Ayan Gilbert 08/22/2024 CS 470 Final Reflection YouTube Link

In terms of professional development, this course had helped me to understand microservices architecture and cloud development at a deeper level. By using real-world situations for the projects and gaining hands on experience, I was able to create a deploy an application on a cloud platform. I was able to complete this by using a serverless platform and learning how to manage cloud resources. As I begin to pursue a professional career, this knowledge will be crucial and the demand for modern application development increases.

Some of the skills I've obtained and perfected in the course include containerization using Docker, managing serverless functions using AWS lambda, and creating a secure cloud environment with AWS services like S3 and API Gateway. These are marketable and transferable skills that would help me to market myself as a candidate. Organizations are always seeking professionals who can not only develop, but also deploy and manage an efficient and scalable application. Even more specifically, using serverless technology to deploy in the cloud. This provides a strong foundation in full-stack development in general.

Now that I have the tools and knowlege to be able to utilize microservices and serverless architectures, I can create scalable and resilient web applications. The applications will be cost-effective and auto scaling, and I would handle error-handling by using built-in features in AWS.

In order to predict cost in a cloud environment, you have to analyze usage and make sure to choose the appropriate pricing models. By using a serverless architecture, you have the ability to predict the cost when there is a variable workload. This allows a client to pay for what they use for consistent and long-running processes.

When considering the possibility of expansion, you have to think about the benefit of the pricing structure vs the limitations in execution time. When you have a lower overhead cost, sometimes you run the risk of latency from cold starting the program. By using containers, you can have more control and make sure that a steady workload is more cost efficient, hwoever would require more management

Using a pay-for-service model is a huge deciding factor when considering future growth. By having elasticity in the model architecture, the scaling of the application can be automatic as demand shrinks or grows. This model will directly reflect the cost of actual usage, which is great for budgeting. This is a flexible solution that adapts based on the change in business demands