Caden Hong

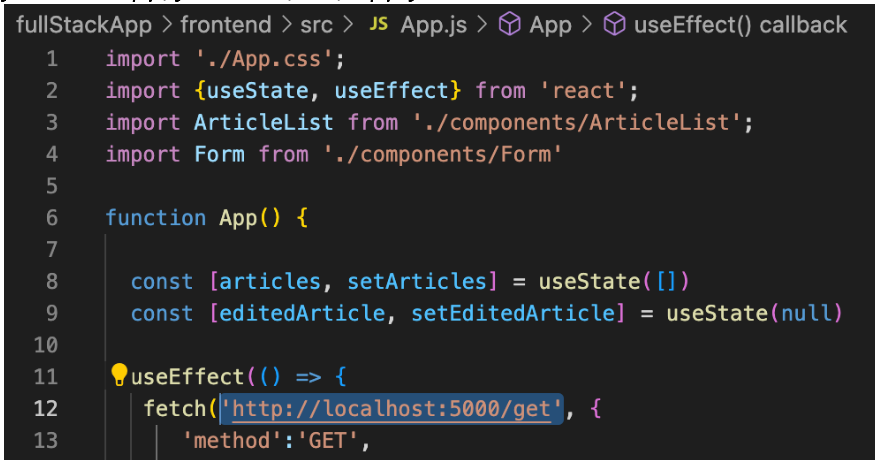
Using Docker Compose and Dockerfiles to Deploy a Full Stack App

Setup

In a project directory called *fullStackApp*, create two sub-directories called *frontend* and *backend* and upload necessary application files inside each of the corresponding folder

For *frontend*, make sure the URLs inside *App.js* and *APIService.js* files are correct (i.e. pointing to http://localhost:5000/ since that's where backend will be):

fullStackApp/frontend/src/App.js



fullStackApp > frontend > src > JS App.js > App > useEffect() callback

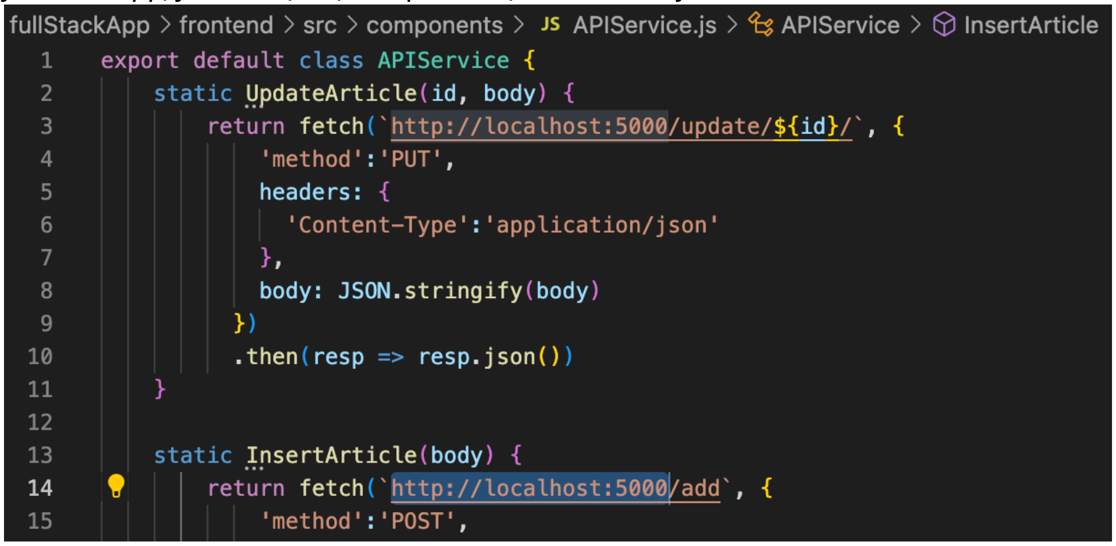
```
import './App.css';
import {useState, useEffect} from 'react';
import ArticleList from './components/ArticleList';
import Form from './components/Form'

function App() {

  const [articles, setArticles] = useState([])
  const [editedArticle, setEditedArticle] = useState(null)

  useEffect(() => {
    fetch('http://localhost:5000/get', {
      'method':'GET',
```

fullStackApp/frontend/src/components/APIServices.js



fullStackApp > frontend > src > components > JS APIService.js > APIService > InsertArticle

