# CS 470 Final Reflection

Katrina S. Gilliland

Global Campus, Southern New Hampshire University

CS-470-13979-M01 Full Stack Development II

Professor Mohammad Alam

March 2, 2025

# CS 470 Final Reflection

**Link to Unlisted YouTube Video with Presentation:**

https://www.youtube.com/watch?v=jSr83I4rsMg

**Experiences and Strengths: Explain how this course will help you in reaching your professional goals.**

- **What skills have you learned, developed, or mastered in this course to help you become a more marketable candidate in your career field?**

Southern New Hampshire University's Software Development II course provided me with valuable hands-on cloud development experience. Specifically in this class, I was given the opportunity to learn about the Docker tool and how it can be used to containerize an application, as well as the various services provided by Amazon Web Services (AWS) for transitioning an application to a serverless architecture. With cloud-based environments becoming increasingly more popular, the ability to leverage both Docker and AWS services to create and support serverless architectures will surely make me an extremely marketable candidate within the tech industry.

- **Describe your strengths as a software developer.**

One of my greatest strengths as a software developer is my ability to communicate technical concepts to non-technical audiences. In this course, I was able to demonstrate this strength by creating a PowerPoint presentation on the intricacies of cloud development. Within this presentation, I walked my audience through the process of migrating a full stack application to an AWS serverless solution and explained the various benefits of cloud computing such as

scalability, cost efficiency, and operational efficiency. Breaking these complex concepts down into simple yet powerful takeaways emphasizes my ability to effectively communicate with others; a skill I feel will make me valuable to any team.

- **Identify the types of roles you are prepared to assume in a new job.**

Thanks to my time at Southern New Hampshire University and my experiences gained from the Full Stack Development I and II courses, within a new job, I now feel prepared to take on roles that deal with full stack design and development, as well as roles dealing with implementing and managing cloud-based environments.

**Planning for Growth: Synthesize the knowledge you have gathered about cloud services.**

- **Identify various ways that microservices or serverless may be used to produce efficiencies of management and scale in your web application in the future.**

Microservices and serverless architectures are beneficial for web applications due to the improved scalability, faster development and deployment cycles, and cost efficiency they offer, among many other major advantages. In terms of scalability, microservices break an application down into smaller services and scale those services up or down independently based on individual demand, and serverless architecture automatically scale resources based on incoming requests, eliminating the need for manual intervention. Regarding promoting faster development and deployment cycles, microservices allow developers the ability to develop, test, and deploy services independently, and serverless architectures remove the need for infrastructure management, leaving more time for developers to focus on the application itself and delivering value to their users. Lastly, when it comes to cost efficiency, microservices optimize resource usage by only allocating the necessary resources to each

service, and serverless architectures only require customers to pay for actual active compute time.

- **How would you handle scale and error handling?**

In future applications, when it comes to scaling and handling errors, I'd specifically utilize AWS Lambda. AWS Lambda scales applications in real-time in response to incoming requests, ensuring that it can handle various levels of traffic without requiring manual intervention. In addition to its automatic scaling capabilities, through AWS Lambda, I can leverage features such as failure retries as well as error logging via its direct integrations with other Amazon services such as AWS CloudWatch (AWS, n.d.).

- **How would you predict the cost?**

AWS helps predict the cost of cloud services, such as AWS Lambda, by providing customers with the AWS pricing calculator. The AWS pricing calculator allows users the ability to estimate the cost of using their AWS services based on their unique cloud infrastructure needs. Through the AWS calculator, users can visualize their potential cost based on their expected and planned usage.

- **What is more cost predictable, containers or serverless?**

Containers can be seen as more cost predictable than serverless architectures as customers are required to carefully plan out and provision their resources ahead of time. With containers, there is a fixed cost based on the number of resources allocated, therefore if a customer knows exactly how many resources they will need, they can plan and budget accordingly for their application. Containers' cost predictability makes them better suited for applications with predictable workloads while serverless architectures, which only require customers to pay for

their actual resource usage rather than a fixed number of resources (active or idle), are considered more attractive for applications with varying, unpredictable workloads.

- **Explain several pros and cons that would be deciding factors in plans for expansion.**

When planning for application expansion, deciding between containers and a serverless architecture would depend on several factors. For example, if a customer has a predictable workload and is looking for more control over their application, they may opt to use containers for expansion over a serverless architecture. Containers allow for more control over applications with consistent, predictable demand by giving the customer the ability to precisely allocate the number of resources needed as well as the ability to manually add or reduce containers based on their expected traffic patterns. In contrast, if a customer has an application with varying traffic patterns or unexpected workloads, they would likely opt to use a serverless architecture for expansion. Through using a serverless architecture, resources are automatically scaled up or down based on real-time demand, allowing the customer to only pay for what they need.

- **What roles do elasticity and pay-for-service play in decision making for planned future growth?**

Elasticity, a system's ability to scale resources based on demand, and the pay-for-service/pay-as-you-go pricing model both play a major role in planning application growth. As mentioned previously, if a customer knows the exact number of resources needed or is confident in their estimation of how many resources will be needed for their application in the future, utilizing containers may be more valuable due to their upfront costs and customizable nature. On the

other hand, if the customer is unsure of their application's level of traffic or how many resources will be needed (such as a start up or new application), utilizing a serverless architecture with a pay-as-you-go pricing model would be more beneficial. Ultimately, choosing between the two would depend on the application itself.

## References

AWS. (Unknown). Using cloudwatch logs with lambda. *Docs.aws.amazon.com*, 1. https://docs.aws.amazon.com/lambda/latest/dg/monitoring-cloudwatchlogs.html