CS 470 Final Reflection

Dante Trisciuzzi

February 28th, 2024

Presentation: https://www.youtube.com/watch?v=87d3C62twFw

Experiences and Strengths

Throughout CS 470, I've learned and honed some vital skills relevant to cloud development, particularly in deploying full-stack web applications in the cloud. This course has equipped me with a better understanding of container orchestration with Docker Compose, serverless architecture with AWS Lambda, and secure cloud service integration. These skills are crucial for my aspiration to become a clou-first developer, enhancing my marketability in the tech industry. My strengths lie mostly outside of cloud development, in systems and architecture. This course has started to prepare me better for roles such as Cloud Developer, DevOps Engineer, and Cloud Architect, where I can leverage my new skills in cloud infrastructure and application development.

Planning for Growth

In planning for the theoretical future growth of my web application, I'd focus on using microservices and serverless architecture to ensure scalability and manageability. Microservices will allow me to deploy and scale parts of my application independently, enhancing agility. Serverless computing, especially, offers significant advantages for managing scale due to its inherent elasticity and cost efficiency.

- Scale and Error Handling: I would plan to use auto-scaling features of cloud services and implement error handling within my serverless functions to manage unexpected spikes in traffic and minimize downtime.

- Cost Prediction: Predicting costs in cloud environments requires understanding the pricing models of the services used. Serverless computing, with its pay-for-use model, offers more predictability for operational costs as it aligns directly with usage.
- Cost Predictability: Between containers and serverless, serverless computing often provides more cost predictability due to its direct correlation with function executions, making it easier to forecast costs based on request volume.
- Pros and Cons for Expansion: Expansion plans would weigh the flexibility and scalability of serverless against the control and consistency of containers. Serverless excels in rapid scaling and lower operational costs, while containers offer more control over the environment and are better suited for complex applications with specific requirements.
- Elasticity and Pay-for-Service: Elasticity ensures that the application can scale resources up or down based on demand, which is critical for handling growth. The pay-for-service model supports this by allowing costs to scale with usage, ensuring that we pay only for what we use. These factors are key in decision-making for future growth, allowing for better resource utilization and cost management.

In conclusion, this course has not only expanded my technical skill-set but also my ability to strategically plan for the scalability and sustainability of cloud-based applications, positioning me to contribute effectively to the evolving cloud computing landscape.