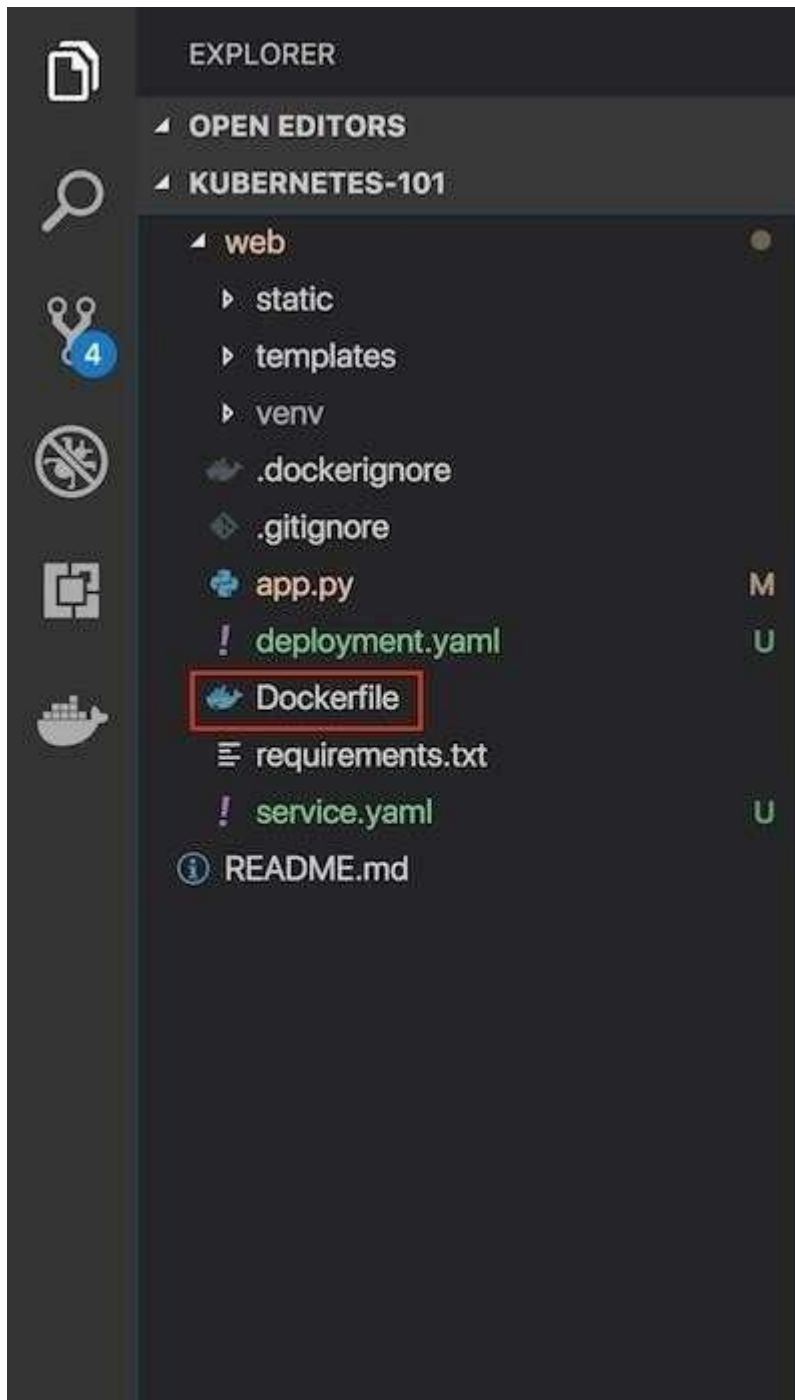


CONTAINERIZE THE APP

- In your project directory, create a file named "Dockerfile."



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
6. Note the maintainer and update the Ubuntu package index.

```
7. RUN mkdir /app
8. WORKDIR /app
9. COPY . /app
```

The command is RUN, which is a function that runs the command after it.

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.

