Daniel Anderson

Full Stack Development 2

04/26/2024

https://youtu.be/ufdvh_ldE7I

CS 470 Final Reflection

This course really helped me get the full picture on how full stack works with the serverless system. Learning how to use the AWS system was a great way to learn how different applications work and how they interact with each other individually and as a complete unit. Learning these skills makes me a more well-rounded developer that can use many different types of applications to help develop programs for different companies. I feel that my biggest strength as a developer thus far is my debugging skills and to track down where errors are coming from and solve them correctly and quickly. I do not have a main career goal in mind but developing new software is where I would like to land with a focus more on medical technology and helping to create devices to help patients have a more stable life that suffer from varying medical conditions.

Using a serverless cloud-based application allows you to scale your application as needed without wasting resources on things you might not use. This scalability allows you to control the number of errors that can occur because you can slowly add different applications and easily control the effects that they have on each other. Using cloud-based allows you to better control and predict the cost of maintaining and running the application because you only pay for what you use and there is no surprises or unknown costs when expanding and growing your application to handle more resources. This type of cost tracking is what allows serverless to be more cost-effective over containerization and remains the best way to pay only for the resources that you use. Pay as you go models is the best way to track costs and to keep them low, but you do become limited on the number of things you can do with your application when it starts getting big. Looking at buying all your resources up front is very costly but depending on the amount of growth your application will use might be better if your application will become very large and end up consuming a lot of resources. Elasticity and pay-for-service models are great at keeping cost low and maintenance to a minimum. These models are great if you do not except a lot of growth and are playing on maintain a very basic and small application. Once your application is planned to become massive these models become harder to use therefore making them ineffective for the business needs.