Danica Hesemann

10/19/2021

Project Two Presentation Link: https://www.youtube.com/watch?v=cycoyituaSg

CS 470 Final Reflection

In this course, I have learned how to migrate an application to the cloud using two different methods. In doing so, I have gained experience using Docker and AWS services such as S3, Lambda, API Gateway, and DynamoDB. I would say my strengths as a software developer and some things that have helped me in this course include an ability to think through problems using different approaches, an ability to learn quickly, and knowing when to look for additional resources or help. These strengths mean I can solve challenges presented by an application's requirements, I can adapt to different tools and techniques relatively quickly, and I understand my own limitations and how to avoid letting them hinder me. I would say that at this point, I would be prepared to handle a development role, or perhaps an operations role. I have some experience with several development tools and frameworks, I am familiar with several programming languages, and I understand the basics of full stack development.

Microservices and serverless are both great options that can increase productivity and decrease costs. Cloud providers like AWS may handle most of the management and maintenance for a serverless application, making management and security much simpler for developers. Various services are offered and integrate easily with each other. The services offered are subject to vender lock-in and may not always work with preferred languages or provide the flexibility a developer may want for security, maintenance, and management, but serverless is simple and efficient when reasonable to use. Serverless options also handle scaling automatically (elasticity) and typically use a pay-for-use model that means developers will only pay for what is used. This can make cost less predictable, but is typically a cost-effective option for event-triggered applications and makes it a lot easier to scale the application for growth as needed.

Microservices or containers are another excellent option because it is possible to build, deploy, and test one component independent of the others. That makes it easier and more efficient to test, scale, and manage microservices, often using less code and whatever frameworks or languages are suited to the specific component. Containers allow for more flexibility and customization than serverless services, but management and maintenance are more involved. The cost will depend on the application and what services and tools are used, some applications may be developed using Docker for free, others will require more resources at a greater cost. Containerization is a probably a better option for longer running or more complex applications.