Chris Wong

June 30, 2024

CS-470: Final Reflection

Experiences and Strengths

This course has been instrumental in advancing my journey toward becoming a proficient software developer, particularly in the realm of cloud computing. I've developed a solid understanding of AWS technologies, honing my skills in deploying applications consistently with tools like Docker and Docker Compose. Additionally, I've learned to navigate serverless architectures, particularly through practical applications of AWS Lambda and API Gateway, enabling me to build scalable applications efficiently without the constant need for server management. I've also expanded my knowledge in database management by transitioning from MongoDB to DynamoDB, enhancing my ability to manage NoSQL databases under various load conditions. My strengths lie in problem-solving, adapting across different programming environments, and ensuring seamless integration between the front end and back end of applications.

Types of Roles

Looking ahead, I'm equipped to engage in roles that leverage my cloud architecture and development skills. As a Full Stack Developer, I aim to optimize and integrate both the front-end and back-end aspects of applications, ensuring comprehensive performance and user experience enhancements. My ability to handle the entire stack allows me to troubleshoot, enhance, and innovate across all layers of development effectively.

As a Cloud Architect, my focus would be on designing scalable cloud solutions that align with organizational goals, ensuring efficient use of cloud resources while maximizing security and performance. This role would allow me to leverage my understanding of cloud services to architect frameworks that support robust, scalable, and cost-effective applications.

Planning for Growth

The insights gained from this course have solidified my foundational knowledge of cloud services, essential for my future projects. Managing scale and addressing errors in a serverless environment involves setting up automatic scaling and robust error handling mechanisms to maintain smooth operations. For cost prediction, leveraging real-time resource monitoring tools aids in effective budgeting. When comparing cost predictability, serverless architecture often proves more advantageous for handling variable workloads due to its pay-for-what-you-use model, whereas containers might be preferable for more stable, predictable workloads due to their consistent resource utilization.

Choosing between expanding with serverless or containers involves a careful evaluation of their respective advantages and challenges. Serverless technology is excellent for reducing operational overhead and enhancing scalability, ideal for businesses prioritizing flexibility and cost-efficiency. However, it can present issues such as potential vendor lock-in and latency issues during cold starts. Containers, while offering greater control and suitability for complex applications requiring specific environments, demand more comprehensive management.

Future planning will heavily rely on elasticity to ensure resources automatically adjust to the application's demands, optimizing both performance and cost. The pay-for-service model is critical in this respect, aligning expenditure with actual usage to minimize waste and support flexible financial planning. These factors are crucial in decision-making processes for planned future growth, ensuring that infrastructure not only meets current needs but is also primed for efficient and cost-effective scaling.