CS-470 Final Project Reflection

Sarah Jodrey

Southern New Hampshire University

CS-470: Full Stack Development II

Professor Nizar Dajani

June 24, 2023

CS-470 Final Project Reflection

Presentation - https://youtu.be/t8POfER_fPo

Experiences and Strengths

This course provided me with the hands-on experience of taking a full stack application to the cloud. I was able to utilize my strengths as a developer to learn new technologies and tools and to debug any issues that arose during the projects transition to the cloud. I am now familiar with containerization and utilizing Docker to improve development efficiency and contributing to other projects. Immense value was gained working with and securing the AWS S3 Bucket, Lambda, API Gateway, and DynamoDB as they've increased my marketability in the software development field. It has also prepared me to assume roles as a Serverless Developer or a Full Stack Developer.

Planning for Growth

One of the strong selling points for AWS is the variety of tools and services provided to help efficiently handle a project in the present and the future. Auto scaling is offered for services like API Gateway and Lambda so that the project dynamically adjusts to high or low usage/demand. For error handling, detailed monitoring and logging is offered via services like CloudWatch to provide usage, performance, and error insight. It's also a good idea to test or simulate any new features to ensure the scalability and error handling services are behaving as expected.

Fortunately, AWS also provides a tool for calculating costs before making any commitments. This provides valuable insight into whether serverless is the best option for the project. It can save a lot of time and headaches for the team by using the tool.

When it comes to the financial predictability of containers vs serverless, the answer isn't exactly hard and fast. Serverless offers the pay-for-use/pay-for-service model with the calculator tool to help estimate the costs for certain use cases. However, factors like complexity and resource optimization can result in unexpected costs. While Containers offer more fixed cost options, resource utilization, and control over infrastructure make them predictable to a certain extent as well. That being said, serverless will generally be the more predictable of the two.

There are a couple of important things to consider before deciding on expansion. These include things like does the demand for the product exist, will it continue to grow, and do you have the funding to handle such expansion. The beauty of using serverless is that the pay-for-service model means you only need to be concerned about whether you have the funding for increased usage. The calculator tool comes in handy here since you can get a rough estimate of what those costs will look like and if they are feasible.

As mentioned above**,** elasticity and the pay-for-service model play a vital role when deciding on plans for future growth. This is because they provide an economical and efficient way for a company to scale their resources and infrastructure. Companies, or individuals, won't need to worry about a large upfront investment so they are free to test out new ideas or make adjustments as their product/project evolves. Ultimately, serverless provides the freedom and support for businesses to innovate and grow in an economical fashion.