Brennan Riley
Cemel Tepe
CS-470
October 27th, 2024

Final Reflection

https://youtu.be/E4Gd13bc9kg

Over the course of this term, I have learned a lot about the process of building functional, efficient, and secure web applications in a cloud-based environment. I have greatly improved both my understanding and ability to write clean and effective code in a number of languages such as Java, HTML, YAML. In addition to improving my technical skills I have improved my ability to communicate the technical aspects of my work to different audiences.

I believe that my strength as a developer lies in my ability to identify potential areas of inefficiency and create effective solutions to address them. I also enjoy creating clean and readable code that can easily be understood, even by non-technical readers using ample comments and documentation.

After completing this course, I feel more prepared to take on several different roles in the IT industry. I am better prepared for roles within cloud infrastructure or development teams that manage the network architecture of an organization. I also feel confident in discussing topics within other disciplines such as cyber security, identity and access management, and data analytics. I have thoroughly enjoyed the multidisciplinary nature of the coursework and feel that it did a great job of providing a great general understanding of cloud-based development principles.

One of the core principles of cloud computing that we covered and discussed this term was scalability both at the service level and as it relates to a serverless architecture.

Microservices provided through AWS allow developers to quickly scale individual components of a larger system such as storage or EC2 instances to meet increased demand. Within a serverless architecture scalability is achieved through the generation of individual instances whenever an event is triggered at the web API level. When it comes to error handling microservices and serverless platforms typically include their own error monitoring services such as AWS CloudWatch.

Costs for microservices are calculated based on the computer time and capacity requirements of containers. This makes predictions easier for components that have consistent usage that rarely fluctuates such as long-term static databases but can be more difficult for those that experience rapid changes in demand. Serverless architectures however accrue cost based on the number of individual function calls and their execution times. Predicting costs for this model is the opposite of microservices as they do well with fluctuating traffic but struggle at scale and under long-term static long-term use.

The potential benefits of microservices are numerous but the three key aspects are control over the system, consistency, and ease of configuration and scaling. The potential downsides to this approach is that management and orchestration of containers can become complex at scale and there is still some need to manage underlying architectures. Serverless architectures also have several benefits such as the ability to automate scaling without the need to manage infrastructure. It also allows for rolling deployments and updates as functions can be accessed, altered, or deleted individually without the need for downtime. This approach however suffers from limitation on runtime and might require the redevelopment of applications to work in a serverless architecture.

Elasticity and pay-for-use are also two key selling points of a cloud-based development approach. Elasticity refers to a cloud-based system's ability to more accurately meet the needs of its users through scalability to fit demand. Pay-for-use means that those who take this approach only pay for the cloud resources they use whether that be an entire container or the individual functions of a serverless application.