CS 470 Final Reflection

Brandon Rhodes

Southern New Hampshire University

CS-470 Full Stack Development II

George Johnson

August 25, 2024

Presentation: https://youtu.be/uitmveYhAOI

Experiences and Strengths

Throughout CS 470, I developed a range of crucial skills that are essential for a career in cloud development. I gained a deep understanding of containerization, particularly using Docker, which allows me to package applications and their dependencies into isolated containers. This skill ensures consistency across different environments, from development to production, and is vital for modern software development. Additionally, I learned how to leverage serverless computing, especially through AWS Lambda, which has enabled me to focus more on writing efficient application logic without worrying about server management. This approach not only simplifies the development process but also ensures that my applications are highly scalable and available.

Another significant skill I acquired is the ability to manage multi-container applications using orchestration tools like Docker Compose. This has reduced the complexity of cloud deployments and improved operational efficiency by allowing me to define and manage multiple services in a single configuration file. I've also developed a strong emphasis on security, learning to implement key measures like Identity and Access Management (IAM) and encryption. This focus on security is something I will continue to carry with me in every project, ensuring that my applications are both functional and protected against threats.

As a software developer, I've found that my strengths lie in being adaptable. I am now comfortable with various cloud migration strategies, whether it's Rehosting, Replatforming, or Refactoring, which means I can tackle a wide range of scenarios in cloud application deployment. My experience with serverless computing and containerization has helped me better understand the cloud environment which will allow me to streamline the development process, allowing me to deliver high-quality, scalable applications quickly.

These experiences have prepared me for several roles in the tech industry. I feel confident stepping into positions like Cloud Developer, where continuing my study of cloud computing will allow me to apply my knowledge of containerization and serverless computing to build scalable and efficient applications. This has also opened possibilities to work as a DevOps Engineer, managing CI/CD pipelines with ease, thanks to my experience with tools like Docker and Docker Compose. Coupled with the knowledge of cloud security, my possibilities have expanded significantly, even broadening my understanding of the role I am in today as a sys admin as we are fully integrated and hosted on the cloud through AWS.

Planning for Growth

Looking to the future, I plan to use my understanding of cloud services to help my web applications grow. For handling scale, I'd rely on AWS Lambda's automatic scaling, which adjusts resources based on demand, making sure the application can handle traffic spikes without a hitch. AWS Lambda also offers great tools for error handling, helping me identify and fix issues quickly. When it comes to cost, serverless computing offers a more predictable pricing model since you only pay for what you use, which is key for budgeting and planning for growth.

As I think about expanding applications, I see both the advantages and challenges of using microservices and serverless architectures. On the positive side, they offer excellent scalability and cost efficiency, especially for workloads that fluctuate. They also cut down on the management overhead by removing the need to manage servers. However, they can add complexity, particularly when it comes to managing communication between services and debugging in a distributed system. There's also the risk of becoming too dependent on a single cloud provider, which could make it difficult to switch if needed.

In planning for the future, elasticity and the pay-for-service model are crucial. Elasticity allows the application to scale resources up or down as needed, which helps avoid unnecessary costs and ensures the app performs well under varying workloads. The pay-for-service model is great for managing costs because it ties expenses directly to actual usage. By keeping these factors in mind, I can ensure that my web applications are not only scalable but also cost-effective, ready to grow and adapt as needed.