CS 470 Final Reflection

Aaron Bias

February 26, 2025

CS 470 Final Reflection

YouTube Link: https://youtu.be/9EFdjNlQSAE

Experiences and Strengths

Throughout my studies in Full Stack development at SNHU (specifically CS 470), I've honed my technical skills into a cloud-capable developer, gaining practical experience with AWS services that complement my problem-solving skills. This course has equipped me with containerization expertise using Docker, serverless implementation through Lambda and API Gateway, and database management with DynamoDB. My transition from traditional development to cloud architecture has positioned me for roles such as Cloud Developer, Solutions Architect, or DevOps Engineer, where I can leverage my unique blend of operational efficiency from retail management with technical cloud skills. My ability to communicate complex technical concepts to both technical and non-technical audiences, as demonstrated in my conference presentation, makes me particularly effective in collaborative environments where bridging understanding gaps is essential.

Planning for Growth

Looking toward future application growth, I would implement serverless architecture to enhance both scalability and efficiency. AWS Lambda functions would automatically scale to meet demand during traffic spikes while scaling to zero during quiet periods, eliminating wasted resources. For error handling, I would implement a comprehensive strategy using CloudWatch for monitoring, dead letter queues for failed operations, and custom error handling in Lambda functions with appropriate HTTP status codes. Cost prediction would be managed through AWS pricing calculators based on anticipated traffic patterns, with cost allocation tags to track spending by feature. Between containers and serverless, containers provide more predictable costs since you're paying for allocated resources, but serverless can be more cost-efficient for variable workloads despite less predictability. The deciding factors for expansion would include cold start latency concerns with serverless, execution time limits (15 minutes for Lambda), and potential vendor lock-in, balanced against zero infrastructure management overhead and automatic scaling.

Elasticity and pay-for-service principles would be central to my growth planning strategy. With AWS's elastic infrastructure, my application could automatically adapt to seasonal fluctuations, marketing campaigns, or gradual user base growth without manual intervention. This eliminates both the waste of over-provisioning and the customer dissatisfaction from under-provisioning. The pay-for-service model transforms infrastructure from a capital expense to an operational expense, reducing upfront investment while creating a direct relationship between costs and actual usage. This financial model would allow me to experiment with new features without major infrastructure commitments and attribute costs precisely to specific application components. For my application's planned growth, I would implement a hybrid approach, using serverless for variable workload components and containers for steady, predictable services, allowing me to optimize both performance and cost as the application scales. This balanced strategy would give me the flexibility to focus on delivering user value while the infrastructure automatically adjusts to match actual demand.