

CMPT 756 Kube Squad/Square Term Project Report

Team Name	Kube Squad/Square				
CourSys team URL	https://coursys.sfu.ca/2022sp-cmpt-756-g1/groups/g-kube-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	risha-bhkaushal07	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, as well as general interest we developed three original micro-services that work together to form a playlist application with authentication and subscription services. The application is visualized below in Figure 1.

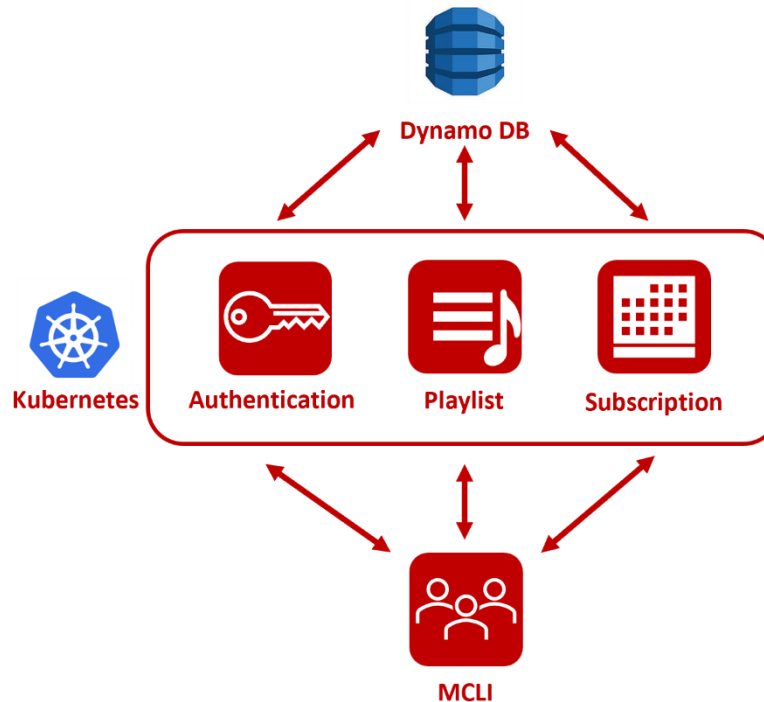


Figure 1: Playlist Application

As stated above, the application was designed to run as three containerized 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 service 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 service requires a paid subscription. We have implemented this in the Subscription service, where after logging in or registering for an account users can choose their preferred subscription option and subscribe by adding a credit card that passes through a simulated validation. Once users have logged in and subscribed, they can access the Playlist service that contains a variety of features. These features include creating, viewing, and editing playlists, as well as finding information related to specific songs such as the lyrics, genre, and artist. All three services use Dynamo DB for storage and independently query the database to access information like user accounts, 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

Beginning with the structure of the GitHub repository, the central branch is “music-service-port” that contains the most recent working build of the application. It is expected that any team member that merges their work into this branch, verifies that the branch remains in an error free condition. It should be noted that once the application is completed, the “music-service-port” branch will be merged

into the “main” branch. Furthermore, we utilize branching extensively during the development process. When a team member is actively working on a feature or bug, they create a new branch to use as their working copy and merge their work into the “music-service-port” branch when ready. This is reflected in the large number of branches that the repository contains that at the time of writing is 15.

To augment our development, we make extensive use of GitHub’s projects and log every item of work as an issue in the “756 Project” that contains our active kanban board. In this board, there are four separate columns that contain the backlog, to do, in progress and done items. In our typical workflow, team members will freely add issues to the backlog that likely need to be completed later. In our weekly scrum backlog items are moved into the to do column when as a team we feel they need to be completed in the coming sprint and are subsequently assigned to a team member. At the same time, other backlog items are often removed from the board entirely when the team decides they are obsolete.

Whenever a user embarks on a new piece of work, they move the corresponding issue into the “in progress” column or if the issue has not already been created, they create the issue first before continuing. Finally, once the issue is completed it is moved to the “done” column. At the time of writing there are 77 issues in the “done” column. It should be noted, we do attempt to link our commits to the relevant issues, however we have found that remembering to do this is rather difficult.

