Adela Martinez

10/27/2024

CS 470 Final Reflection

YouTube link: https://youtu.be/8EXViKUHRyo

- ***Experiences and Strengths:*** *Explain how this course will help you in reaching your professional goals.*

In this course I learned how to successfully migrate a full stack web application to the cloud. I was able to develop my skills of working with Amazon Web Services. I learned how to create containers, use S3 buckets, DynamoDB database, and APIs. I've never worked with cloud development before this course and now I feel more comfortable and confident with cloud development. Even though I only have some experience with cloud development I feel that I would be a marketable candidate for a future job because it is better to have some experience than none.

As a software developer, one of my top strengths is learning a new programming language or environment quickly. My other strengths include being a team player when working within a group, not afraid to ask for help or research on the internet when I am having issues with a project I am working on, and I depend on trial and error until I can find the correct solution.

A role that I am interested in applying for would be project manager. I enjoy working in a team setting, sharing ideas and working through all the developmental process of creating an application and delivering the final product to the client.

- ***Planning for Growth:*** *Synthesize the knowledge you have gathered about cloud services.*

Both serverless and microservices enhance scalability, efficiency, and keep costs low. With serverless functions are used, like Lambda, that auto scale the amount of HTTP requests that come in from the users. Microservices components run independently where scaling is done based on command. Both serverless and microservices both have customed error messages for error handling. When it comes to predicting the costs, it is different between serverless and microservices. Serverless costs are based on the number of executions and time of those executions of the functions that are in use. Microservices costs are based on the CPU/ memory that is being used within the containers. Containers would be more cost-effective since you pay for the size and number of containers that are needed. The cost of serverless could surge if there is high traffic, making it harder to predict.

For plans of expansion, pros of serverless would be automatic scaling, pay-as-you-go pricing, and developers can focus on the code rather than the configuration and maintenance. While cons of serverless would be costs can vary depending on traffic, could experience delays when not used recently, and limited control and customization. Pros of microservices(containers) are portability, better predictability of costs, and more control over the infrastructure. Cons of microservices are some scaling issues, managing the infrastructure can be complex, and costs still incur while containers are idle. The deciding factor between serverless and microservices would depend on which pros and cons would be best for your needs and expansion goals.

Elasticity and pay-for-service allow for scalability, cost efficiency, and flexibility strategies that could help with future growth. Scalability allows for quicker adaptability to demand, being cost efficient allows for making better financial decisions, and the flexibility allows for long term planning.