Henry Herisse

10/20/2022

CS 470 Final Reflection

https://youtu.be/j6Wx_izSib0

Before taking this class, my knowledge of both ends of application development was not great. In this class, I was able to gain some great knowledge of how to develop an application from the front end to the back end. With the coursework, I had the opportunity to use the tools like Docker and Amazon Web Services also known as AWS. Throughout the course, I had the opportunity to learn about containerization and how to convert applications into serverless architecture in AWS. One important tool I was able to manage was my time. This class requested a lot from me, but it was rewarding in the end. I think the skillset I gained in this course will help me in the future with some great opportunities such as cloud software engineer positions. Another important thing I learned is the fact that you only pay for what you use. It is a straightforward service in terms of you paying for what you want instead of paying a set price and receiving part of the tools you would to convert your application to a serverless platform.

In microservices, scale is one of the advantages. For instance, if a service requires additional capacity, it can just spin up a new instance to handle the load instead of modifying something else to accommodate the extra load. It is important to not ignore errors when errors are ignored time is wasted that could be spent recovering from them. I think one of the best ways to handle errors is to structure the program to make recovery easy.

The cost of cloud services varies. Some services require monthly subscriptions while others are free. The yearly cost of running a service is based on your own specific needs. Cloud functions are free with an unlimited number of functions therefore you can test them before you start paying for anything.

Both containers and serverless are more cost predictable than virtual servers. With virtual servers, they can be set up to utilize more resources than they need. Containers are probably more cost-predictable because once the container runs, it has all the code it needs to run. Thus, leaving less room for mistakes and less over usage of being charged for resources that you are using. Cost models are predictable; however, it all comes down to the organization processes of your application.

When you decide on what to do for an expansion plan, it is vital to remember that your strategy will affect how you manage your business. For instance, if you are selling a product or service that can be utilized on a variety of devices, this will give you an advantage over other organizations. Nonetheless, because not all users will be using the same type of device, this means that their features will be different. This could impact accessibility and security because you could lose customers who need access to these features without either of these factors working properly. To build and maintain your application, you need to consider how well your strategy will work for you and what it will cost you. The pros of expansion would be higher revenue and better odds of sustainable success with growth. The obvious con to any expansion, on the other hand, would be a capital expenditure which is followed by the time it would take to build the whole infrastructure.

Lastly, elasticity permits the resource to be moved up or down as needed. It allows you to use what you need. This is when the pay-of-service method comes in into play because you only pay for what you use. With AWS you can calculate exactly what you need to use and then charge you accordingly. The combination of both allows any organization the necessary tools to achieve sustainable growth at a fair cost.