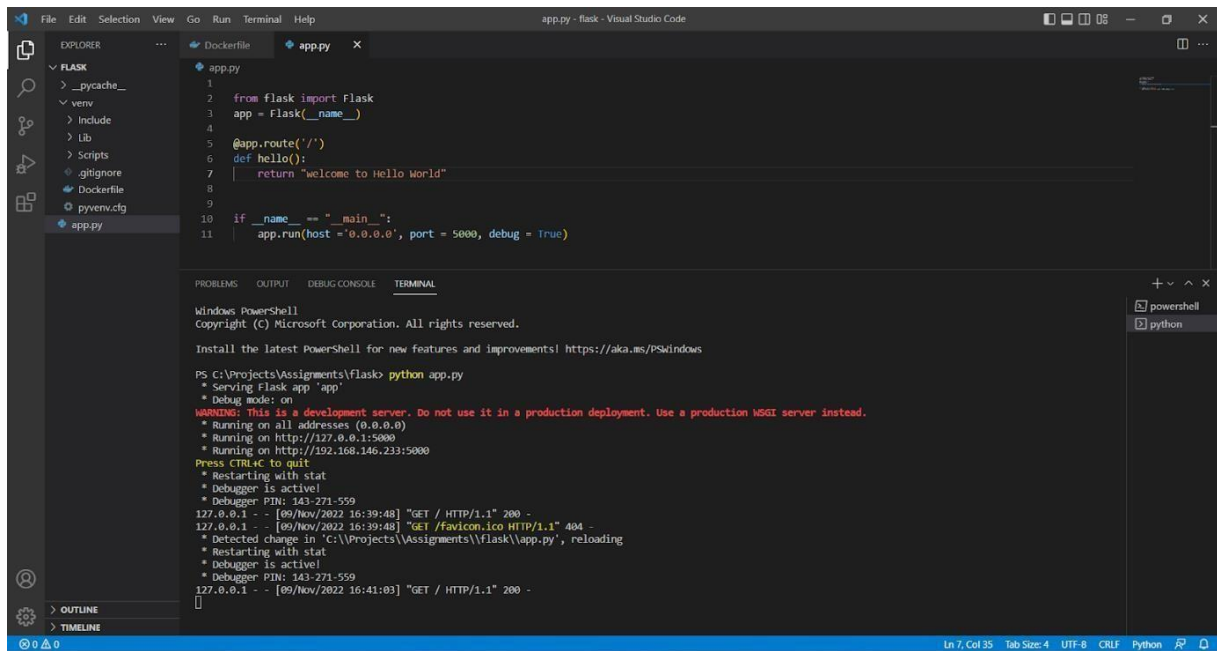
```

The Dockerfile documentation is displayed in the bottom pane, listing various options for container configuration:

- `--runtime string`: Runtime to use for this container
- `--security-opt list`: Security Options
- `--shm-size bytes`: Size of /dev/shm
- `--sig-proxy`: Proxy received signals to the process (default true)
- `--stop-signal string`: Signal to stop a container (default "15")
- `--stop-timeout int`: Timeout (in seconds) to stop a container
- `--storage-opt list`: Storage driver options for the container
- `--sysctl map`: Sysctl options (default map[])
- `--tmpfs list`: Mount a tmpfs directory
- `-t, --tty`: Allocate a pseudo-TTY
- `--ulimit ulimit`: Ulimit options (default [])
- `-u, --user string`: Username or UID (format: `username[:uid[:group[:gid]]]`)
- `--users string`: User namespace to use
- `--uts string`: UTS namespace to use
- `-v, --volume list`: Bind mount a volume
- `--volume-driver string`: Optional volume driver for the container
- `--volumes-from list`: Mount volumes from the specified container(s)
- `-w, --workdir string`: Working directory inside the container



Test by running the code in localhost



The screenshot shows the Visual Studio Code interface with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code editor displays a Python file named `app.py` with the following content:

```
1 from flask import Flask
2 app = Flask(__name__)
3
4
5 @app.route('/')
6 def hello():
7     return "welcome to Hello World"
8
9
10 if __name__ == "__main__":
11     app.run(host='0.0.0.0', port=5000, debug=True)
```

The terminal window shows the output of running the application. It starts with a Windows PowerShell prompt, followed by the command `python app.py`. The output indicates that the application is running on `http://127.0.0.1:5000` and `http://192.168.146.233:5000`. It also shows a warning about using a development server and a message about the debugger being active. The terminal output includes the following lines:

```
PS C:\Projects\Assignments\flask> python app.py
* 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 all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.146.233:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 143-271-559
127.0.0.1 - - [09/Nov/2022 16:39:48] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Nov/2022 16:39:48] "GET /favicon.ico HTTP/1.1" 404 -
* Detected change in 'C:\Projects\Assignments\flask\app.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 143-271-559
127.0.0.1 - - [09/Nov/2022 16:41:03] "GET / HTTP/1.1" 200 -
```

