Jeremy Lyons
CS-470-H7084 Full Stack Development II 23EW2
8-1 Assignment: CS-470 Final Reflection
Southern New Hampshire University
December 17, 2023
Project Two Presentation: https://youtu.be/PR_MO6tladM

Experiences and Strengths

In CS-470, I learned and developed several essential skills that will enhance my marketability in the tech industry. I have gained hands-on experience with containerization and AWS services, including S3, Lambda, API Gateway, IAM, and DynamoDB. I have learned best practices for building and securing serverless applications. This course has helped me develop my strengths as a software developer, including adaptability, problem-solving, attention to detail, and technical and non-technical communication. These experiences have prepared me for roles such as Full-Stack Developer, Cloud Solutions Architect, or DevOps Engineer.

Planning for Growth

Microservices / Serverless for Efficiency

Microservices offer modular software development, while serverless architectures improve efficiency by eliminating server infrastructure management. Serverless automatically scales to meet demand, and the stateless nature of AWS functions makes them resilient to errors. Additionally, AWS Step Functions can be used to manage workflow and errors. Containers have more predictable costs, but unused resources can waste money. Serverless pay-for-use models only charge for usage, but costs may be less predictable. However, tools like AWS Pricing Calculator can help predict the cost of serverless solutions.

Pros and Cons for Expansion

Pros of using serverless include scalability, adaptability, ease of development, built-in security, and cost-effectiveness. Cons include increased system complexity, vendor lock-in, and limited cost predictability.

Elasticity and Pay-for-Service

Elasticity and pay-for-service play essential roles in making decisions for planned future growth. Elasticity allows applications to scale resources automatically based on demand. This ensures smooth performance during traffic spikes without needing to manage infrastructure. Pay-for-service offers a cost-effective model where payment is based on actual usage. This reduces the upfront investment necessary and eliminates wasting costs on unused resources.