Joshua Morris
CS470 Final Reflection - April 2024

Link to accompanying presentation below:
https://youtu.be/1FTaplSZho8

Through this course experience, I have had the opportunity to explore modern cloud-based development. My work with Docker and Docker Compose has helped me to develop an understanding of containerization and cloud migration, both of which are staples of modern web-development. Building on work done in previous courses, I have learned how to execute the complete process of creating a full stack web application, moving that application to containers, and preparing the application for upload to the cloud.

I have also learned the basic principles necessary for migrating an application to a serverless environment. This was accomplished using the AWS service, one of the largest cloud providers in the world. Through this course, I created and ran a full-stack application in the AWS environment, including an Angular front-end built in S3 bucket, a full back-end using Lambda scripts and API Gateway, and a database using DynamoDB. All of these tools are commonly used for modern web services, so developing these skills will be crucial for my intended career in information management.

Serverless and microservice platforms are becoming increasingly popular, as they allow companies to cut costs and maximize efficiency. As I explain in my presentation, the traditional practice of maintaining large banks of servers to support an online application can be problematic, especially when the application is not receiving enough traffic (and therefore creating enough revenue) to justify the expense. Having dedicated servers allows for greater control, but I would suggest that there are several factors to consider before deciding between a traditional, dedicated server, and a serverless model.

First, determine whether the application requires direct management by the client. The primary disadvantage of serverless and microservice environments is that they limit the tools available for application management. While this does not mean that these services are not incredibly versatile, there will always be situations in which the host company wishes to support certain applications or functions that are simply not possible in an externally managed environment. For these applications, having company-managed servers or dedicated cloud space is the better option.

The second consideration is the cost and predictability. Dedicated servers and purchased cloud space are generally predictable in terms of cost, while serverless platforms' cost will fluctuate depending on usage. This can be a benefit, because the cost will drop during periods of low usage. However, it could also be a danger, if an unexpected spike in usage is not accompanied by a corresponding spike in revenue.

Finally, I would consider the need to plan for future scalability. Serverless models have a huge advantage here, because they scale automatically to account for usage. This 'always available' model is one of the primary selling points for the serverless environment, because it makes scaling an almost seamless process. Traditional cloud-based storage or physical servers will require purchasing additional resources, which will also entail work to bring those resources online. This does not mean that these types of environments are unusable, but again, it is something to consider.