Austin Scarinza

CS-470 Full Stack Development II

December 20, 2024

https://youtu.be/sZCuC9iS2AI

The skills I have learned, expanded upon, and mastered include but are not limited to, setting up and hosting S3 buckets and websites, Creating and mapping HTTP requests through API Gateway, applying business log through Lambdas, as well as security connecting to and manipulating DynamoDB via Lambda using secure IAM Roles and policies. Becoming more familiar with these cloud-based tools helps provide another tool alongside containerization that can be used to host serverless applications. When it comes to creating, monitoring, and testing software, my attention to detail and ability to stay on task are two of my prevalent qualities. These qualities will be valuable to many teams as they pertain to software design, development, testing, and security roles. I am eager to start using my skills as well as learning new ones in a entry level position where I can learn from experienced professionals.

Scaling microservices in a serverless environment is an important aspect of choosing to use microservices in the first place. Using microservices makes it easier to automate scaling up to meet traffic spikes when needed. Implementing error handling and tagging error prone areas would allow a team to better monitor certain areas of the application. Having fail safes in place such as adding additional zones or regions in a case where errors are common would be a good 'band-aid' fix until the main issue is addressed. Using serverless tools helps bring down costs when services are inactive. To get an accurate estimate of costs, I would use common metrics to gauge the average consumption of hardware, as well as the given costs associated with spikes and scaling up, adding that into the estimate based on time frames a how often these spikes happen. While containers may be more cost predictable due to the predetermined costs associated with traffic, serverless allows savings when applications slow down or become inactive.

When expanding while using a serverless environment, the whole process is made quite simple. The serverless environment is made to scale so powering up the entire process would not be too difficult; it could just end up costing more. With elasticity and pay-for-service helping our case, even if an application were to scale up 100x a few times a day and then scale back down to its normal pricing, there would be a significant increase in performance, but not in costs.