Malik Robinson

August 19th, 2025

CS-470 Final Reflection

https://youtu.be/S2s1B_I6DRs

This being a capstone course for full stack development, I developed skills in Docker, Docker Compose and using AWS. These skills will reflect in my field of focus being front-end development. These skills will help in that field because the more skills I have the more reliable and productive I can be as a developer. I can perform tasks outside of my field reliably. Since I have been developing my skills in front-end development, I believe my strengths as a developer are organization, problem solving, and my design skills. Along with those strengths my ability to cooperate and work in teams while communicating efficiently are more of my strengths. With the knowledge I gained from this course the roles I am prepared to assume are full stack developer and software developer.

Microservices or serverless support the handling of efficiency across the application stack. Lambdas controls the ability to manage and handle errors that can occur in the process. AWS's auto scaling makes it, so storage problems are non-existent. When it comes to predicting the cost, the base storage need is an important indicator for the cost because AWS has a pay-per-use model. Along with predicting the storage that is needed AWS has a tool named AWS pricing calculator that allows for accurate estimates on the cost based on use cases. Auto scaling can change the price drastically depending on the changes that are made when it comes to the amount of storage or services used. In my opinion when it comes to cost predictability serverless is more predictable because the storage or size of the service is known beforehand and if no major changes are made then the cost can be accurately predicted.

It is always important to properly plan and determine the pros and cons that can come from planning an expansion of your service. One con is the budget restrictions that can occur from the expansion. Another con is the amount of time that can come with building the infrastructure along with properly planning out how it will be integrated into the existing system. A pro that can come along with the expansion is the possibility for more traffic and revenue that can come along with it. Another advantage is the ability to restructure preexisting services or code to make for easier expansions in the future. Elasticity and pay-for-service play major roles in the planning of future growth because they dictate the types and number of changes that can and will be made to the preexisting system. They work in tandem with each other because elasticity removes the storage restrictions that maybe present in a plan and pay-for-service adapts to the amount that is used only making it so there is no need to worry about purchasing too much storage since it will adapt to the amount you need and will only cost the amount you use. These two concepts take away two major worrying points when it comes to planning an expansion and gives the developers the room to focus on the substance of the design.