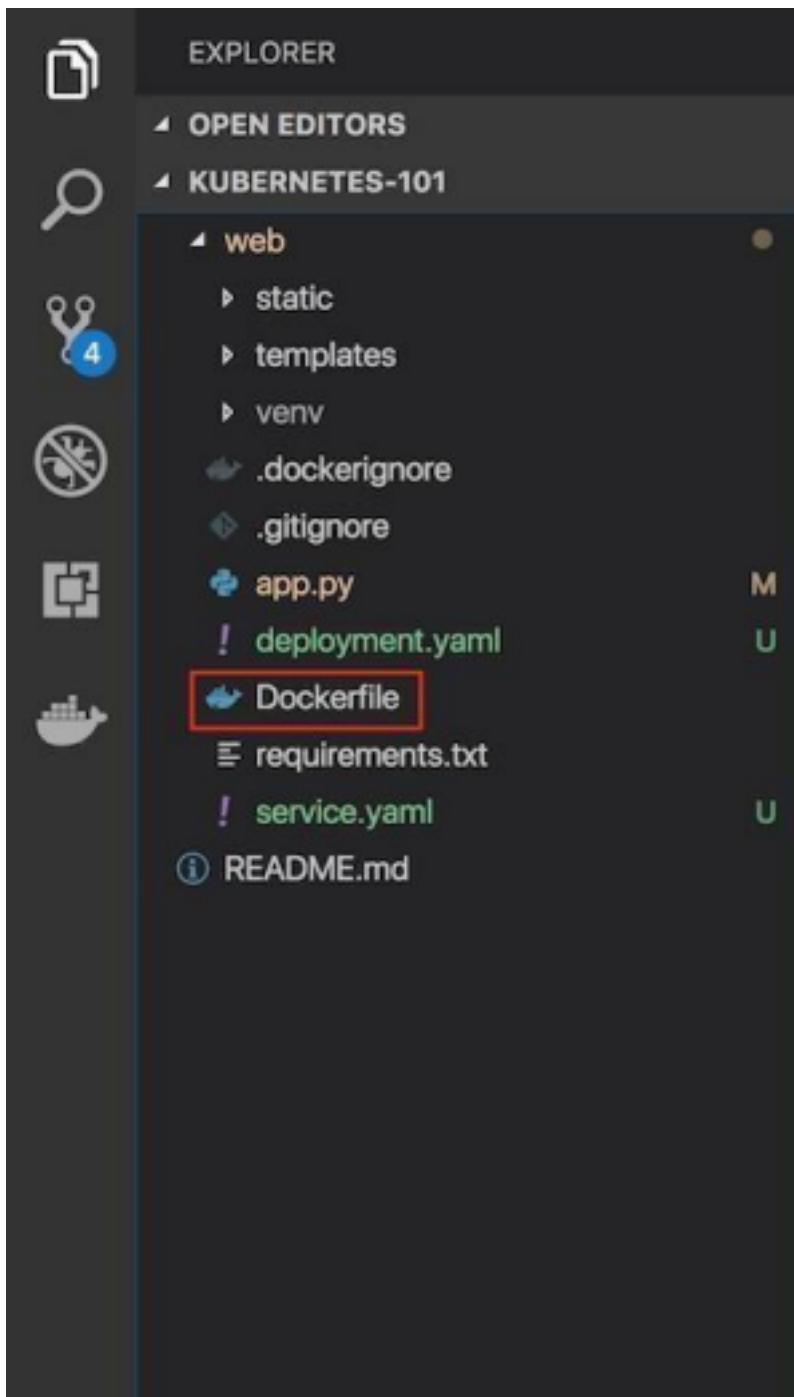


CONTAINERIZE THE APP

TEAM ID	PNT2022TMID33378
PROJECT NAME	SKILL AND JOB RECOMMENDER APPLICATION

Containerize your Flask application

- In your project directory, create a file named "Dockerfile." *Suggestion: Name your file exactly "Dockerfile," nothing else.*



A "Dockerfile" is used to indicate to Docker a base image, the Docker settings you need, and a list of commands you would like to have executed to prepare and start your new container.

- In the file, paste this code:

•

```
FROM python:2.7
LABEL maintainer="Kunal Malhotra, kunal.malhotra1@ibm.com"
```

- RUN apt-get update
- RUN mkdir /app WORKDIR /app COPY . /app
- RUN pip install -r requirements.txt
- EXPOSE 5000
- ENTRYPOINT ["python"]
- CMD ["app.py"]

Show more

Explanation and breakdown of the above Dockerfile code

1. The first part of the code above is:
2. FROM python:2.7

Show more

Because this Flask application uses Python 2.7, we want an environment that supports it and

already has it installed. Fortunately, DockerHub has an official image that's installed on top of Ubuntu. In one line, we will have a base Ubuntu image with Python 2.7, virtualenv, and pip. There are tons of images on DockerHub, but if you would like to start off with a fresh Ubuntu image and build on top of it, you could do that.

