

Brian Russell

June 22, 2021

CS-470 8-1 Assignment: Final Reflection

[CS-470 Project Two Presentation](#)

### **Experiences and Strengths**

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

Throughout the course of Full Stack Development II, there are a few things that I learned which helped me to understand more of the software development world. During the course, we learned about static websites and how to transfer them to the cloud, specifically, Amazon's Web Services. For me, this was new and exciting. I have never tried to use AWS nor any other cloud-based services. I have always done things via programming, so learning about it was an eye-opening experience. What I learned is that there are many ways that a database server, website server, or even APIs can have access to the outside world. Using AWS, one can give access through APIs to a database by using HTTP requests and responses, coupled with roles, policies, and permissions. By learning AWS and how the communication is handled, it gives me a better candidacy for a career choice.

Describe your strengths as a software developer.

One of my major strengths is the ability to adapt, not only to the environment, but also to the required languages needed within the career field. I have been fascinated by handles, queries, loops, game mechanics, and even Artificial Intelligence programming which is why I believe I learn each language quicker than most. Usually within one month, I am fluent in it and have a grasp on how the language is handled.

Another great strength is my attention to detail. When I program, I make sure to read over my work during the process phase, then again after I am finished. One of the highlights is I am very thorough in my work. I make sure to look at my work as a user's perspective when I write it, and make sure its readable and flows neatly.

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

As a person that has been in a leadership role all my life, I see myself being a leader, whether paid or not for it. Currently at my work I am considered a "heavy hitter", meaning I am good at my job. People tend to come to me for questions because I pride myself in learning anything and everything in this world. I take that same mentality into my job. Many people tend to come to me for help and answers because I try to learn everything I can about my work.

### **Planning for Growth**

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

Some of the benefits of microservices and/or serverless applications is the ability to use APIs as a middleman solution to client and server applications. For example, a website and its pages were transferred to a cloud-based solution. Then, the database is transferred as well. For the database and website to talk to each other without exposing security holes, APIs are introduced. APIs handle the to and from data links using HTTP request and responses. And because its serverless, the amount of to and from data can be scaled to fit the needs of the business by allocating more resources in a dynamic setting.

How would you handle scale and error handling?

To correctly scale in a serverless environment, first look at the requirements that your application needs. Does it need hard drive space, or require a lot of resources? Then, assess the initial cloud-based environment that would correctly support the application needs. Once the applications are transferred to the cloud, scaling becomes easier. The required resources can be allocated using multiple systems within the cloud. Also, transferring to another location is easy as well. In a serverless environment, migration is used to scale.

For error handling in AWS, they have AWS Step Functions that is used in conjunction with Lambda. AWS Step Function can create a workflow that handles errors. It can use conditional logic to pinpoint where the error is coming from. What this does is allows you to create an error handler API that will be used as a link with a role and permissions pointed to this handler. Not only can you catch any coded errors, but a catch all can be used as well.

How would you predict the cost?

Like mentioned before, to predict the cost, one would have to evaluate the current resources that are used for the current application(s). Then, any dependencies that are needed which could use resources as well. Once the total is found, the estimate cost would be based on the current market choices and their services. This method would provide a rough estimate as to the requirements needed.

What is more cost predictable, containers, or serverless?

Depending on the application, serverless can be more cost predictable over containers. In a container, with each iteration of the application, comes with packaging it for distribution. This unpredictable size for future iterations becomes more costly. Also, a container must be maintained as well which adds to its cost. For a serverless environment, you only pay for what you use, meaning if the application is running but no traffic is happening, then there are no resources being used. So, your cost would decrease. The same applies to scaling, the more resources used, the more the cost goes up.

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

For a pro factor, if the application was to be a static type, website and scalability is needed, then serverless would be the “go to” choice. But a con of this is the latency that follows. Since the website is now at a different location while being online, the time to travel would change and customers or users would see what is called lag. Another pro factor is the ability to have your application up and running quicker using cloud-based than it would be with a container. With a container, one would have to build or have a space built, then install all necessary software before the application can be used. The third pro to con is the cost. The cost to build, maintain and upgrade a container is more costly than having it serverless.

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

Elasticity in a server environment means it can adapt to the workload and its changes by adjusting resources as needed. Pay-for-service means you pay for the resources used. When combining both for planned growth, the decision stems to how fast the growth will be. Some questions could be considered like, how much growth? Do we move to a different network to accommodate the traffic, or is the current bandwidth location enough? In a serverless environment, scalability is quicker because the network provider has better elasticity.