Achna Hettiarachchi

February 26, 2024

CS 470 Final Reflection

<https://youtu.be/hf1fEyILlR4>

**Experiences and Strengths**

In CS 470, I was able to gain a diverse set of skills that are helpful in software development in today’s world. First, I learned about containerization using Docker and recognized the importance of that in smooth deployment and scalability. The next important concept that I came across was serverless architecture. In particular, I studied how to use AWS Lambda, API Gateway, and S3 for efficient and cost-effective application hosting. Finally, I did assignments that helped me understand cloud-based development principles via database management (MongoDB, DynamoDB).

As a Computer Science graduate planning to enter the competitive job market learning how to develop and deploy full-stack web applications in cloud environments was very useful. Containerization and orchestration are among the strengths that I gained via this course. Now I am capable of serverless architecture, optimizing resource usage and cost efficiency. Along with all the above, I improved my problem-solving and debugging skills, in the development and testing of web applications.

With some more practice, I will be well-prepared for roles involving full-stack development in cloud-based environments. With a comprehensive understanding of containerization and orchestration, I am ready for responsibilities in containerization, serverless architecture, and cloud service management. I can also contribute to contribute to projects focusing on scalability, efficiency, and optimal resource utilization.

**Planning for Growth**

Efficiency in our cloud-based web application centers around smart choices like microservices and serverless architecture. When it comes to handling growth, these architectures play a crucial role by dividing tasks into manageable parts. What's nice about serverless is its built-in error-handling features. They quietly ensure our application stays robust and reliable even in the face of unexpected issues. Predicting costs in the cloud involves analysis of usage patterns and a consideration of pay-for-service models. However, serverless architectures bring a welcome clarity. Their pay-as-you-go model, based on actual usage, gives us predictability and transparency. It's like paying only for what you use, a concept that resonates with anyone managing a budget. In a nutshell, choosing microservices and serverless isn't just about scale and resilience. It's also a smart financial move, providing efficiency and cost predictability for our cloud-based web app.

Scalability occurs as a robust asset, naturally accommodating the growth of our application as per the user’s demands. Cost efficiency is a helpful factor too. Deployment is an efficient process, while resource optimization becomes second nature. However, the path to expansion has some complications. Introducing more dependencies must be done with careful consideration. Additionally, we must expect a learning curve for team members to master the concepts.

In orchestration, we encounter two key ideas: elasticity and pay-for-service. Elasticity is the ability to adapt our application to varying workloads, ensuring consistent and optimal performance. On the financial front, pay-for-service takes the lead. This aligns costs precisely with actual resource usage, promoting cost-effectiveness. It's like paying only for the resources consumed, a prudent financial model that ensures transparency and efficiency.