Observations

Reflection on Development

What did you observe from applying and using the scrum methodology? What worked well? What didn’t? What surprised you?

After applying and using scrum methodology, we made three significant observations. The first is that it allowed us to assess more accurately our currently velocity towards completing the project. Due to the nature of course work that often comes in waves, team members development capacity often significantly fluctuated week to week. By having weekly scrums, we were able to properly define what the definition of done for the upcoming sprint such that it was achievable. Furthermore, in the case a team members capacity changed on short notice (ex: assignment was pushed back), having weekly sprints allowed us to pivot quickly.

The second observation we made was that the consistent cadence that scrum often requires did not work. As mentioned above, team members development capacity would significantly fluctuate week to week. As a result, there were sprints where the entire team’s development capacity was effectively zero and the resulting sprint goal was nothing. Thus, the team was unable to maintain a consistent cadence that at times defeated the purpose of scrum because there was simply no point in having one since no one had capacity to perform any work.

The third and final observation we made was that it normalized the behaviour of demonstrating one’s work to the team. By applying scrum, it normalized the expectation of demonstrating the completed items from the last sprint. We found this to be particularly useful because it served as both an opportunity for everyone to demonstrate their work, but also as an opportunity to teach others how to utilize their completed work. An example of this was the containerization of the application where each service was ported to work as containers. After demonstrating the applications running as

containers, each team member became familiar with the process of building and running the application.

Reflect on the readings over the course of the term. What ideas were you able to apply? How did these turn out?

Beginning with the second reading that featured a comparison between mono and poly repositories, we applied a mono-repository to great success. Due to the size of the application, using a poly-repository would have created additional overhead in managing the location of each micro-service. In contrast, by using a mono-repository we were able to easily create a single source of truth that contained the application that facilitated rapid deployment and iteration. This, rapid deployment was further enhanced by the “infrastructure as code” ideas that were also presented in the second reading. To implement this, we wrote a comprehensive make-file that contains every command required to deploy the application in both a local and cloud format. This enabled a consistent deployment practice that eliminated all confusion around the deployment of the application. It should be noted that our use of containers and make files to facilitate the deployment of the application captured ideas from the third reading as well that discussed containerization and make-files.

TO BE DISCUSSED – KUBERNETES IMPLEMENTATION – READING 4 – While the application has been configured to run on Kubernetes, we are waiting to test the application under load before discussing this. This section will be completed in the final report.

Moving on to the seventh reading that discussed micro-services, we found implementing our application as three micro-services enabled rapid parallel development. Because of the isolated nature of each service, we were able to rapidly complete all of them in the span of approximately two weeks with minimal integration issues and in a continuous fashion as described in reading 8. By having each developer merge their work into the “music-service-port” branch, we were able to continually develop the application while maintaining a working and stable version.

If you have professional experience with scrum, how did your team perform in comparison to past teams?

Our team performed both well and poorly compared to our professional experiences with scrum. From a cadence perspective our team performed quite poorly. As discussed previously we were unable to maintain a consistent cadence that contrasts with our professional experiences. In our previous professional environments, our focus would often be on a single project and as a result we were able maintain a consistent cadence towards completing items in each sprint. On a positive note, during times when our team was actively developing, utilizing scrum enabled us to ensure there was no duplication of work. This is in stark contrast to some of our professional experiences where we would often perform a significant amount of work only to find later that there was existing work that could have easily been utilized instead. An example, of this is writing large portions of code only to find out that there was existing code base that could have been easily ported over to the current project.

Reflection on Operating the Application

We are currently configuring Gatling to apply load to our application. This is expected to be completed within this current sprint (by March 21). As a result, we cannot write this portion of the report yet. This will be completed in the final report.

Analysis

Scaling Analysis

We are currently configuring Gatling to apply load to our application, as well as Grafana to monitor our application. This is expected to be completed within this current sprint (by March 21). As a result, we cannot write this portion of the report yet. This will be completed in the final report.

Failure Analysis

We are currently configuring Gatling to apply load to our application, as well as Grafana to monitor our application. This is expected to be completed within this current sprint (by March 21). As a result, we cannot write this portion of the report yet. This will be completed in the final report.