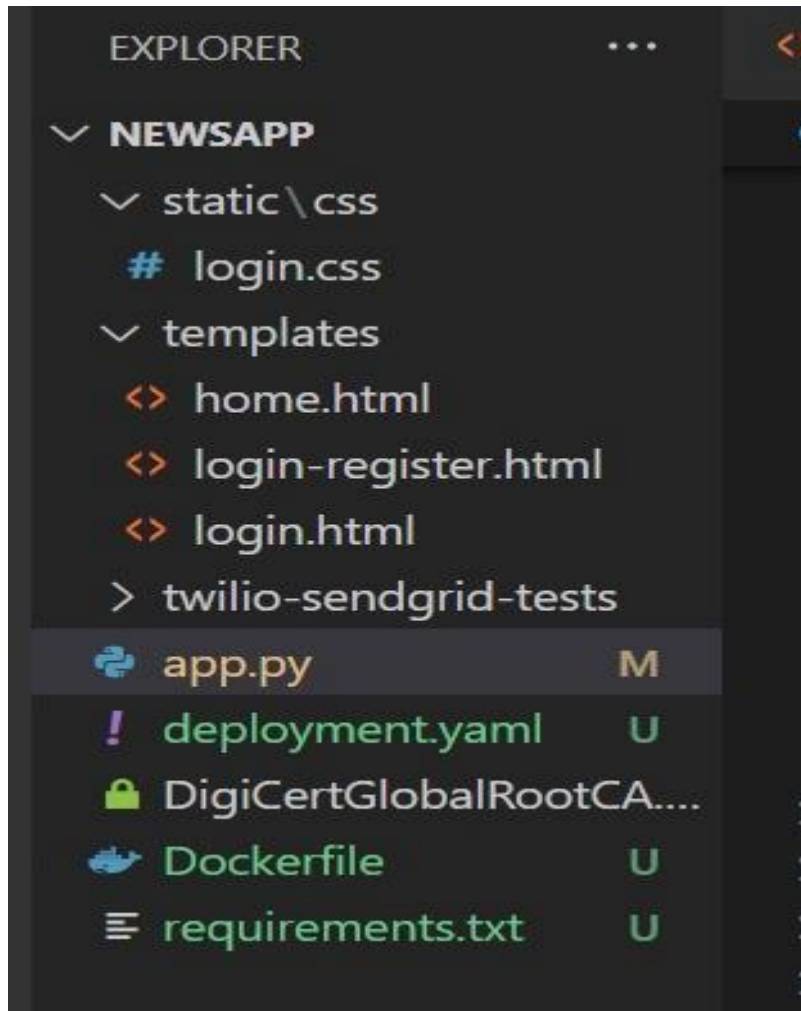


## Containerize your Flask application

- 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:

```
requirements - Notepad
File Edit Format View Help
Flask
ibm_db
sendgrid
```

Dockerfile - Notepad

```
File Edit Format View Help
FROM python:3.6
WORKDIR /app
ADD . /app
COPY requirements.txt /app
RUN python3 -m pip install -r requirements.txt
RUN python3 -m pip install ibm_db
EXPOSE 5000
CMD ["python","app.py"]
```

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: jobportal
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: flasknode
10   template:
11     metadata:
12       labels:
13         app: flasknode
14     spec:
15       containers:
16         - name: flasknode
17           image: icr.io/ibmtest1/NewsApp
18           imagePullPolicy: Always
19           ports:
20             - containerPort: 5000
```

## 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-mbp:web kunalmalhotra$ docker build -t app:latest .
Sending build context to Docker daemon 348.2kB
Step 1/8 : FROM python:2.7
--> 6c76e39e7cfe
Step 2/8 : LABEL maintainer="Kunal Malhotra, kunal.malhotra@ibm.com"
--> Using cache
--> d8b57d41591c
Step 3/8 : RUN apt-get update
--> Using cache
--> 6262a134e40e
Step 4/8 : COPY . /app
--> f07f7377099f
Step 5/8 : WORKDIR /app
Removing intermediate container f9010b99d2fe
--> 0bcc6af20e3d
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/c1/8806f99713ddb993c5366c362b2f908f18269f8a792aff1abfd700775a77/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/e7/08578774ed4536d3242b14dab4696386634607af824ed997202cd0bed4b/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/a60ccaba945c00fd6d08d8975131ab3f25b22f2bcfe1dab221165194b2d4/itsdangerous-0.24.tar.gz (46kB)
Collecting Jinja2==2.10 (from -r requirements.txt (line 4))
  Downloading https://files.pythonhosted.org/packages/7f/ff/ae64bacdfc95f27a016a7bed8e8686763ba4d277a78ca76f32659220a731/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/de/32d741db316d8f6b7680822dd37001ef7a448255de9699ab4bfcdbf4172b/MarkupSafe-1.0.tar.gz
Collecting Werkzeug==0.14.1 (from -r requirements.txt (line 6))
  Downloading https://files.pythonhosted.org/packages/20/c4/12e3e56473e52375aa29c4764e70d1b8f3efa6682bef8d0aae04fe335243/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/61/5599631c1554768c6290b08c02c72d7317910374ca602ff1e5
  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/eb4e9a5c612fffe1c5a632146b16596f9e64676768661e4e46
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
--> 66d2636b97bc
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
--> d8b6b83763a5
Successfully built d8b6b83763a5
Successfully tagged app:latest
kunals-mbp:web kunalmalhotra$
```

## 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.

```
kunals-mbp:web kunalmalhotra$ docker run -d -p 5000:5000 app
3c2bbf86f758b9a60606b52a2ef389ea8400eb88263137ca5543c60c616247
kunals-mbp:web kunalmalhotra$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3c2bbf86f758	app	"python app.py"	Less than a second ago	Up 5 seconds	0.0.0.0:5000->5000/tcp	compassionate_kelajsh

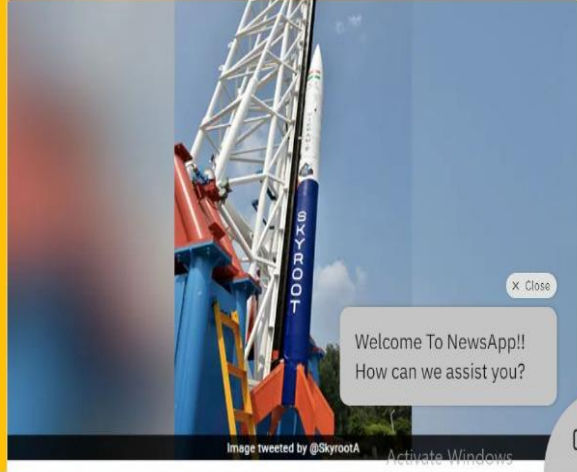
```
kunals-mbp:web kunalmalhotra$
```

# Welcome To NewsApp

Enter keyword ....

Search

## Today's Headlines



X Close

Welcome To NewsApp!!  
How can we assist you?

Image tweeted by @SkyrootA

Activate Windows

Go to Settings to activate Windows