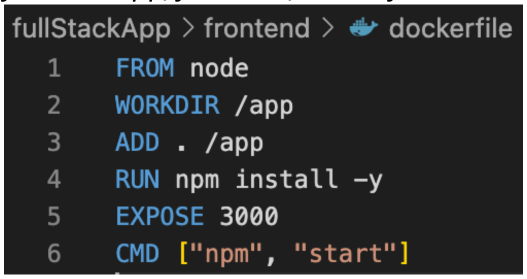
```
export default class APIService {
    static UpdateArticle(id, body) {
        return fetch(`http://localhost:5000/update/${id}/`, {
            'method':'PUT',
            headers: {
              'Content-Type':'application/json'
            },
            body: JSON.stringify(body)
          })
          .then(resp => resp.json())
    }

    static InsertArticle(body) {
        return fetch(`http://localhost:5000/add`, {
            'method':'POST',
```

Dockerfile

In each of the folders, create a *dockerfile* to build frontend and backend images:

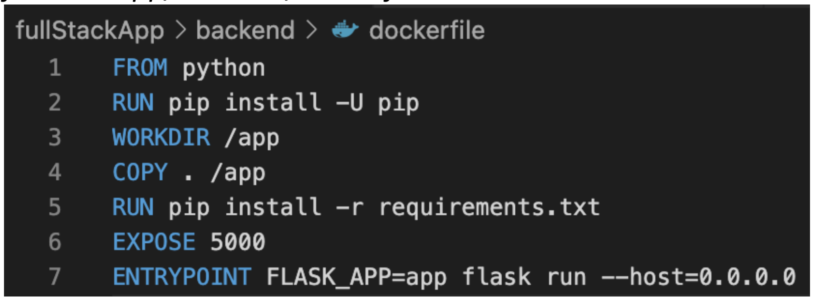
fullStackApp/frontend/dockerfile



fullStackApp > frontend > dockerfile

```
FROM node
WORKDIR /app
ADD . /app
RUN npm install -y
EXPOSE 3000
CMD ["npm", "start"]
```

fullStackApp/backend/dockerfile



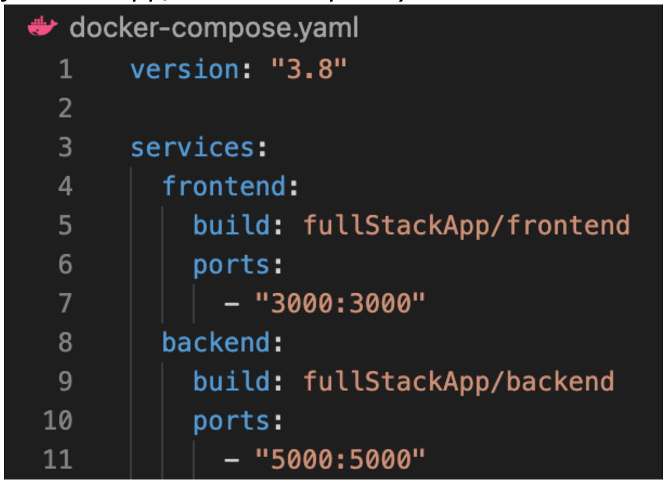
fullStackApp > backend > dockerfile

```
FROM python
RUN pip install -U pip
WORKDIR /app
COPY . /app
RUN pip install -r requirements.txt
EXPOSE 5000
ENTRYPOINT FLASK_APP=app flask run --host=0.0.0.0
```

Docker Compose

Then, create *docker-compose.yaml* at the project directory level:

fullStackApp/docker-compose.yaml



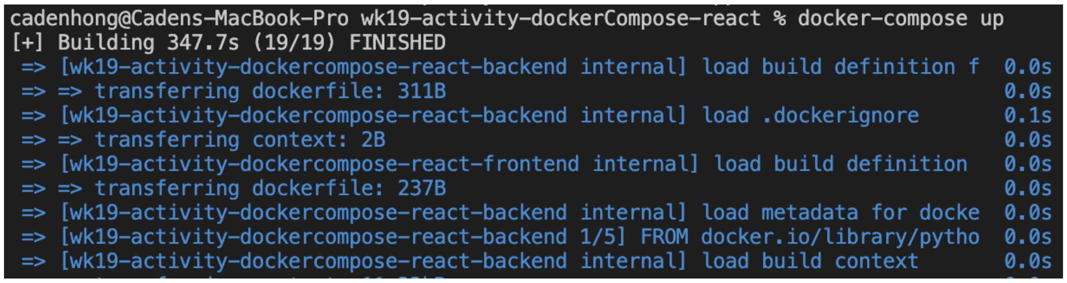
docker-compose.yaml

```yaml
1  version: "3.8"
2
3  services:
4    frontend:
5      build: fullStackApp/frontend
6      ports:
7        - "3000:3000"
8    backend:
9      build: fullStackApp/backend
10     ports:
11       - "5000:5000"
```