10. 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.
11. RUN pip install --no-cache-dir -r requirements.txt
12. We want to expose the port(5000) the Flask application runs on, so we use EXPOSE.
13. EXPOSE 5000
14. ENTRYPOINT specifies the entrypoint of your application.
15. ENTRYPOINT ["python"]
16. CMD ["app.py"]

Build an image from the Dockerfile

Open the terminal and type this command to build an image from your Dockerfile: `docker build -t <image_name>:<tag> .` (note the period to indicate we're in our app's top level directory). For example: `docker build -t app:latest .`

```

kunal@stephen-kunal@kunal:~$ docker build -t app:latest .
Sending build context to Docker daemon 348.2kB
Step 1/8 : FROM python:2.7
--> 6c76b9e7cfe
Step 2/8 : LABEL maintainer="Kunal Mishra, kunal.mishra@ppl.com"
--> Using cache
--> 88c5941591c
Step 3/8 : RUN apt-get update
--> Using cache
--> 6266134e6e
Step 4/8 : COPY ./app
--> f877777000f
Step 5/8 : WORKDIR /app
Removing intermediate container f8c8b994cfe
--> 06cd570b2c
Step 6/8 : RUN pip install -r requirements.txt
--> Running in 81534488b7
Collecting click==6.7 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/34/c1/88619573d8695c396c362c21968f3829360752off1a6f400775e77/click-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/7f/67/8857874e8f536d342b144c3d963663468fa324ed957262c0bedb6f1ask-1.0.2-py2.py3-none-any.whl (81kB)
Collecting itsdangerous==0.24 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/d0/b4/60228b945c08f608a887131a6f29a22f2cfe1a82211651946246/itsdangerous-0.24.tar.gz (46kB)
Collecting Jinja2==2.10 (from -r requirements.txt (line 4))
  Downloading https://files.pythonhosted.org/packages/7f/7f/ae64eeefc95f27f80a7e68a886763a64277a78a76f3289328a71175eja2-2.10-py2.py3-none-any.whl (128kB)
Collecting MarkupSafe==1.0 (from -r requirements.txt (line 5))
  Downloading https://files.pythonhosted.org/packages/4d/dc/32f41d835a8f4b768822a03700ef7e4825ee929e4fcbaf417b/MarkupSafe-1.0.tar.gz
Collecting Werkzeug==0.14.1 (from -r requirements.txt (line 6))
  Downloading https://files.pythonhosted.org/packages/20/c4/f22a3c6473e3275a29c4764476d18f3ef06629ef580ea4f635243/Werkzeug-0.14.1-py2.py3-none-any.whl (325kB)
Building wheels for collected packages: itsdangerous, MarkupSafe
  Running setup.py bdist_wheel for itsdangerous: started
  Running setup.py bdist_wheel for itsdangerous: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/20/46/61/550961c1547886529068680c736731751874c4602f1a5
  Running setup.py bdist_wheel for MarkupSafe: started
  Running setup.py bdist_wheel for MarkupSafe: finished with status 'done'
  Stored in directory: /root/.cache/pip/wheels/c3/56/20/eeef5cd027f7a1c5d63246b18930f96467070862e4e46
Successfully built itsdangerous MarkupSafe
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed Flask-1.0.2 Jinja2-2.10 MarkupSafe-1.0 Werkzeug-0.14.1 click-6.7 itsdangerous-0.24
Removing intermediate container 81534488b7
--> 06cd570b2c
Step 7/8 : ENTRYPOINT ["python"]
--> Running in bc3c8315d
Removing intermediate container bc3c8315d
--> 750fc38c1c
Step 8/8 : DO ["now.py"]
--> Running in a784830bdf
Removing intermediate container a784830bdf
--> 88cd8376d5
Successfully built 88cd8376d5
Successfully tagged app:latest
kunal@stephen-kunal@kunal:~$

```

Run your container locally and test

After you build your image successfully, type: `docker run -d -p 5000:5000 app`

This command will create a container that contains all the application code and dependencies from the image and runs it locally.

```

kunal@stephen-kunal@kunal:~$ docker ps -a -p 5000:5000 app
3131af6775b0c0a664c12e7363cd996b43131c0543d8b63247
kunal@stephen-kunal@kunal:~$ docker ps

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3131af6775b0	app	"python app.py"	less than a second ago	Up 5 seconds	0.0.0.0:5000->5000/tcp	compassionate_beldyts

```

kunal@stephen-kunal@kunal:~$

```

