Ryan Stork

CS470: Module 8 Final Reflection

Southern New Hampshire University

10/21/2021

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?

Through the course of this project, we took a static web application, and then we containerized it so that it could communicate with the backend database. Then we took the pieces of the application and we brought it to the AWS cloud. Moving through each week brought us closer to bringing a cloud application online. We put the website on S3, then we created the DynamoDB tables. The third part was to create the lambda functions that would serve as our API calls from the website to the database. Finally, we used the API gateway to bridge the Lambda functions for a successfully working cloud application. The skills that I've learned was the relative ease of putting an application on the cloud. The ones that I have mastered the most were creating the Lambda functions, as we did that more than any other part. I have learned how to use DynamoDB as it relates to a cloud-based application.

Describe your strengths as a software developer.

As a software developer, I believe one of my strengths is my willingness to learn. I enjoy learning new things with any programming language. I think it is interesting to be able to take written words and transform it to a working program that other people can see and use. And any time I learn something new, it makes me a more successful developer. A second strength that I have is perseverance. When I have a programming issue, I will ask for help, but I will not give up. Sometimes, I try to break a large issue into smaller, more manageable issues. I could be eating dinner and a solution comes to my head. And I will rush to see if the solution works as intended.

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

For me, I think the role that I am most prepared for is a backend developer, most likely in C++. It is the language that I have the most practice. I don't have the most practice with developing the front end of the application other than through text.

Planning for Growth:

The cloud, AWS in particular, tries to be efficient in the services that they provide and a reasonable cost. **How would you handle scale and error handling?**

When it comes to the cloud, scale is not an issue. If your application becomes popular overnight, AWS can scale automatically for you. You do want to be careful, though. You don't want a bot pinging your application over and over causing rapid expansion. This may cost you a lot of money. Error handling would be easier with AWS Lambda. We can test the functions to make sure that they connect properly and the result that it will bring. If there is an error, we can correct it then, before full deployment.

How would you predict the cost?

The storage with S3 is easier to calculate. You know how large your project is and you can make backups of previous versions to also stay with S3. And then multiplied by the number of regions that you want your product on. API costs are depending on runtime. The shorter that the function runs, the smaller the cost. You can set a maximum time for timeout. You can estimate the number of times the API is getting called by the timeout time to get a representative cost for that section. DynamoDB is dependent on the size of the database and the number of read and write requests. Depending on your application, you can gauge the size of each document in the database and calculate the number of times the database is accessed.

What is more cost predictable, containers or serverless?

Typically, containers are more cost predictable. You would build servers on site and manage them. The cost is all up front, the price is known. If you have a local storage that has been running, then you can take that data and calculate it to see the estimated cost of going to the cloud. In the case of a new application, there is a little bit of a guessing game when it comes to a cloud cost.

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

The number of times the database is accessed can be a large factor in determining if someone would expand to the cloud. If you have read and write requests happening all the time, then the cost could quickly escalate, while if you had the database locally, there is no cost for accessing the database locally.

Another deciding factor would be the amount of money that your company wants to spend on the project before it is up and running. If your project remains small, it may be advantageous to stay on the cloud to save money.

A third deciding factor would be downtime. If your application is critical, then you might choose the cloud because it is up over 99.9% of the time. If there is an issue with the server, a copy is brought up to make sure that your application is always running.

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

Elasticity allows you to scale with peace of mind. You can look at diagnostic logs to see how much your application is being used and if you need more processing power. The cost is nearly the same for every tier. If we need more processing power, then we can estimate the pay-for-service price and plan accordingly. But we wouldn’t need to buy more local servers and maintain them, as the servers would be run by Amazon, off site.