Web Service and Cloud Based - 2024 February

The project is structured as follows.

- wscb
  - docker-compose.yml
  - README.md
  - **Authentication_Service**
    - app.py
    - models.py
    - mysql_config.py
    - utils.py
    - Dockerfile
    - wait-for-it.sh
    - requirements.txt
  - **Url_Shorten_Service**
    - app.py
    - models.py
    - mysql_config.py
    - utils.py
    - Dockerfile
    - wait-for-it.sh
    - requirements.txt
  - **mysql**
    - Dockerfile
    - init_db.sql
  - **nginx**
    - Dockerfile
    - nginx.conf
  - test
    - A bunch of test scripts of Canvas
  - docs
    - A bunch of assignment descriptions of Canvas
    - Reports
  - deprecate

> - Codes no longer used

Container Virtualization

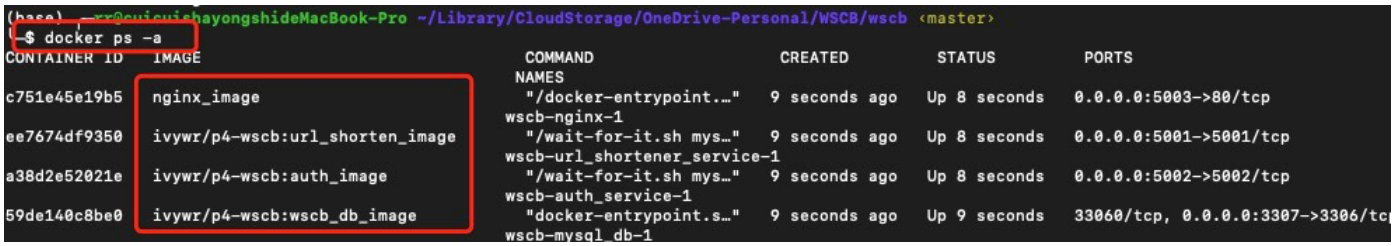
This part is about to start two Flask applications (Authentication_Service and Url_Shorten_Service) and MySQL database containers using Docker Compose and let the containers to communicate with database and the database data to persist.

How to Run Demo with Docker Compose

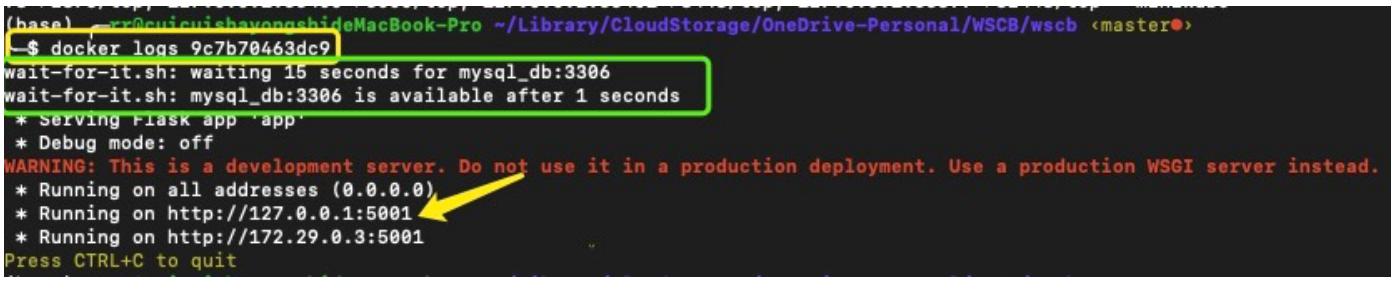
Navigate to the project directory (that has docker-compose.yml there)and execute the following command:

```
docker-compose up -d
```


```
docker ps
```




```
docker logs <url_shorten_container_id>
```



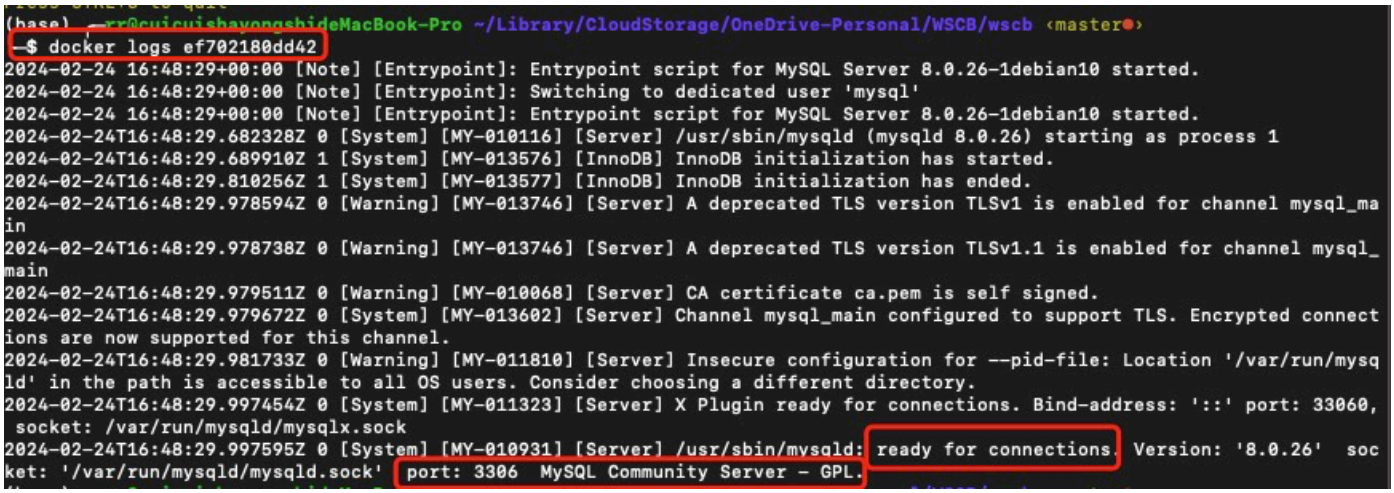
By checking the container logs, you can see that the url shorten service has been started on port 5001 (which we specified).

Inside the green box, it shows that we used the *wait-for-it.sh* script before running the python command to start the flask application. The reason for using this third-party wait script in the startup command of the Flask app is to **wait for the database port to become available and then start flask application**.

> You can get wait-for-it.sh by executing this on command line:
>
> wget https://raw.githubusercontent.com/vishnubob/wait-for-it/master/wait-for-it.sh

Similarly you can see authentication service has been started on port 5002 (which we speicified).

You can also check logs of MySQL database and see the mysql starts on port 3306. Note that this port can only be accessed within the containers (can only access by URL_shorten_service_container and Authentication_service_container).



You can use Postman to test by sending requests to http://0.0.0.0:5003/ for url shorten service and http://0.0.0.0:5003/users/ for identity authentication service. We use **nginx proxy** to make two services available on one single entry.

To remove the containers started by docker-compose up, you can use

```
docker-compose down
```