Bryan Molina

10/25/2024

CS 470 Final Reflection

https://www.youtube.com/watch?v=N6X2xDO81Iw

The course has been a significant step toward achieving my career goals. Working in a call center, I've gained skills in problem solving and customer support, but I've been eager to expand into development, with cybersecurity as my goal. The technical concepts covered in this course, especially containerization, cloud based application scaling, and serverless architecture, have been invaluable in moving me closer to these goals.

One of the skills I've developed here is containerization with Docker and Docker Compose, allowing me to build applications from scratch. This gave me a clear understanding of creating, managing, and deploying services efficiently. I also gained insight into AWS services, reinforcing the basics I'd encountered before but expanding into more hands-on work with scalable cloud tools.

My strength as an aspiring developer is my adaptability and eagerness to learn. Moving from a non technical role, I'm continually pushing to develop new skills and refine what I know. Completing this course has increased my confidence, showing me that I can handle the complexities of backend development and cloud infrastructure, key areas that align with both development and security focused careers. This prepares me for roles such as junior developer or cloud security analyst, where foundational development knowledge is essential.

Planning for Growth

When planning for the future scalability of a cloud based web application, I would apply the serverless and microservices approach. By using AWS Lambda and similar serverless tools, I could focus on code rather than infrastructure, allowing my application to adjust to demand automatically without extra management. Additionally, serverless would help isolate issues so that errors in one service don't bring down the entire application.

To anticipate future costs, I'd evaluate serverless based on usage data, knowing it scales with demand and offers flexibility. However, containers are often easier to predict for cost since they have a fixed runtime environment. I'd consider the stability of containers for applications needing constant availability, while using serverless for lighter or more scalable processes.

Serverless microservices bring benefits like cost effective scaling and elasticity, as you only pay for what you use, but they may not be as effective for complex applications with high dependencies. Elasticity and the pay for service structure are essential to my growth planning as they offer the adaptability needed for expansion without excessive cost increases. By leveraging these, I can ensure the application remains scalable, cost effective, and resilient as demand grows.