Group 20 Distribution Plan

Aamir Baloch, Ryan Fontaine, Joshua Krug, Rose Trescott

Our project is a recipe management system application that involves the use of a Flask web server integration, MySQL database, and HTML. The first objective of distribution would be to properly containerize all the components of the application. Doing this ensures proper portability as well as minimal resource use. For this instance, we agreed that setting up a Docker container would have been the best choice because it would have been free, as well as our combined experience in using Docker for other projects. The Dockerfile would build the image of the Flask app, with all the dependencies and configuration as well as the access port. We would then have a separate Dockerfile for the SQL database containing the cache information, with all the data being stored in the database. We would then have some sort of cloud deployment now that we have it containerized, and we decided that an AWS cloud would probably have the smoothest integration. We would make a cluster to fit our instance and networking specifications and set up a registry to host our image securely. From this repository, we can have specific ports and variables set to fit our application to ensure proper access over a network. From this, we would have to figure out how to handle multiple instances of traffic through scaling and having high availability. We would also have to take into consideration, updates to the backend or front-end code and how those would be pushed to our application. After doing some research a CI/CD pipeline would be the best way to do this. We would integrate our GitHub to automatically rebuild the docker upon every commit or have scheduled refreshes intermittently. We would have to monitor traffic and the cost of the AWS but for this hypothetical and the duration of the assignment, we don’t think it would've been too expensive to host a cloud instance for such a short time.