Jasmine Casciola

Professor Alam

CS 470

April 24, 2024

CS 470 Final Reflection

Youtube Video: https://www.youtube.com/watch?v=nh1iP6q8me0

This course has helped me gain experience with overhauling a full stack application to a cloud framework with AWS. I also learned how to use multiple AWS tools like Lambda and API Gateway in order to ensure proper frontend-backend connections for my application. This has helped me gain confidence in my experience as a software developer. Some of my strengths as a software developer include the ability to visualize from an eagle's eye perspective, and also troubleshoot. Being able to see the application from the big picture helps with organizing the full stack application in the cloud and understanding where everything is and/or why it is being used. Troubleshooting helps ensure those frontend-backend connections. For example, one issue I had was Docker having trouble hosting my application. This was done with a minute mistake, where I simply used the wrong quotation mark. Some roles I am prepared to assume in a new job include interfacing with not only the application, but also the team, owner(s), and stakeholder(s) behind it. While this job is technical, it is just as important to ensure you are able to communicate and build relationships with the people behind it.

Microservices and serverless applications help move things quickly, although they come with a few caveats. One includes scale and error handling. I would ensure that our application

handles the growth or contraction of client-server demands by running regular maintenance checks to analyze the activity of the application. This could be done with Amazon CloudFront. Error handling can be done by programming the proper troubleshooting catches and codes. Cost could be predicted by analyses from Amazon CloudFront since it tracks activity and/or usage of the application. Between containerized and serverless applications, the latter is more cost effective in the long run since containers need to constantly run, whereas serverless operates on a pay-per-use model. Serverless applications only trigger when needed and otherwise act "dormant". Some pros and cons that would be deciding factors in plans for expansion include the amount of client-server demands, storage, and data being transmitted. Whenever data is at rest, in motion, or in use, this could affect the activity of the application, which results in the demand and speed. Any time these factors start to change, the developers and company should begin to think about expanding and/or contracting services for their application. Elasticity and pay-for-service help by allowing developers to expand or contract the backend of applications as needed. They do not need to reserve a certain amount of servers, since AWS and other similar services only charge based on use. Therefore it keeps overhead to a minimum while also being able to predict and prepare for future operational costs.