CMPT 756 Kube Squad/Square Term Project Report

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Team Name** | Kube Squad/Square |  |  |  |  |
| **GitHub project repo URL** | https://github.com/scp756-221/term-project-kube-squad/tree/music-service-port |  |  |  |  |
| **Team Member Name** | Aidan Vickars | Rishabh Kaushal | Anant Awasthy | Karthik Srinatha | Abhishek Nair |
| **Email (SFU)** | avickars@sfu.ca | rka73@sfu.ca | asa404@sfu.ca | ksa166@sfu.ca | asn4@sfu.ca |
| **Github ID** | avickars | rishabhkaushal07 | asa-404 | Karthiksrinatha | abs990 |
| **Additional Notes** | Currently the development branch we are using is "music-service-port". We will be changing to "main" once the project is complete. | | | | |

# Summary of Application

Due to the large number of members in our group, and general interest we developed three original micro-services that work together to form a playlist application with corresponding authentication and subscription services. The application is visualized below in Figure 1.

Graphical user interface, diagram

Description automatically generatedFigure 1: Playlist Application

The architecture of the application has been split into three separate micro-services that run on Kubernetes. Beginning with the Authentication service. This service allows users to create or login into their account respectively and subsequently access the Playlist microservice that will be described shortly. Like any normal application, accessing features often requires a paid subscription. Our application is no different, accessing the features in the playlist micro-service requires a paid subscription. We have implemented this in the Subscription service, where users after logging in can subscribe by adding a credit card that passes through a simulated validation. Once users have logged in and subscribed to the service, they can access the Playlist service that contains a variety of features. In this micro-service users can create, view, and edit playlists, as well as find information related to specific songs such as the lyrics, genre, and artist. All three services leverage Dynamo DB and independently query Dynamo DB to access information like users, song information, subscription status etc. Finally, from the user perspective, we have modified the given MCLI application to make HTTP requests to each micro-service. The MCLI application continues to run as a command line application.

# GitHub Guide

# Observations

# Analysis