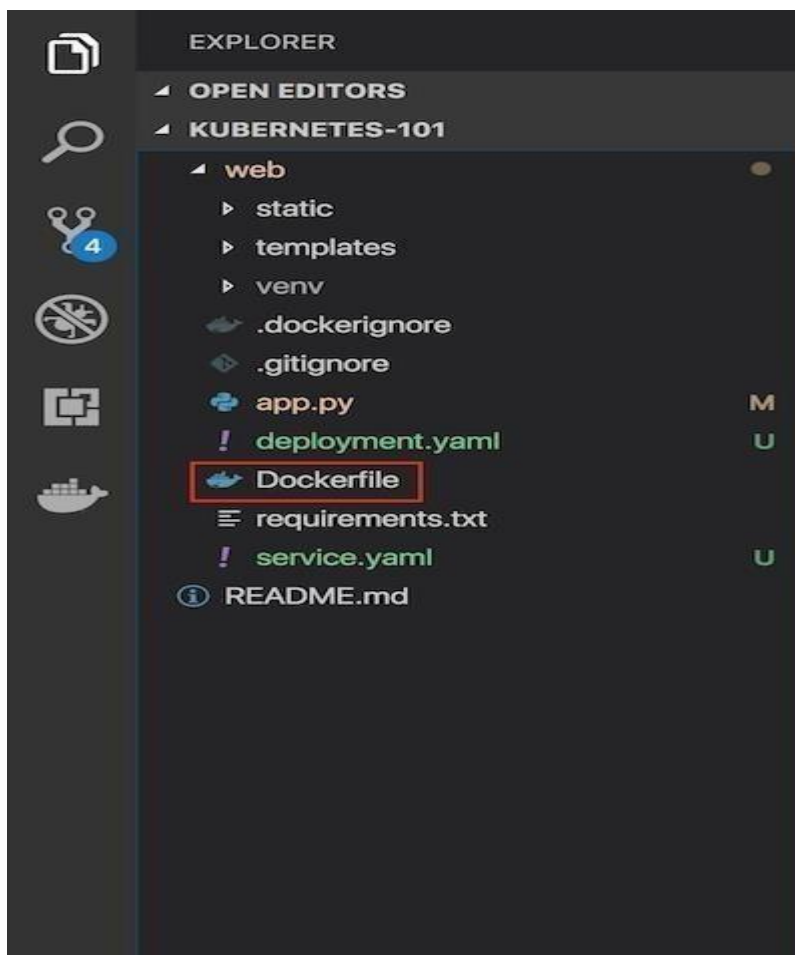


## Containerize the App

Date	18 November 2022
Team ID	PNT2022TMID21269
Project Name	Plasma Donor 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="Santhosh, santhosh@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 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="Santhosh, santhosh@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

```
root@gokul: /home/gokul/dev/application
File Actions Edit View Help
gokul@gokul: ~/dev x root@gok...lication x gokul@go...lication x

(root@gokul)-[/home/gokul/dev/application]
# ibmcloud cr login
Logging 'docker' in to 'icr.io'...
Logged in to 'icr.io'.

OK

(root@gokul)-[/home/gokul/dev/application]
# docker push icr.io/plasma/gokulap/plasma-donor-app:v1
The push refers to repository [icr.io/plasma/gokulap/plasma-donor-app]
1f8f34a4b5f9: Pushed
8b057373df61: Pushed
c373926928f1: Pushed
7be0bc584390: Pushed
aa4c808c19f6: Pushed
8ba9f690e8ba: Pushed
3e607d59ef9f: Pushed
1e18e7e1fcc2: Pushed
c3a0d593ed24: Pushed
26a504e63be4: Pushed
8bf42db0de72: Pushed
31892cc314cb: Pushed
11936051f93b: Pushed
v1: digest: sha256:b8a6ab6b86800b3f23f3a8c586a050e2b4fa1932ec47df9358ffde6584a42643 size: 3052

(root@gokul)-[/home/gokul/dev/application]
#
```

```
root@gokul: /home/gokul/dev/application
File Actions Edit View Help
gokul@gokul: ~/dev x root@gok...lication x gokul@go...lication x

OK

(root@gokul)-[/home/gokul/dev/application]
# ibmcloud plugin install container-service
Looking up 'container-service' from repository 'IBM Cloud'...
Plug-in 'container-service[kubernetes-service/ks] 1.0.459' found in repository 'IBM Cloud'
Plug-in 'container-service[kubernetes-service/ks] 1.0.459' was already installed. Do you want to re-install it or not? [y/N] > y
Attempting to download the binary file...
26.63 MiB / 26.63 MiB [=====] 100.00% 3s
27922432 bytes downloaded
Installing binary...

OK
Plug-in 'container-service 1.0.459' was successfully installed into /root/.bluemix/plugins/container-service. Use 'ibmcloud plugin show container-service' to show its details.

(root@gokul)-[/home/gokul/dev/application]
# ibmcloud plugin install observe-service
Looking up 'observe-service' from repository 'IBM Cloud'...
Plug-in 'observe-service 1.0.82' found in repository 'IBM Cloud'
Plug-in 'observe-service[ob] 1.0.82' was already installed. Do you want to update it with 'observe-service 1.0.82' or not? [y/N] > y
Attempting to download the binary file...
13.23 MiB / 13.23 MiB [=====] 100.00% 2s
13869056 bytes downloaded
Installing binary...

OK
Plug-in 'observe-service 1.0.82' was successfully installed into /root/.bluemix/plugins/observe-service. Use 'ibmcloud plugin show observe-service' to show its details.
```

## Build an image from the Dockerfile

## Run your container locally and test

After you build your image succesfully, 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@kali:~/web$ docker run -d -p 5000:5000 app
3c2b0f86f758e9a60606b52a2ef389ea8400e88263137ca5543c60c616247
kunal@kali:~/web$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3c2b0f86f758	app	"python app.py"	Less than a second ago	Up 5 seconds	0.0.0.0:5000->5000/tcp	compassionate_keldysh

```
kunal@kali:~/web$
```

