**CS470 Final Reflection**

Samuel Hanson

12/10/2023

Video Link: <https://youtu.be/R9ie1MxMd1A>

* What skills have you learned, developed, or mastered in this course to help you become a more marketable candidate in your career field?

While I had some experience with fullstack development beforehand, this course provided me with a more in depth look on many of the tools I had already used. While I knew how to use Lambda, I wasn't as familiar with API Gateway and the importance it has in Lambda's security.

* Describe your strengths as a software developer.

My main strengths lie in understanding, creating and improving backend code. I enjoy the process of working with algorithms and finding ways to make them as efficient as possible.

* Identify the types of roles you are prepared to assume in a new job.

What's helpful about fullstack development is that I now have at least a basic knowledge of each role within software development. While my skills would lean more towards writing code on the backend, I also have the knowledge to work with frontend code and databases.

* Identify various ways that microservices or serverless may be used to produce efficiencies of management and scale in your web application in the future.

Because microservices and serverless have a pay-for-use price model, companies will only pay for what's going to be used. Traditional architectures require costs to be done upfront for a set amount of resources, but this method will almost never be the exact amount needed by the company, meaning the company paid for wasted resources or not enough resources were given and the software cannot work effectively.

* Explain several pros and cons that would be deciding factors in plans for expansion.

As companies expand, they must allocate more resources to their services. If a company uses a traditional software architecture, purchasing more upfront resources is risky, as the company may not make that money back. This isn't a problem for serverless architectures, though. Serverless may not be a great choice for companies needing more control of their backend, though. A traditional server architecture gives developers full control over their backend, but serverless cloud providers limit the control of developers. Serverless also may not be the right choice for companies needing to use long running functions, as they'll be charged for the runtime.

* What roles do elasticity and pay-for-service play in decision making for planned future growth?

As previously mentioned, pay-for-service architectures are very cost efficient, as companies don't need to worry about resources going unused. Elastic models are similar, because companies don't need to worry if they purchased too many resources or don't have enough, as elastic models change the amount of resources used to meet demand. If a company cannot predict how much traffic they'll receive, a pay-for-service and elastic model will be preferable.