

CS 470 Final Reflection

This course has taught us, from a high level, the different service offerings available on the cloud. I've gained awareness of technologies I didn't know existed. We've also gained hands-on experience using and configuring several AWS products. This is important to my professional growth because all of these tools are used daily and widely throughout the software industry. These elements have added up to what I believe is one of the most valuable courses I've taken at SNHU.

I don't think I'm the fastest learner out there. But what I do have is a structured approach to problem-solving that I think a lot of new graduates are not taught. I only know it because I didn't learn it at university. It's especially helpful for solving coding problems, but it can sometimes apply to architecture design too. What I also have that many new graduates don't have is experiencing learning on my own. I mean without any structure or program curriculum at all. It takes an entirely different mindset. Passion gets you started, but passion fades. It takes discipline to force yourself to keep going even when you don't feel like it. With this experience I'm ready to enter the workforce as a software engineer.

Handling scale with a traditional monolithic architecture has inherent challenges. Software tends to become more and more complex as it grows in its feature set and capabilities. Even if the code is well-written, it takes a human being time and energy to read through and comprehend all of it. Then if a change needs to be made, it's difficult to find a fast way to do that without breaking the app in one or more places.

A microservice architecture allows for decoupling of services, making them easier to change when desired. Scaling becomes easier because adding new code doesn't break the application from top to bottom.

Predicting cost for any software service can be difficult because the amount of user utilization is unpredictable. That's one of the big value-adds for cloud services – they adapt to the needs of your users. Capital is spent efficiently. On-prem hardware doesn't grow or shrink quickly. It needs to be built, configured, and maintained by staff who can only work at human speed. If you decide to spend a fixed amount on hardware and labor, you can predict your cost for your on-prem application. It matters little at that point whether you store your app in containers or VMs if you have a fixed budget. Serverless cost isn't always predictable, as your app may experience high traffic volume spontaneously which incurs large costs. However, you will at least have confidence that your app will be able to serve all users.

It's important to remember that though serverless and cloud have powerful benefits, they are not always the "best" choice. No infrastructure is 100% safe from security exploits. Even Amazon. When you host on-prem, you know every intimate detail about the security of your hardware, platforms, software, and data. You have complete control. You can make any change you want at any time. Any downtime for your app is your problem and not the fault of someone else.

YOUTUBE PRESENTATION:

<https://youtu.be/wFHex3PlyzA>