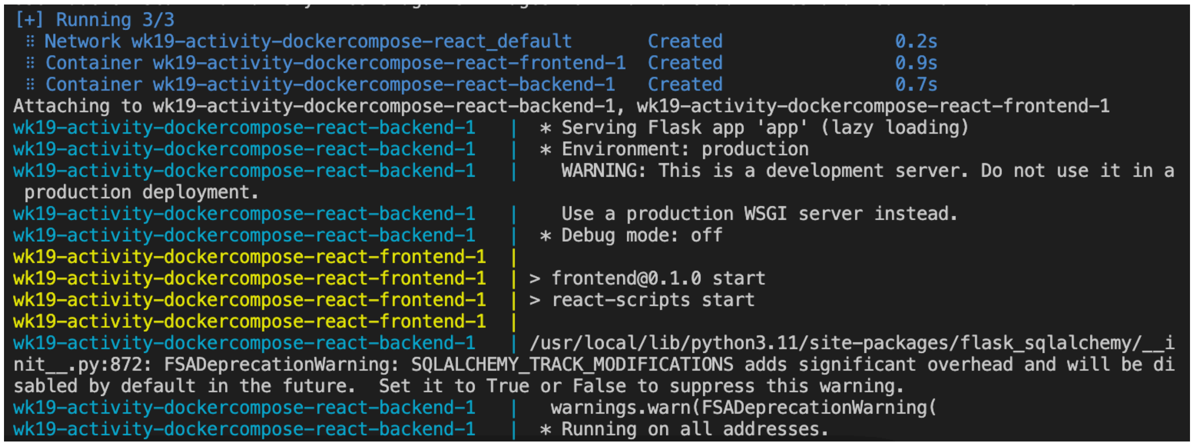
- Lines 5 and 9 indicates the location of *dockerfile*, which contains the build configuration for creating container images
- Lines 6-7 and 10-11 indicates the mapping of host port to container port

Build Images and Containers

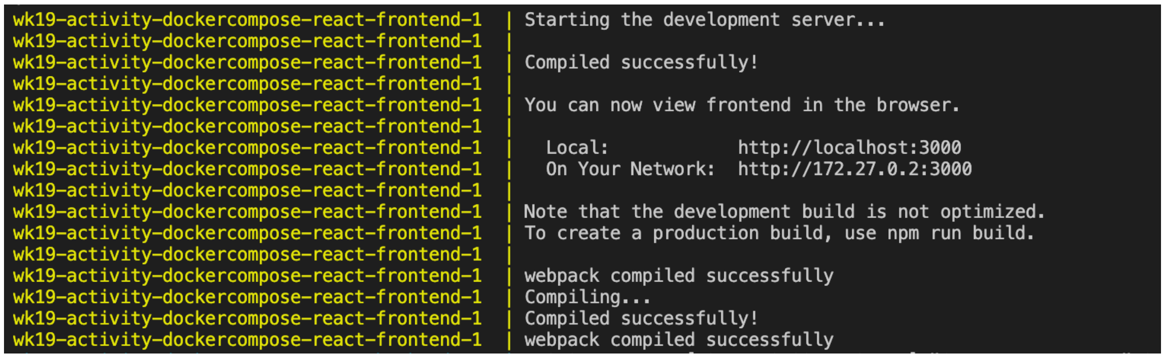
To create the images, build the containers and then start the containers, run the command ***docker-compose up***:



```
cadenhong@Cadens-MacBook-Pro wk19-activity-dockerCompose-react % docker-compose up
[+] Building 347.7s (19/19) FINISHED
 => [wk19-activity-dockercompose-react-backend internal] load build definition f  0.0s
 => => transferring dockerfile: 311B                                              0.0s
 => [wk19-activity-dockercompose-react-backend internal] load .dockerignore       0.1s
 => => transferring context: 2B                                                   0.0s
 => [wk19-activity-dockercompose-react-frontend internal] load build definition   0.0s
 => => transferring dockerfile: 237B                                              0.0s
 => [wk19-activity-dockercompose-react-backend internal] load metadata for docke  0.0s
 => [wk19-activity-dockercompose-react-backend 1/5] FROM docker.io/library/pytho  0.0s
 => [wk19-activity-dockercompose-react-backend internal] load build context       0.0s
```



