

Takiyah Edwards

June 28, 2024

CS-470 Full Stack Development II

CS 470 Final Reflection

https://youtu.be/gvGabc7E_Jc

Experiences and Strengths

This course has been influential in helping me reach my professional goals as a software engineer. By exploring the intricacies of cloud-native applications and AWS microservices, I have gained a deeper understanding of modern development practices that are fundamental in today's tech landscape. The skills I have learned, such as containerization, orchestration, serverless architecture, and cloud integration, have made me a more marketable candidate in the competitive field of software development.

I have developed proficiency in using Docker and Docker Compose for containerization, learned to deploy serverless applications with AWS Lambda, and mastered the integration of frontend and backend services using API Gateway. These skills ensure that I can efficiently manage and scale web applications in a cloud environment. My strengths as a software developer include problem-solving and adaptability. In my current role as a Software Engineer, I have developed the skills that will allow me to transition to a Full Stack Developer and apply my knowledge to build scalable, efficient, and secure applications.

Planning for Growth

Through this course, I have gained extensive knowledge about cloud services, particularly in the context of AWS. In the very near future, I plan to use microservices and serverless architectures to enhance the efficiency and scalability of web applications as my company is gearing towards migrating our applications to the cloud. Microservices will allow me to develop and deploy independent components, facilitating better management and faster updates. Serverless solutions like AWS Lambda will enable automatic scaling and reduce the need for infrastructure management, making my applications more resilient and cost-effective.

To handle scale and error handling, I would implement auto-scaling policies and use AWS CloudWatch for monitoring and logging. This will ensure that my application can dynamically adjust to varying loads and quickly recover from errors. Predicting costs will involve analyzing usage patterns and leveraging AWS's cost management tools. Generally, serverless architectures tend to be more cost-predictable compared to containers, as you only pay for the actual execution time rather than pre-allocated resources.

When planning for expansion, it is important to consider the pros and cons of both containers and serverless solutions. Containers offer greater control over the environment and are suitable for long-running processes, but they require more management effort. Serverless, on the other hand, provides ease of deployment and automatic scaling but may have limitations in execution duration and cold start latency. Elasticity and the pay-for-service model play a crucial role in decision-making for future growth, as they ensure that my application remains cost-efficient and can handle varying workloads without manual intervention.