3. Let's look at the next part of the code:
4. LABEL maintainer="Kunal Malhotra, kunal.malhotra1@ibm.com"
5. RUN apt-get update

Show more

6. Note the maintainer and update the Ubuntu package index. The command is RUN, which is a function that runs the command after it.
7. RUN mkdir /app
8. WORKDIR /app
9. COPY . /app

Show more

10. Now it's time to add the Flask application to the image. For simplicity, copy the application under the /app directory on our Docker Image.

WORKDIR is essentially a **cd** in bash, and COPY copies a certain directory to the provided directory in an image. ADD is another command that does the same thing as COPY, but it also allows you to add a repository from a URL. Thus, if you want to clone your git repository instead of copying it from your local repository (for staging and production purposes), you can use that. COPY, however, should be used most of the time unless you have a URL.

11. Now that we have our repository copied to the image, we will install all of our dependencies, which is defined in the requirements.txt part of the code.
12. RUN pip install --no-cache-dir -r requirements.txt

Show more

14. EXPOSE 5000

15. **ENTRYPOINT** specifies the entrypoint of your application.

```
17.CMD [ "app.py" ]
```

```

kubernetes@kali:~/k8s$ docker build --rm -t app:latest
Sending build context to Docker daemon 346.2kB
Step 1/8 : FROM python:2.7
--> c0f9c3d97cf6
Step 2/8 : LABEL maintainer="Karel Nohava, karel.nohava@firma.cz"
--> Using cache
--> d895c411393e
Step 3/8 : RUN apt-get update
--> Using cache
--> 6262c11a9eb6
Step 4/8 : COPY *.app
--> ff07117889ff
Step 5/8 : RUN chmod +x *.app
Removing intermediate container f93db79623fe
--> bdecdf9bdc
Step 6/8 : RUN pip install -r requirements.txt
--> Running in 8120b400a0ef
Collecting click==6.7 (from -r requirements.txt [line 2])
  Downloading https://files.pythonhosted.org/packages/f6/a2/3bb8f7bf15a8899c33ba3623f7f81c2391629af11ab5c78077a77711cb-6.7-py2.py3-none-any.whl (71kB)
Collecting Flask==1.0.2 (from -r requirements.txt [line 2])
  Downloading https://files.pythonhosted.org/packages/71/b7/8657673aa93365024314abc4403380334070f24ca05703cc84b77cask-1.0.2-py2.py3-none-any.whl (29kB)
Collecting Werkzeug==0.14 (from -r requirements.txt [line 2])
  Downloading https://files.pythonhosted.org/packages/c6/a3/00bc8cd95e08f5a8648975131ac3750272bcfc3ee021368346234/tzanderwerkzeug-0.14.tar.gz (#464)
Collecting Jinja2==2.10 (from -r requirements.txt [line 2])
  Downloading https://files.pythonhosted.org/packages/71/b7/8657673aa93365024314abc4403380334070f24ca05703cc84b77cask-2.10-py2.py3-none-any.whl (126kB)
Collecting MarkupSafe==1.0 (from -r requirements.txt [line 2])
  Downloading https://files.pythonhosted.org/packages/d4/ae/33d36b23652375aa25c476a76dad3ef9a662cefd8ea48ff032126/Werkzeug-1.0.1.tar.gz
Collecting Werkzeug==0.14.1 (from -r requirements.txt [line 2])
  Downloading https://files.pythonhosted.org/packages/0b/d4/33d36b23652375aa25c476a76dad3ef9a662cefd8ea48ff032126/Werkzeug-0.14.1-py2.py3-none-any.whl (#1268)
Building wheels for collected packages: tzanderwerkzeug, MarkupSafe
Running setup.py install.sheel for tzanderwerkzeug: started
Running setup.py install.sheel for tzanderwerkzeug: finished with status 'done'
Stored in directory: /root/.cache/pip/wheels/13/56/0b/4e494de32ff7fcc5a0214501d39f76447676664ee46
Running setup.py install.sheel for MarkupSafe: started
Running setup.py install.sheel for MarkupSafe: finished with status 'done'
Stored in directory: /root/.cache/pip/wheels/13/56/0b/4e494de32ff7fcc5a0214501d39f76447676664ee46
Successfully built tzanderwerkzeug MarkupSafe
Installing collected packages: click, tzanderwerkzeug, Jinja2, Werkzeug, Flask
Successfully installed Flask-1.0.2 Jinja2-2.10 MarkupSafe-1.0 Werkzeug-0.14.1 click-6.7 tzanderwerkzeug-0.14
Removing intermediate container 8338a4dbb7
--> 846c30a097e
Step 7/8 : ENTRYPOINT ["python"]
--> Running in bc0c3f1815ef
Removing intermediate container bc0c3f1815ef
--> f3a6c18acc
Step 8/8 : CMD ["*.app.py"]
--> Running in c784d33a48ef
Removing intermediate container c784d33a48ef
--> 88ad35763d
Successfully built app:latest
Successfully tagged app:latest
kubernetes@kali:~/k8s$
```

[illegible]

