Pravishna Nand

CS 470

8/25/24

# CS 470 Final Reflection

Link to my presentation: https://youtu.be/H7T-HD7IY3w

## Experience and Strengths

The CS 470 Full Stack Development II course at Southern New Hampshire University has been pivotal in advancing my skills in cloud-based application development, directly contributing to my professional aspirations in software engineering and cloud computing. By focusing on the practical application of containerization and AWS cloud services, I have been able to build on the foundational knowledge gained in Full Stack Development I, leading to a deeper understanding of how to deploy and manage applications in a cloud environment. This course has equipped me with critical skills such as setting up Docker containers, leveraging Docker Compose for orchestration, and utilizing AWS services like Lambda and API Gateway. These are highly sought-after skills in the current job market, making me a more competitive candidate for roles that require proficiency in cloud technologies. Moreover, the course emphasized the importance of articulating complex technical concepts to different audiences, a skill that is essential for roles such as DevOps Engineer, Cloud Architect, and Full Stack Developer, where clear communication is crucial for successful project execution and collaboration.

Throughout the course, I have developed a strong proficiency in containerization, cloud infrastructure management, and the implementation of microservices. These strengths position me well to take on roles that involve designing and maintaining scalable, cloud-native applications. My ability to work with AWS cloud services and to integrate them seamlessly with full-stack applications is a significant asset, particularly in environments that prioritize efficiency and scalability. Furthermore, the hands-on experience with cloud-based frameworks has reinforced my problem-solving abilities, particularly in handling the challenges associated with distributed systems and ensuring their reliability. I am now prepared to step into roles that require not only technical expertise but also the ability to adapt to the rapidly evolving landscape of cloud computing. Whether it’s in a DevOps role, where continuous integration and deployment are key, or in a software development role focused on cloud-native applications, I am confident that the skills and knowledge gained from this course will be instrumental in my professional growth.

## **Planning for Growth**

The knowledge gathered from the CS 470 course on cloud services has provided me with a comprehensive understanding of how to leverage cloud technologies for the efficient management and scaling of web applications. Specifically, the course introduced me to the concepts of microservices and serverless computing, both of which offer significant advantages in terms of modularity and scalability. Microservices, by allowing an application to be broken down into smaller, independent services, make it easier to scale specific components based on demand, while serverless architectures, such as AWS Lambda, provide a cost-effective solution by charging only for the compute time used. In planning for future growth, these architectures will be critical in ensuring that applications can scale efficiently without incurring unnecessary costs. Additionally, understanding how to handle scale and error handling within these frameworks is essential, as it ensures that applications remain resilient and performant under varying loads.

When it comes to predicting costs and planning for expansion, the choice between containers and serverless architectures is a key consideration. While serverless models offer more predictable costs at smaller scales, containers might provide better cost control and resource management as the application grows, especially in scenarios where long-running processes or complex dependencies are involved. The elasticity of cloud services, which allows for automatic scaling of resources based on real-time demand, plays a significant role in this decision-making process, as does the pay-for-service model that ensures costs are aligned with actual usage. However, the trade-offs between operational overhead, control, and flexibility must be carefully evaluated. Containers offer more customization and control, which might be necessary for complex applications, but they also require more management. On the other hand, serverless solutions reduce operational overhead but might lack the flexibility needed for certain workloads. In planning for future growth, I will need to balance these factors to ensure that my applications can scale effectively while maintaining cost efficiency and operational simplicity.