Alis Gurung
04/20/2023
CS 470 Final Reflection
https://youtu.be/92WNPs0r03Q

Experiences and Strengths:

CS-470 course which revolves around the full stack web development was heavily focused on cloud development and cloud deployment. This course allowed me to learn more about cloud hosting of a full stack web application. It focused on the front-end , back-end and database which is a basic and essential component for a full stack web applications.
I have never developed a full stack web applications, and this course allowed me to dive more deep in to the AWS services like APIs, Lambdas and DynamoDB.
APIs and Databases were a new thing for me, I also believed my knowledge about Lambdas and S3 as well as IAM roles and policy strengthen my existing skill set. I feel more confident in AWS compared to the start of the course. I have been a back-end engineer and it was a great experience learning about node.js and angular development. Now, i can say this course allowed me to enhance my skillset and add front-end skills to my current skills. Being a software developer, we should be ready to analyze,develop, deploy and maintain any applications we are entitled to. The strength as a software developer should be provide a full fledge architecture about the applications and eagerness to maintain it against any kind of vulnerabilities.

Planning Growth

The cloud provides computing resources over the internet, including servers, storage, databases, networking, and software. Users pay for the resources they consume on a pay-as-you-go basis when these services are offered by third-parties.
In the cloud, microservices and serverless are popular methods for scaling and optimizing web applications. Serverless enables developers to write code without worrying about infrastructure and scales automatically, while microservices break complex applications into smaller, independent components.
In a microservices or serverless architecture, developers can use tools like load balancers, auto-scaling groups, and monitoring systems. In load balancers, traffic is distributed across multiple instances of a service. In auto-scaling groups, instances are added or removed based on traffic patterns. System monitoring alerts developers to errors and can help diagnose problems.
Developing serverless applications and microservices can be expensive. However, cloud providers typically provide pricing calculators that allow users to estimate costs based on usage patterns. Serverless applications are often more cost-predictable than container-based applications, as users only pay for the exact amount of resources used, rather than paying for a fixed amount of capacity that may be underutilized.
When deciding between microservices and serverless, there are several pros and cons to consider. Microservices provide greater control over infrastructure and can be more easily

ported across different cloud providers, but require more upfront investment in development and management. Serverless, on the other hand, allows developers to focus on code and abstracts away much of the infrastructure management, but can be more difficult to debug and optimize. Elasticity and pay-for-service are important factors to consider when planning for future growth in a cloud-based application. Elasticity allows the application to scale up or down based on demand, providing cost-efficiency and ensuring that the application can handle traffic spikes. Pay-for-service pricing models allow users to pay only for the resources they use, rather than investing in fixed infrastructure, making it easier to scale the application without incurring unnecessary costs.