Kao Chao

12/12/2022

CS 470 Final Reflection

I progressed as a developer in a multitude of ways courtesy of the material in the Full Stack Development II course. The abilities I developed using serverless architecture concepts and containerization tools such as Docker to convert programs are essential. It was quite an experience learning about Node.js and how we constructed the front end and back end of a website that interacted with the server. This was all very new to me and interesting at the same time. However, this course provided a great introduction to paths for us to choose what we are interested in. We have just scratched the surface and there is still so much to learn. But because we were building upon each module and seeing how each AWS service interacts with one another to build our project, I also had the ability to enhance my skills as a software developer I had never done anything like this before this course, so it was amazing to see how it developed week by week, giving me more confidence for what's to come. I gained problem-solving abilities from this course, which will undoubtedly be useful for me in the future, no matter what company I work for.

I enjoy learning, and since technology is continuously evolving, this is a strength I have as a software developer. There is always something new to learn. Finding the topic that interests you the most and staying with it are better options than trying to study everything at once, which would be disastrous because there is so much information to recall in just one path that it would take up most of your time. As a new software developer, the role I’m prepared for is to be a team player who can think critically and solve problems.

Microservices and serverless tools from AWS provide methods for handling effectiveness throughout an application's stack. Step Functions and Lambdas both have the capacity to properly manage mistakes. AWS Auto Scaling, which enables automatic capacity modifications depending on the consumption of EC2 and DynamoDB instances, can also be used to handle scalability challenges. AWS offers a practical cost calculator as well. Cost estimates can be made using a tool called the AWS Pricing Calculator which is based on specific use cases. Utilizing the pricing calculator and serverless architecture allows for some cost certainty. Compared to containerization, serverless systems typically offer higher cost predictability, however, it ultimately relies on the use case.

The largest drawback to the growth might be the cost of doing so. Additionally, it might be challenging to overcome the difficulties of maintaining a network or managing workspaces, recruiting, and training new hires, and the time required to build out infrastructure. The positives of expansion can result in higher revenue and a company that is better situated for achieving success in the future. The cost of the infrastructure is reduced thanks to this elasticity. This is not applicable to all environments; rather, it is useful to address just those situations where the resource requirements abruptly increase and decrease for a set period. The business concept of "pay-for-service," where you only pay for the services you really use. As a result, you just utilize the resources that you require at any given time and pay for what you utilize.