

Breunna Bingham

SNHU CS 470

10/15/2023

CS 470 Final Reflection

YouTube Presentation Link: <https://www.youtube.com/watch?v=p3h-0PvO9DY>

I have learned a lot from this course. From Full Stack Development I we started this web development adventure and ended it here in Full stack Development II, using AWS and cloud services. I had only heard about AWS prior to this course. This course has taught me so much about AWS and all of it's components and how a lot of them if not all are easier to use than the leading competitor. For example, I really enjoyed using DynamoDB over Mongo DB. The reason for this is that for me, it is much easier to use and very fast to deploy. All of the back-end work is done within AWS and there isn't a lot to configure like there would be for MongoDB, and this includes security layers. Paired with my quick learning ability, I believe learning about the cloud services that AWS has to offer will help me to find a job that utilizes AWS having had this experience.

With AWS scaling is automatically done whenever needed. They have a Pay-as-you-go model, which allows you to pay for only what you use. This can maximize savings that can be allocated to other portions of development. Utilizing internal services through AWS, such as their API Gateway, Lambda and their DynamoDB tables, one can get a seamless application up and running incredibly fast. Testing can be done with authorizations, and authentications, by revising the policies and permissions on individual roles. When it comes to cost, it depends on what the budget is, to determine if Containers or serverless would be the best option. Containers are more predictable because you pay for the container being used and can upgrade to more containers as you go, but at least the cost is more predictable. Serverless application costs can go up and down depending on usage. Think of this like an electric bill. If you use more electricity one month over the previous month, that bill will be much more costly than the previous month's bill. The same idea applies here. With more calls to the server through the API gateway, and function calls to lambda to access the database, each of those add to the bill, however the impressive thing about going serverless through AWS, is that it automatically scales up or down to allow you only to be changed what you are using. This also makes it very reliable to customers, because even if the traffic is high, the automatic scaling capability allows the API to handle any level of traffic while maintaining customer satisfaction. If a company wants to remain reliable to its customers and would like to scale their company up on a more frequent basis, then serverless might be the better option. Having the elasticity to scale up or down as needed can be a great benefit to a company that is just starting up and needs to maximize their costs by utilizing the pay-as-you-go models. The downside to this option is that the amount they would pay for these services would vary month to month.