James Fowles

04/28/2024

CS 470 Final Reflection

Presentation Link: https://youtu.be/_z45CPO2FO4

Experiences and Strengths

This course has been one of the most informative classes I have taken because it involved hands-on practice with technology that is not only current but arguably the future of web development. In this course, I have learned what "Serverless" computing really is and how to use API gateways and Lambda, and why they are important. Knowing this information gives any developer an advantage and a better understanding of the technology they may one day be asked to develop.

I have been a full-time software developer for many years, yet this course has still been a big benefit to me, as I primarily work in the .NET framework and, as such, had not had much exposure to AWS or serverless computing. I think knowing about these technologies will increase my strengths and give me some new ideas and ways of looking at application design. And as such I think with a little more practice, this course is a good steppingstone to an entry-level position in cloud computing/development.

Planning for Growth

When planning for growth, there are some things to consider such as how to handle scale and error handling. For handling scale, I think microservices, which means breaking down large applications into smaller independent services, would be a good solution as these are perfect for the AWS Lambda model and each could be connected to run together and can scale up and down on demand.

Error handling is also made easier in this model as each individual service can be turned off independently, making it easier to isolate the issue versus the traditional method of stepping through a whole program looking for the bug.

Predicting cost is tricky using microservices because of the auto-scaling nature; however, AWS does have an impressive suite of monitoring tools that allow you to see the status and usage of your services such as when they are being used and what are heavy use timeframes, allowing you to do some calculations on total cost.

When it comes to what is more predictable, containers or serverless, I think serverless are more unpredictable due to the nature of them only costing money when they are in use. Meaning, during a slow month, you could have very little cost, and then the next month, you could have

big costs. But personally, I believe the benefits outweigh the negatives as it's better to pay only when you are using the services than paying for being idle.

When I'm thinking about the pros and cons, deciding factors in plans for the expansion of cloud services, some of the pros are scalability, flexibility, and resilience. Each of these pros are hands-off and will be handled by the provider, allowing more focus on the code/product. The cons are complexity and cost. To use these services, you need to be prepared to pay different amounts each month depending on your growth, and you need to consider the complexity of managing all the different services that are required such as S3, API Gateways, and Lambda.