Michael Medina

3/2/2025

CS 470 Final Reflection

https://youtu.be/2mOHb9NkI_8

Before this course all my computer science projects were developed locally on my machine. This course taught me how to leverage the resources of cloud computing to develop apps faster or to migrate existing apps to the cloud. I learned how to develop a full-stack web apps using the resources available in the Amazon Web Services ecosystem. I learned skills such as containerization, storing data in S3 buckets, using the EC2 environment, using the DynamoDB database, creating Lambda functions, setting IAM roles, and using the API gateway. All these skills will make me more attractive to future employees. At the moment, I feel like my strengths are problem solving and critical thinking. However, I need to improve my design and research skills. Some of the roles that I feel comfortable with are jr. web developer and web app tester.

Microservices or serverless functions seen in AWS automate the scaling of your app reducing developer intervention. Errors are also handled by microservices such as AWS CloudWatch and logging systems. Early detection of errors prevents future complex errors. Since serverless pricing is based on usage, you can predict cost by determining the number of executions, memory requirements, and execution time. Cost estimation tools such as AWS Pricing can help you forecast cost based on workload. Containers cost is more predictable for steady workload while serverless is more cost-effective for unpredictable workloads.

Some of the factors that may help determine future expansion would be scalability, overhead management, cost efficiency, performance, flexibility, and security. Serverless comes with automatic scaling, AWS full management, event driven pricing, delays may impact latency, limited flexibility, and security is managed by cloud provider. On the other hand, Containers need to be configured to auto-scale, require orchestration, better for constant workloads, consistent performance, more control over runtime and environment, and more control over security.

Elasticity in serverless computing is the ability to scale up or down when needed. The system will automatically detect traffic spikes and traffic downtimes and scale the app accordingly to reduce costs by limiting resources without affecting performance. Pay-for-

service allows businesses to scale in an efficient way without having to pay for idle time. This makes it a great choice for startups and growing applications.