**In-class Activity 2**

Galgamuge Emmanuel Fernando

C0918066

CBD 3324: Containerization and Container Delivery

November 19, 2024

**Repository URL:** <https://github.com/RuFerdZ/cbd-3324-in-class-activity-02>

1. **Project Structure**

The below image shows the folder structure I have used for the project and its deployment.

The decrypted secrets and the encryption key is ignored while been pushed to the repository.

A screenshot of a computer

Description automatically generated

1. **Deploy the database**

The below image shows the deployment of all resources of the database in the ***db-ns*** namespace and also the deployment of the secret after decrypting it.

A screenshot of a computer program

Description automatically generated

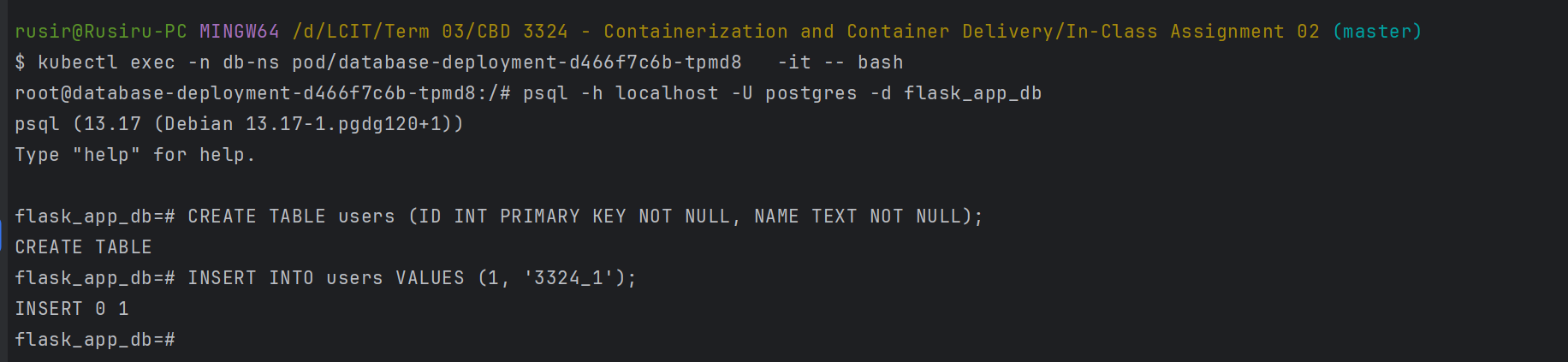
The below image shows all the resource created in the ***db-ns*** namespace.

A screenshot of a computer screen

Description automatically generated

1. **Set up the Database table.**

SSH to the database pod and then log in to the database as the *postgres* user, create the *users* table and insert a sample user.



1. **Deploy Redis Cache**

The below image shows the deployment of all resources of the Redis in the ***redis-ns*** namespace.

A screenshot of a computer

Description automatically generated

The below image shows all the resource created in the ***redis-ns*** namespace.

A screenshot of a computer screen

Description automatically generated

1. **Create Docker Image for Flask application**

Initially, a Dockerfile was created for the flask application.

A screenshot of a computer program

Description automatically generated

Next, I built and pushed it to DockerHub, (<https://hub.docker.com/repository/docker/ruferdz/ica-02-c0918066/general>) using the following commands.

* *docker build -t ruferdz/ica-02-c0918066:v2 .*
* *docker push ruferdz/ica-02-c0918066:v2*

A screenshot of a computer

Description automatically generated

1. **Deploy flask application**

The below image shows the deployment of all resources of the flask application in the ***app-ns*** namespace and also the deployment of the secret after decrypting it.

A screenshot of a computer program

Description automatically generated

The below image shows all the resource created in the ***app-ns*** namespace.

A screenshot of a computer screen

Description automatically generated

1. **Port forward flask application service to access it.**

The service serving the flask application is port-forwarded so that we can access it publicly (in this case the localhost) via port 5000 for testing.

A screenshot of a computer

Description automatically generated

1. **Access the endpoints**

* First, I accessed the root path- *http:localhost:5000*

A screenshot of a computer

Description automatically generated

* Next accessed the /users path - *http:localhost:5000*/users

This will return the users stored in the Postgres database.

A screenshot of a computer

Description automatically generated

* Next accessed the /cache path - *http:localhost:5000*/cache

If the caching works successfully, it will return the below message.

A screenshot of a computer

Description automatically generated