Rosario Robinson
October 13, 2022
CS 470 Final Reflection

YouTube Link: https://youtu.be/uZEqs81NlIg

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?

This course provided a lot of great and informative information to help me become a more marketable candidate in my field. I have certainly heard of Amazon Web Services before, especially being a computer science major, but I have not had the opportunity until this course to understand and implement an application within AWS. Learning AWS is a skill set that should be further honed to become proficient at it; however, what I have learned during this class, whether it is creating containers using Docker for an Angular application, to understanding how to implement DynamoDB within AWS, has provided me with a great skillset to ensure I understand the fundamentals of building an application with AWS.

Describe your strengths as a software developer.

My strengths as a software developer, I believe, lie in my desire to learn. I am always curious to expand my knowledge and current skillset to continue improving upon myself. This means that I enjoy learning and understanding new concepts, especially when it comes to aspects of growing as a software developer. Taking this course, I was not expecting to be learning AWS or Docker and having the opportunity to learn about each within a school course was exciting. Another strength is adapting to new environments and learning concepts. I believe that SNHU prepares students well in this regard, since we are working at a more accelerated rate, and are introduced to new concepts constantly. This has allowed me to adapt to new environments and aspects of technology more than ever.

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

The types of roles I am prepared to assume in a new job will vary, although I have been focusing on becoming a software engineer. However, by taking this course and previous courses that focus on networking, databases, and applications, I do find systems operations engineering an interesting career as well. Having more experience understanding and implementing applications using cloud services like AWS provided me insight into roles like SysOps engineers and how they implement on the cloud side.

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.

Since we have focused on understanding and using Docker and AWS throughout this course, I will focus on how these applications/services provide microservices or serverless are used. Microservices are essentially a specific type of architectural and organizational approach to software development that breaks down the services, such as users, threads, and posts such as APIs (What Are Microservices? | AWS, n.d.). It essentially allows for faster scaling, especially considering when applications can grow in size and become more complex for developers to maintain.

To handle scale and error handling, in one instance, with AWS, Lambda assists in handling this for you. For example, Lambda has integrated with services such as Amazon CloudWatch and AWS X-Ray to ensure you as a developer have multiple means to log and identify errors within your application, as well as provide solutions to solving these errors (Error Handling and Automatic Retries in AWS Lambda - AWS Lambda, n.d.). AWS has extensive documentation that provides solutions for multiple different programming languages so that you can code your project as needed.

Predicting the cost and cost efficiency would depend on the scale of the application, and what services you are using. Overall, however, serverless applications are considered more scalable and cost-effective (Serverless Computing Vs. Containers | How to Choose, n.d.). This is why in most cases, for large-scaled projects, utilizing services like AWS is the better option, especially since it is a cloud-based platform.

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

There are pros and cons to expanding any application or service. One pro that comes to mind is scalability. Scalability refers to how the backend will inherently and automatically scale to meet demand (Serverless Computing Vs. Containers | How to Choose, n.d.). So, while expanding, this will allow a more seamless transition for developers and easier setup moving forward. Another pro that comes into mind is maintenance, especially from a serverless perspective. Since developers, while expanding to services like AWS, will not have to worry regarding maintenance or updates required, AWS would take care of that for them. Cons, however, would focus most on cost, which is what I previously touched on. Although AWS is serverless and more cost-efficient, depending on the level of scalability and expansion needed for an application, it can be very expensive for a company. There is also the issue of security, which depending on the service developers decide to expand with, there is always a risk of data leaks, no matter the serverless provider chosen (Serverless Computing: Uses, Advantages, and

Disadvantages | Okta, n.d.). However, many still rely on services like AWS since they are one of the best at providing great features.

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

Elasticity, or the ability to acquire resources when needed and release them when no longer required, is a huge feature in many services, especially within AWS (Elasticity - AWS Well-Architected Framework, 2020). This type of feature is hugely necessary when considering growth and expansion, especially for developers. This provides developers with the opportunity to scale out when needed for applications, and when they find that certain features are either no longer in use or no longer needed, they can scale in. AWS provides so many seamless and integrated features that this takes a lot of weight concern from developers and those that hope to scale their businesses or applications further in the future.

References

What are Microservices? | AWS. (n.d.). Amazon Web Services, Inc. Retrieved October 10, 2022, from https://aws.amazon.com/microservices/

Error handling and automatic retries in AWS Lambda - AWS Lambda. (n.d.). Retrieved October 10, 2022, from https://docs.aws.amazon.com/lambda/latest/dg/invocation-retries.html

Serverless computing vs. containers | How to choose. (n.d.). Cloudfare. Retrieved October 10, 2022, from https://www.cloudflare.com/learning/serverless/serverless-vs-containers/

Serverless Computing: Uses, Advantages, and Disadvantages | Okta. (n.d.). Okta, Inc. Retrieved October 10, 2022, from https://www.okta.com/identity-101/serverless-computing/

Elasticity - AWS Well-Architected Framework. (2020, July 2). Retrieved October 10, 2022, from https://wa.aws.amazon.com/wellarchitected/2020-07-02T19-33-23/wat.concept.elasticity.en.html