Sawyer Prestwood
Oct 24, 2021
CS 470 Final Reflection
https://youtu.be/sUMkJhnGFL0

Final Reflection

This course has brought to my attention the various ways developers are both creating for cloud-based environments and migrating older content to cloud-based systems. The tutorials throughout the AWS experience were very helpful and, while I don't claim to be a master at developing for this particular system, I do believe I have a fundamental understanding of how to migrate code to a cloud-hosted system for deployment. The content of this course has also taught me the structure and general flow of information for cloud applications, which can be applicable to most any other cloud application deployment solution (despite the proprietary aspects of AWS). My strengths as a developer lay mostly unit testing and building code from scratch, which while a worthwhile skill to have, isn't exactly the most common thing seen in the industry. This and other courses are exposing me to the workflow of taking existing code and modifying it to meet certain requirements. Roles I am prepared to assume in any given new environment would be an entry-level developer role, mostly likely testing existing code or writing smaller functions to expand functionality rather than working on full integration of systems.

Microservices and the serverless cloud architecture is very useful for scaling and segmenting an application. Aspects such as the API Gateway, function calls, and data storage all being separate allows developers to place a more strict focus on security through the use of polices; the built-in "sandboxing" of application components also helps prevent unwanted traversal through the application from bad actors, as each segment would require (ideally) a different form of authentication. Scaling an application in this environment would be as simple as paying for more storage options, importing existing code to meet the needs of the data requirements, and applying policies and API routes to manage server traffic. The ease of use for scalability can also be seen as a pitfall of serverless computing; despite the “pay what you use” model, developers still need to anticipate the amount of storage required for their application, and the fact that it is so easy to spend to receive more storage - while considerably less expensive than traditional server-based application deployment - can still become an issue cost-wise if the application is not making use of all the resources payed for. There’s also the learning curve involved with transitioning an application to the cloud. However, the overall cost-effectiveness of a cloud-based application would more than pay off for any of the pitfalls of the transition.