

Jason Ditullio

Professor Cemal Tepe

CS-470

4/20/2024

### Final Reflection

Narrated Presentation Video Link: <https://youtu.be/MEudTrqE3Q4>

After taking this course, one skill that I have learned is how to create containerized applications from full stack applications using Docker Compose. This is very useful for the future as many applications are run in containers and now, I know how to set them up. Another skill that I have learned is how to transfer container applications from local to cloud based hosting. Specifically in this course I learned how to use Amazon Web Services which is used by many applications to host and manage their data in serverless environments. Lastly, I learned how to secure the data I upload to AWS through creating roles and policies in order to make sure that only authorized users and resources are accessing my data. I believe that one of my main strengths as a software developer is that I am constantly testing my code throughout the development process. Rather than write up the entirety of my applications and then deal with bugs or errors, I split my code into manageable chunks that I test overtime with each added section of code. This makes it easier for me to catch any errors early on in development as well as preventing me from being confused as to what may cause the problems. When looking at what roles I am prepared to assume after my degree, I am prepared to run any entry level software developer position as well as any entry level web development role from my experiences in both backend and frontend development.

Looking ahead to the future of my web application developed within this course, the benefit of having it hosted on a serverless environment is that I don't have to worry about the scale of the application over time. This is because through AWS S3 bucket, the storage I need can change over time and I am only charged with what storage space I am using. For error handling, the API gateway would allow me to view any errors or access logs to my application and allows me to simply update any problematic code without needing to setup the server again. Going back to the serverless environment, since the application can be scalable based on how much data I do use, I wouldn't need to predict the cost in the long run as AWS would only charge me for what I do use. This also makes me believe that serverless is more cost predictable than something like containers because you can put in as much data into storage as you need and are only billed for what you use. For container, it is still on local storage so you would have to eyeball how much storage you would need overall and risk either overspending or spending too little on hard drive storage. A benefit that would come from expanding to a serverless environment such as AWS is that you wouldn't need to worry about managing and setting up an entire server for your application as it is managed by your cloud provider. Also using a serverless environment is beneficial to your applications efficiency as the application can be run from any

server that is closest to the users. One con to expanding to serverless environments such as AWS is that not all of your applications dependencies or libraries may be supported by your cloud provider. Another downside is that you may be restricted on what data types or files are allowed for your database based on the chosen cloud provider as well.

To conclude this reflection, elasticity and a pay-for-service model play an important role in deciding on future growth for applications. Elasticity allows applications resources to decrease or increase at any time based on application usage. This is beneficial to developers as you aren't being forced to pick and choose a certain amount of storage or number of resources you may need for your application as it can change over time. A pay-for-service model means that for developers, they are only charged based on how much their application is used rather than a set monthly cost. This is important because if your application at some point has a lull in its use, you are charged only on its use rather than for all the time it could have spent idle. The time I spent studying at SNHU for my career has been enlightening and helpful. I am ready to see what is in store for any future career paths now that I have moved on to graduating.