Eli Rubin-Calvert

CS470

3/3/2024

CS 470 Final Reflection

https://www.youtube.com/watch?v=zu0-CXNDi9E

This course can help me in reaching my professional goals in a variety of ways. Learning more about fullstack development is definitely useful for me, since I have had jobs as a fullstack developer in the past, and plan to have more of them in the future. I especially liked that we learned about Aws services such as lambda functions and S3, I know from past experiences how useful lambda functions are. But getting an opportunity to practice them more was very helpful. Learning more about DevOps and cloud infrastructure management can be very beneficial to my career as a software developer. I am prepared to assume roles that relate to programming and cloud management such as a backend developer, a full stack developer, and a cloud developer. Knowing more about the cloud ecosystem can also help me in a role in the future such as CTO or Software Architect.

Microservices or serverless may be used to produce efficiencies of management and scale in my web application in the future in a variety of ways. Microservices can allow for more modular development, easier deployment, and greater scalability. Microservices allow for horizontal scaling, and can make development of a large codebase easier. Serverless architecture abstracts away infrastructure management, allowing developers to focus on writing code without worrying about provisioning or managing servers. Since developer time can be expensive, there can be a lot of cost savings by implementing these things.

In a microservices architecture, scale can be managed through horizontal scaling, deploying additional instances of individual services to handle increased demand, error handling can be implemented at the service level, ensuring that failures are isolated and do not affect the entire application. serverless architectures abstract away infrastructure management and automatically handle scaling based on demand, with error handling mechanisms built in.

When planning for expansion of a web application, several factors need to be considered. Microservices provide flexibility and modularity, allowing for independent scaling of individual services based on demand. Managing a large number of microservices can lead to increased complexity in deployment and communication between services so you need to think about the needs of the application. Serverless architectures offer more predictable costs based on actual usage, while microservices may require more careful monitoring and optimization to control costs effectively.

Elasticity refers to the ability of the infrastructure to dynamically scale resources up or down based on demand. This is important for handling changes in user traffic and ensuring optimal performance without using too many resources needlessly. pay-for-service models, such as those commonly found in serverless architectures, tie costs directly to resource consumption. Instead of paying for fixed infrastructure regardless of usage, organizations only pay for the resources they actually use. This can make cost estimation easier and may be a better fit for some applications.