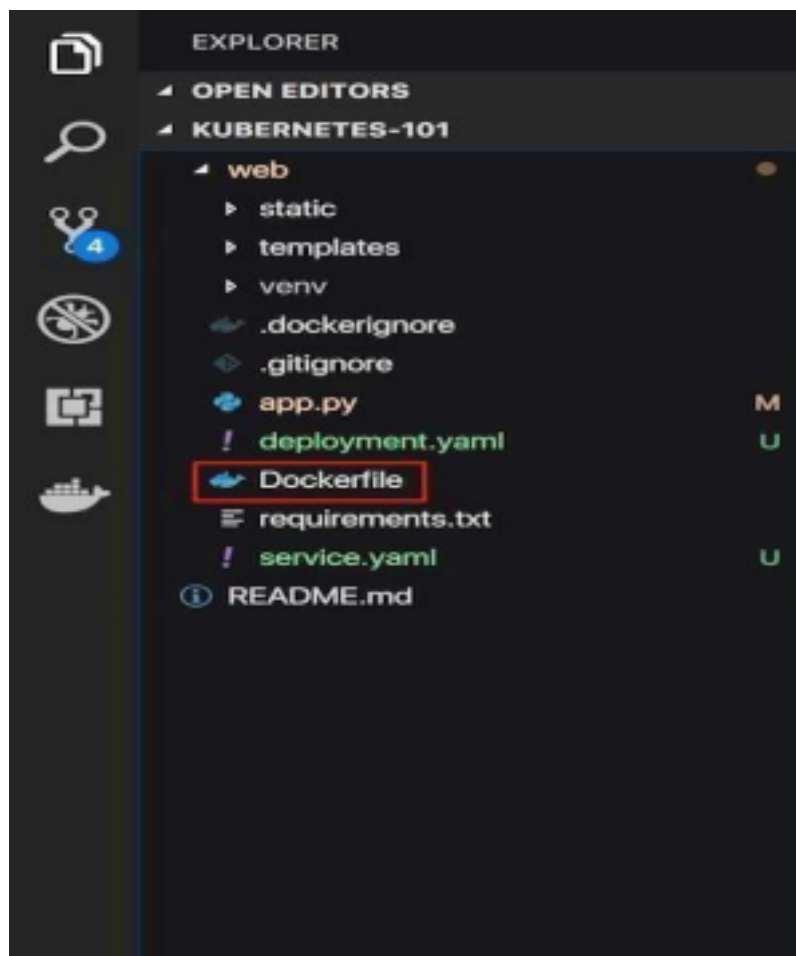


Containerize the App

Team ID	PNT2022TMID48406
Project Name	Nutrition AssistantApplication

Containerize your flask application

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



A "Docker file" 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" ]

```

Explanation and Breakdown of above Docker File

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 Docker File

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.`

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.

```
kunda@kunda-rtb:~/kunda$ docker run -d -p 5000:5000 app
3c28ef86f7548e5a0006e5c0de7339e04486a81703137c0543c9a056247
kunda@kunda-rtb:~/kunda$ docker ps
CONTAINER ID        IMAGE                COMMAND                  CREATED              STATUS              PORTS               NAMES
3c28ef86f754      app                  "python app.py"         Less than a second ago   Up 5 seconds        0.0.0.0:5000->5000/tcp   compositionone_jollysh
kunda@kunda-rtb:~/kunda$
```