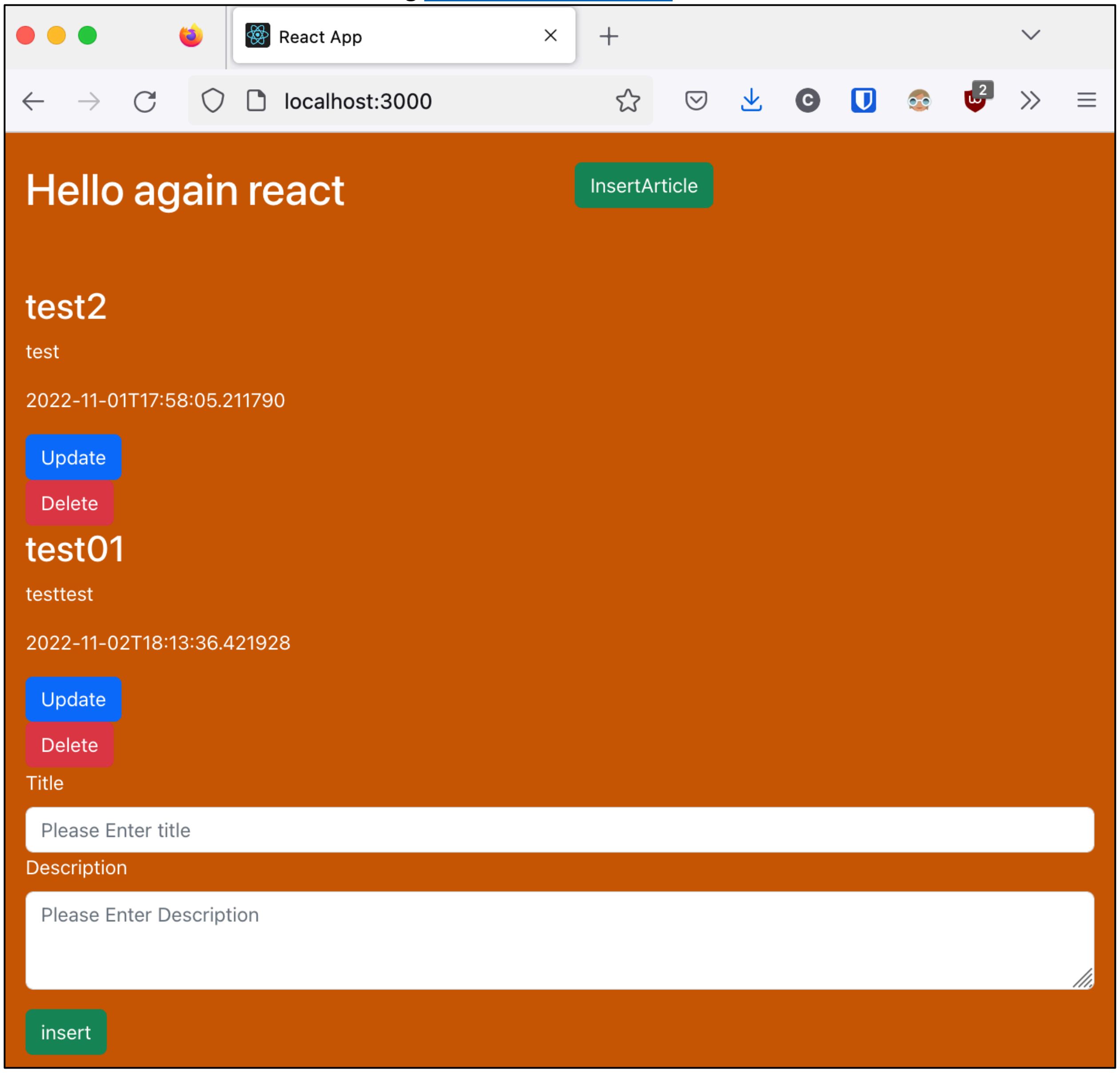
```
[+] Running 3/3
 ⠿ Network wk19-activity-dockercompose-react_default       Created                0.2s
 ⠿ Container wk19-activity-dockercompose-react-frontend-1  Created                0.9s
 ⠿ Container wk19-activity-dockercompose-react-backend-1   Created                0.7s
Attaching to wk19-activity-dockercompose-react-backend-1, wk19-activity-dockercompose-react-frontend-1
wk19-activity-dockercompose-react-backend-1   |  * Serving Flask app 'app' (lazy loading)
wk19-activity-dockercompose-react-backend-1   |  * Environment: production
wk19-activity-dockercompose-react-backend-1   |    WARNING: This is a development server. Do not use it in a
 production deployment.
wk19-activity-dockercompose-react-backend-1   |    Use a production WSGI server instead.
wk19-activity-dockercompose-react-backend-1   |  * Debug mode: off
wk19-activity-dockercompose-react-frontend-1  |
wk19-activity-dockercompose-react-frontend-1  | > frontend@0.1.0 start
wk19-activity-dockercompose-react-frontend-1  | > react-scripts start
wk19-activity-dockercompose-react-frontend-1  |
wk19-activity-dockercompose-react-backend-1   | /usr/local/lib/python3.11/site-packages/flask_sqlalchemy/__i
nit__.py:872: FSADeprecationWarning: SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and will be di
sabled by default in the future.  Set it to True or False to suppress this warning.
wk19-activity-dockercompose-react-backend-1   |   warnings.warn(FSADeprecationWarning(
wk19-activity-dockercompose-react-backend-1   |  * Running on all addresses.
```



```
wk19-activity-dockercompose-react-frontend-1  | Starting the development server...
wk19-activity-dockercompose-react-frontend-1  |
wk19-activity-dockercompose-react-frontend-1  | Compiled successfully!
wk19-activity-dockercompose-react-frontend-1  |
wk19-activity-dockercompose-react-frontend-1  | You can now view frontend in the browser.
wk19-activity-dockercompose-react-frontend-1  |
wk19-activity-dockercompose-react-frontend-1  |   Local:            http://localhost:3000
wk19-activity-dockercompose-react-frontend-1  |   On Your Network:  http://172.27.0.2:3000
wk19-activity-dockercompose-react-frontend-1  |
wk19-activity-dockercompose-react-frontend-1  | Note that the development build is not optimized.
wk19-activity-dockercompose-react-frontend-1  | To create a production build, use npm run build.
wk19-activity-dockercompose-react-frontend-1  |
wk19-activity-dockercompose-react-frontend-1  | webpack compiled successfully
wk19-activity-dockercompose-react-frontend-1  | Compiling...
wk19-activity-dockercompose-react-frontend-1  | Compiled successfully!
wk19-activity-dockercompose-react-frontend-1  | webpack compiled successfully
```

