Thomas R. Hummel

2023-10-19

CS-470 Final Reflection

YouTube video link of final presentation: https://youtu.be/6uL-IVadRug

Experiences and Strengths

During the move of a MEAN stack application to a cloud-based application I was able to learn many things. At the most basic level I was able to learn about the specific workings of many parts of the AWS family of tools. More importantly though, I was able to learn many of the concepts and patterns behind cloud-based software development which can carry over to most vendors' implementations of cloud hosting. For example, I am now more familiar with the layering of APIs with other functionality and how pieces of code should be broken out into proper containers.

As a software developer I think that my biggest strengths are my detail-oriented nature which comes from years of database development and my ability to understand complex problems and come up with viable solutions.

Although I have been working as a database developer for many years my work in this course and others at SNHU leaves me feeling much more confident in my ability to take on jobs in other areas of software development, such as front-end coding, back-end coding, and full stack development.

Planning for Growth

One way that microservices in a cloud deployment can produce efficiencies is the ability to log and track individual services. It's much easier to identify bottlenecks either in CPU, memory, or network latencies. The elasticity of cloud hosted microservices then makes it much easier to ramp up or down the resources for that specific microservice. Or, if appropriate, a microservice can be rewritten or rearchitected to be more efficient.

Detailed logging can also be used for scaling and planning future growth of the system as well as much more efficient error handling. Rather than getting an error from a monolithic application and having to track the error down to a specific piece of code, errors can quickly be traced to their specific microservice. It’s always possible that the underlying error comes from another service, but this will still be a much more efficient approach in general.

Most cloud hosts allow for configurations that make it much easier to control costs, whether that’s for a container-based solution or a simple serverless hosting solution. However, if you are going under the assumption that you never want to let your system go offline due to high usage it seems to me that a serverless system will be more prone to unexpected resource usage under high traffic since the resources must be shared by the entire system. So, if one part of the system experiences high resource usage across many connections it can overwhelm other parts of the system.

Knowing that the system can be easily expanded to match future growth thanks to elasticity is certainly a big upside for serverless and container solutions. It’s possible that costs could unexpectedly run out of control with a pay-for-service host, so setting upper limits might be necessary for a system. High usage of a system is generally a good thing for a business though as long as their business model is built to scale.