Aeriel Denmark
Sunday, April 27th, 2025
CS 470 Final Reflection
Presentation Video Link: https://youtu.be/oRAYg_X03wU

Experiences and Strengths

CS 470 helped me bring everything together. I built and deployed a full stack application, worked with cloud services, and shared my work through a professional-level presentation. This was one of the most useful and complete projects I have done during my degree. It made everything I learned feel real.

I gained hands-on experience with Docker, Docker Compose, AWS Lambda, API Gateway, DynamoDB, and S3. I learned how to write and connect backend services, organize resources, and secure everything using IAM roles and policies. These are skills I can use in real jobs that involve cloud development or backend engineering.

As a developer, I am good at staying organized, following through, and solving problems in a practical way. I know how to look at the big picture while also paying attention to the details. I am ready for entry-level roles in backend development, DevOps, or cloud support, where I can build on what I have learned and continue to grow.

Planning for Growth

In this course, I learned that moving an application to the cloud is not just about where it runs. It is about how you manage it, how you plan for growth, and how you keep it efficient. I would use serverless functions and microservices to keep the system flexible. That way, I could update or scale parts of the app without affecting everything else.

To plan for growth, I would monitor performance with AWS tools like CloudWatch. I would set up alerts and track how different parts of the system behave under stress. For error handling, I would add logging, retries, and fallback methods to make sure the app can recover from small issues before they become big ones.

When it comes to cost, I learned that serverless is usually more predictable. It works well for apps with changing traffic. Containers are better for services that need to run all the time, but they can cost more. I would use the AWS pricing calculator to estimate the cost of each choice and make a plan based on usage.

There are pros and cons to both containers and serverless. Serverless is easy to manage and keeps costs low, but it has some limits. Containers give more control but take more effort to maintain. My decision would depend on what the service needs to do and how often it runs.

Elasticity and pay-for-use help with planning. Elasticity lets the app grow when needed without overpaying when things are quiet. Pay-for-use means I only get charged for what I use. Together, they help make the app scalable and affordable.

This course helped me understand how to plan ahead. I do not just think about what works now. I also think about how it will work later, how much it will cost, and how I can improve it over time.