

Christian Reynolds

02/26/2023

Assignment: CS 470 Final Reflection

Presentation Link: <https://www.youtube.com/watch?v=YP5s42bcl7U>

### **Experience and Strengths**

In this course, I worked on bringing my code to production. Previous courses were primarily focused on design and development, while I feel this course was focused on deployment. Being able to migrate a full stack web application to cloud services is a valuable skill that undoubtedly will serve me in many different computer science disciplines.

My strengths as a software developer include the ability to design and implement a solution from the ground up. This course, however, has given me the ability to design and implement solutions in a serverless setting, and also migrate existing solutions to a serverless setting.

I think this makes me a perfect fit for a full stack developer position. I have now been exposed to many different facets of web development and feel I am able to quickly learn and adapt to the rapidly emerging technologies in this particular field. That's not to say that these skills cannot be applied to other areas of development.

### **Planning for Growth**

In a serverless setting, scaling and error handling are simple. For error handling, I can simply connect my specific services to a logging service. In the case of AWS, this is CloudWatch. If I can get logs of errors, then they will be that much easier to pinpoint and fix. Scaling, on the other hand, is done by the serverless platform itself. I just pay for the extra use and or traffic on a particular service. The cost can be predicted by looking at the serverless platforms cost breakdown and then predicting how much my service will be used. Most platforms charge by use and data throughput. Containers work in a similar manner, but serverless cost is more predictable

than container cost. Containers still rely on the host machines resources and quite a bit more thought must be directed to server provisioning. Serverless eliminates this worry.

Elasticity and pay-for-service are very important in planning future growth. I never want to pay for more than what I need, so being able to pay more when I need more, and pay less when I need less is necessary for keeping costs down. If I only pay for what I use, I can focus more on creating a well running application that scales when I need it to and not before or after.