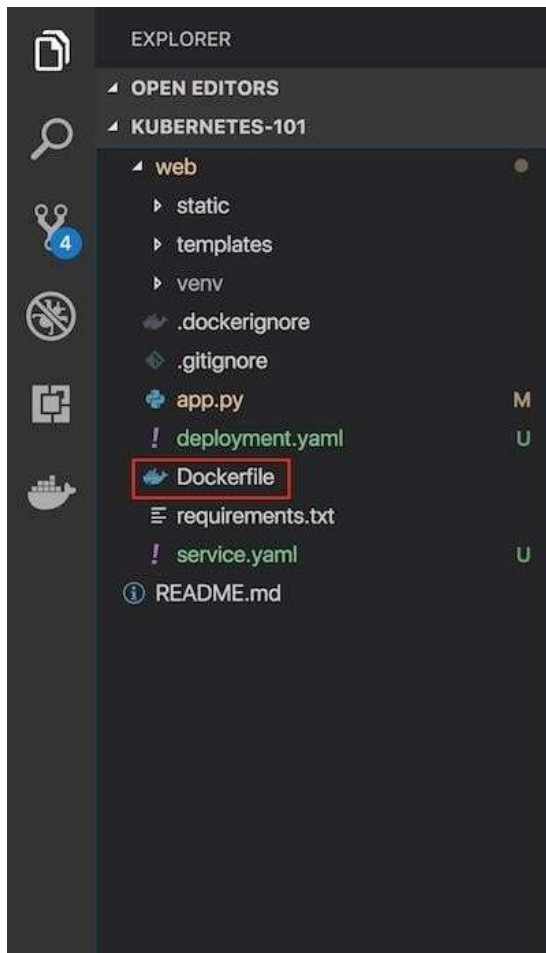


## Containerize your Flask application

Project Name	Project -Skill / Job Recommender Application
Team ID	PNT2022TMID18265

- 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

13. We want to expose the port(5000) the Flask application runs on, so we use EXPOSE.

## 14. EXPOSE 5000

Show more

15. ENTRYPOINT specifies the entrypoint of your application.

16. ENTRYPOINT [ "python" ]

17. CMD [ "app.py" ]

Show more

## 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 apps top level directory). For example: `docker build -t app:latest` .

```
kunals@btp:web kunalmlhotra$ docker build -t app:latest .
Sending build context to Docker daemon 348.24kB
Step 1/8 : FROM python:2.7
--> 6c76e967cfe
Step 2/8 : LABEL maintainer="Kunal Malhotra, kunal.mlhotra@ibm.com"
--> Using cache
--> d8b57d41591c
Step 3/8 : RUN apt-get update
--> Using cache
--> 6262a134e0e
Step 4/8 : COPY . /app
--> f07f737099f
Step 5/8 : WORKDIR /app
Removing intermediate container f9010b99a2fe
--> 0acc6a720e3d
Step 6/8 : RUN pip install -r requirements.txt
--> Running in 8153040b00b7
Collecting click==6.7 (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/34/cl/886f99713dd999c3366c362b2f908f18269f8a792aff1abfa700775a77/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/d7/08578774ed4536d324b14dcb469636634607af824e0997202ca8bd4b/Flask-1.0.2-py2.py3-none-any.whl (91kB)
Collecting itsdangerous==0.24 (from -r requirements.txt (line 3))
  Downloading https://files.pythonhosted.org/packages/dc/b4/d60cbdb945c00f66088d8975131ab3f29b22f2bcef1abb221165194b204/itsdangerous-0.24.tar.gz (46kB)
Collecting Jinja2==2.10 (from -r requirements.txt (line 4))
  Downloading https://files.pythonhosted.org/packages/7f/ff/ae64bdcff95f27a016a7bed8e868763ba4277a78ca76f32659220a731/Jinja2-2.10-py2.py3-none-any.whl (126kB)
Collecting MarkupSafe==1.0 (from -r requirements.txt (line 5))
  Downloading https://files.pythonhosted.org/packages/4d/0a/22a7f11db31608fde76808226d37001ef7e448255d9699ab9fcbdf4172b/MarkupSafe-1.0.tar.gz
Collecting Werkzeug==0.14.1 (from -r requirements.txt (line 6))
  Downloading https://files.pythonhosted.org/packages/20/c4/12b3e56473e52375a23c4764e70d1b8f3ef66682bef840a0e84fe335243/Werkzeug-0.14.1-py2.py3-none-any.whl (322kB)
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/2c/4a/61/5599631c154768c629b08c02c7267317910374ca082ff1e5
  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/33/56/20/bbe49a5c612ffefc5a632140a16596f964676768661e4e46
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 8153040b00b7
--> 85da830d97be
Step 7/8 : ENTRYPOINT [ "python" ]
--> Running in bdc1c83815e1
Removing intermediate container bdc1c83815e1
--> 73cef38ac1c
Step 8/8 : CMD [ "app.py" ]
--> Running in a784d430dd6f
Removing intermediate container a784d430dd6f
--> d8b083763a5
Successfully built app:latest
Successfully tagged app:latest
kunals@btp:web kunalmlhotra$
```

## Run your container locally and test

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

```
kunals@btp:web kunalmlhotra$ docker run -d -p 5000:5000 app
3c2b0f867383b969608652a2ef309a040a088263137ca543c9a6c616247
kunals@btp:web kunalmlhotra$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3c2b0f86738	app	"python app.py"	Less than a second ago	Up 5 seconds	0.0.0.0-5000->5000/tcp	compassionate_jellyfish

```
kunals@btp:web kunalmlhotra$
```

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