

# 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. We use **nginx proxy** to make two services available on one single entry.

## 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
```

```
(base) ~rr@cuicuishayongshideMacBook-Pro ~/Library/CloudStorage/OneDrive-Personal/WSCB/wscb <master>
$ docker-compose up -d
[+] Building 0.0s (0/0)
[+] Running 5/5
✓ Network wscb_default Created 0.0s
✓ Container wscb-mysql_db-1 Started 0.0s
✓ Container wscb-url_shortener_service-1 Started 0.0s
✓ Container wscb-auth_service-1 Started 0.0s
✓ Container wscb-nginx-1 Started 0.0s
```

```
docker ps
```

```
(base) ~rr@cuicuishayongshideMacBook-Pro ~/Library/CloudStorage/OneDrive-Personal/WSCB/wscb <master>
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND NAMES	CREATED	STATUS	PORTS
c751e45e19b5	nginx_image	"/docker-entrypoint..." wscb-nginx-1	9 seconds ago	Up 8 seconds	0.0.0.0:5003->80/tcp
ee7674df9350	ivywr/p4-wscb:url_shorten_image	"/wait-for-it.sh mys..." wscb-url_shortener_service-1	9 seconds ago	Up 8 seconds	0.0.0.0:5001->5001/tcp
a38d2e52021e	ivywr/p4-wscb:auth_image	"/wait-for-it.sh mys..." wscb-auth_service-1	9 seconds ago	Up 8 seconds	0.0.0.0:5002->5002/tcp
59de140c8be0	ivywr/p4-wscb:wscb_db_image	"docker-entrypoint.s..." wscb-mysql db-1	9 seconds ago	Up 9 seconds	33060/tcp, 0.0.0.0:3307->3306/tcp

```
docker logs <url_shorten_container_id>
```

```
(base) ~rr@cuicuishayongshideMacBook-Pro ~/Library/CloudStorage/OneDrive-Personal/WSCB/wscb <master>
$ docker logs 9c7b70463dc9
wait-for-it.sh: waiting 15 seconds for mysql_db:3306
wait-for-it.sh: mysql_db:3306 is available after 1 seconds
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5001
* Running on http://172.29.0.3:5001
Press CTRL+C to quit
```

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 specified).

```
(base) ~rr@cuicuishayongshideMacBook-Pro ~/Library/CloudStorage/OneDrive-Personal/WSCB/wscb <master>
$ docker logs aa3ae2c3e726
wait-for-it.sh: waiting 15 seconds for mysql_db:3306
wait-for-it.sh: mysql_db:3306 is available after 1 seconds
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5002
* Running on http://172.29.0.4:5002
Press CTRL+C to quit
```

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).

```
(base) ~rr@cuicuishayongshideMacBook-Pro ~/Library/CloudStorage/OneDrive-Personal/WSCB/wscb <master>
$ docker logs ef702180dd42
2024-02-24 16:48:29+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.26-1debian10 started.
2024-02-24 16:48:29+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2024-02-24 16:48:29+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.26-1debian10 started.
2024-02-24T16:48:29.682328Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.26) starting as process 1
2024-02-24T16:48:29.689910Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2024-02-24T16:48:29.810256Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2024-02-24T16:48:29.978594Z 0 [Warning] [MY-013746] [Server] A deprecated TLS version TLSv1 is enabled for channel mysql_main
2024-02-24T16:48:29.978738Z 0 [Warning] [MY-013746] [Server] A deprecated TLS version TLSv1.1 is enabled for channel mysql_main
2024-02-24T16:48:29.979511Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2024-02-24T16:48:29.979672Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
2024-02-24T16:48:29.981733Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
2024-02-24T16:48:29.997454Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqlx.sock
2024-02-24T16:48:29.997595Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.26' socket: '/var/run/mysqld/mysql.sock' port: 3306 MySQL Community Server - GPL.
```

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
```

```
(base) ~rr@cuicuishayongshideMacBook-Pro ~/Library/CloudStorage/OneDrive-Personal/WSCB/wscb <master>
$ docker-compose down
[+] Running 5/4
✔ Container wscb-nginx-1 Removed
✔ Container wscb-url_shortener_service-1 Removed
✔ Container wscb-auth_service-1 Removed
✔ Container wscb-mysql_db-1 Removed
✔ Network wscb_default Removed
```