

Title: 8-1 Assignment: Final Reflection

Name: Kursheeka Milburn

Instructor: Mohammad Alam

Course: CS-470-13374-M01 Full Stack Development II

Date: August 24, 2025

Table of Contents

Experiences and Strengths	3
Planning for Growth	4
Clear Communication	5
Conclusion	6
References	7

Experiences and Strengths

Completing CS 470 has given me real-world experience with cloud technologies and full stack development. One of the most important skills I gained was learning how to use Docker and Docker Compose to containerize applications. This allowed me to package software and its dependencies into containers that run consistently on any machine. I also learned to use Docker Compose to manage multi-container applications, which made it easier to connect services like the frontend, backend, and database together (Merkel, 2014).

Another major part of this course was deploying applications on AWS. I created an S3 bucket to host a static Angular frontend and learned the benefits of cloud storage compared to local storage. Unlike local storage, which has limited disk space and hardware risks, S3 provides unlimited, durable, and redundant storage that is secure and scalable (Amazon Web Services, 2023). I also created serverless backends with AWS Lambda and connected them to API Gateway so that users could interact with my application.

I practiced building and using NoSQL databases through DynamoDB, which gave me an understanding of managed databases that scale automatically. Comparing DynamoDB with MongoDB showed me how cloud-managed databases reduce the need for manual scaling and maintenance.

From these projects, I discovered strengths in problem-solving, debugging, and connecting cloud services together. I also became more confident in explaining complex topics in a simple way, which I showed in my presentation script. This course has prepared me for roles such as full stack developer, DevOps engineer, or cloud developer, and it supports my long-term goal of becoming a cloud solutions architect.

Planning for Growth

Planning for growth in a web application is one of the most important lessons from this course. If my application had to scale in the future, I would design it using microservices and serverless computing. Breaking the application into smaller services makes it easier to update and scale only the parts that get heavy traffic. Using AWS Lambda also means my code would scale automatically without me managing servers. AWS provides built-in logging, retries, and monitoring tools like CloudWatch to handle errors and keep applications reliable (Garcia Lopez et al., 2018).

When thinking about cost, serverless computing is usually more predictable for applications with changing workloads. Since you only pay for requests and runtime, there is no charge when the system is idle. Containers can be more predictable for applications with steady traffic, but they can also waste resources if they are always running during low demand (Roberts & Chapin, 2019).

Here are some points that guide my planning:

- **Serverless pros:** automatic scaling, no server management, pay only for what you use.
- **Serverless cons:** possible cold start delays, limited by provider environment.
- **Container pros:** portability, consistent runtime across machines, stable for long-running tasks.
- **Container cons:** require more management, costs continue even when not used heavily.

The concepts of elasticity and pay-for-service are key to making smart decisions. Elasticity lets applications automatically expand during peak traffic and shrink during slow times. The pay-

for-service model ensures I only pay for what I use, which prevents waste and supports cost-effective growth (Mell & Grance, 2011).

Clear Communication

This course also gave me practice with communication. By building my PowerPoint and script, I had to explain cloud concepts clearly to both technical and nontechnical audiences. Learning to organize information and present it in a professional way is an important skill, and I believe I improved at it throughout this course. Being able to explain what I built is just as important as building it.

Conclusion

CS 470 gave me valuable hands-on experience with containers, serverless functions, cloud deployment, and managed databases. I learned how to plan for growth by using cloud-native ideas like elasticity and pay-for-service, and I developed stronger skills in explaining technical concepts. These lessons will help me in future jobs and move me closer to my career goals. Overall, this course has prepared me to build scalable, cost-effective, and secure applications in the cloud.

References

- Amazon Web Services. (2023). *Amazon S3: Object storage built to retrieve any amount of data from anywhere*. <https://aws.amazon.com/s3/>
- Garcia Lopez, P., Cabrera, J., Gómez Sáez, S., & others. (2018). Serverless computing: Programming, applications, and challenges. *Future Generation Computer Systems*, 83, 50-63. <https://doi.org/10.1016/j.future.2018.01.022>
- Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing*. National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-145>
- Merkel, D. (2014). Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal*, 2014(239), 2.
- Roberts, M., & Chapin, J. (2019). *Serverless architectures on AWS*. O'Reilly Media