Jason O’Connell

4/18/2023

CS 470 Final Reflection

https://youtu.be/OJCXZx9F7vU

Experiences and Strengths

I had a small bit of experience in cloud-hosting before coming to this course. I had built an application that needed to be hosted so I checked out Google Cloud Services. This course has taught me more of the intricacies of how cloud-hosting works and when it’s a good idea. Specifically, we migrated a full stack application to a cloud-hosting environment. These skills are important because self-hosting is inefficient in a lot of cases which has led businesses to migrate to cloud-hosting. Having knowledge in how to develop cloud deployment is quite valuable.

In a new job, I would be confident in scheming a deployment plan for either a new stack or an existing one for cloud deployment.

Planning for Growth

Scale is easier in cloud-hosting because much of the physical scaling is done automatically by the service provider. With any application, the bigger the scale the harder it is to manage. Fortunately, cloud service providers offer feature rich services that allow you to monitor and report errors with your business logic.

From a cost perspective, you go through the motions of what a typical user would do with the application. Then, multiply that by how many users you expect to have using the service. Containers are more expensive because they require more computational resources than bare serverless. Provisioning hardware yourself is difficult to forecast because every expense, foreseeable or not, is left to you.

Serverless environments have become very attractive options over self-hosting. But self-hosting still has its place. Self-hosting is particularly useful in air-gapped systems where your closely guarded servers with highly sensitive information require a hardline connection. You wouldn’t want to host that in a serverless environment because anything that is on the web has an attack surface and can therefore be compromised. But if you don’t need that much control and have no interest in running your own servers, serverless is a great option. Serverless is elastic which means you do not have to worry about adding more servers if your app takes off. Instead, you only pay for what you use. This model can also help you better cost forecast as opposed to self-hosted.