Caden Hong

Access the frontend container using http://localhost:3000:



Hello again react

test2

test

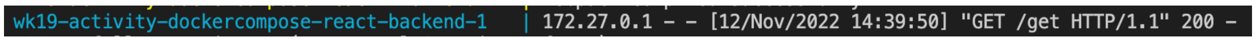
2022-11-01T17:58:05.211790

test01

testtest

2022-11-02T18:13:36.421928

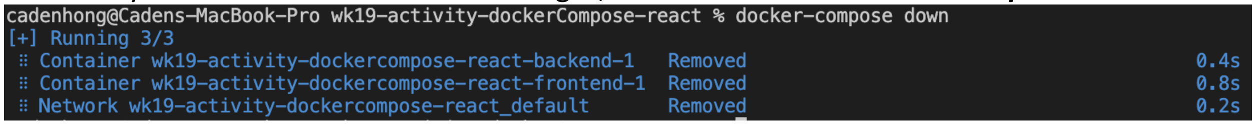
Log shows that the GET request to backend data was successful:



```
wk19-activity-dockercompose-react-backend-1   | 172.27.0.1 - - [12/Nov/2022 14:39:50] "GET /get HTTP/1.1" 200 -
```

Clean Up

To destroy the created containers and images, run the command ***docker-compose down***:



```
cadenhong@Cadens-MacBook-Pro wk19-activity-dockerCompose-react % docker-compose down
[+] Running 3/3
 ⠿ Container wk19-activity-dockercompose-react-backend-1   Removed                                          0.4s
 ⠿ Container wk19-activity-dockercompose-react-frontend-1  Removed                                          0.8s
 ⠿ Network wk19-activity-dockercompose-react_default       Removed                                          0.2s
```

****
Notice that there is a network that gets created with ***docker-compose up*** and deleted with ***docker-compose down***:

Docker Compose Documentation:

> *By default Compose sets up a single network for your app. Each container for a service joins the default network and is both reachable by other containers on that network, and discoverable by them at a hostname identical to the container name.*

****