8-1 Assignment: Final Reflection

Kyle Dale

CS-470-Full Stack Development II

Professor Ogoh

https://youtu.be/XrzVrHMjbS4
4/28/2024

Experiences and Strengths: Explain how this course will help you in reaching your professional goals.

Throughout the course of this term, I honed and refined the concepts worked on in Full Stack Development I, and applied them to a full cloud-native environment. Rather than building a full stack application utilizing something like the MEAN or MERN stack, in this course, we packaged a pre-built Angular application into containers using Docker, and managed the multi-container application utilizing Docker Compose, before migrating the application to a cloud-native environment. This gave me invaluable experience in containerizing applications for deployment across a multitude of different environments, a much more efficient deployment than the more traditional methods of deployment learned in Stack Dev I. Then, through migrating the application, I gained further invaluable experience in developing in a cloud-native atmosphere, migrating applications to cloud native, and spent much time analyzing the comparative strengths and weaknesses of a cloud-native solution as compared to more traditional methods. All of this has positioned me to be capable of marketing myself to a wider sect of development, and provided me with the capability to expand my own development knowledge and ability.

I believe my strengths as a developer mirror the strengths seen in my personality. One of my primary strengths as a developer is my incessant dedication to improvement and understanding. I am not one to easily be discouraged when hitting a wall in development, and rather, oftentimes have difficulty stepping away from a problem. I rationalize that there must be a solution to the issue, and at that, a solution I'm capable of understanding and implementing. So, with enough time and dedication, finding the solution is a foregone conclusion. This dedication is not merely limited to the prospect of understanding and growth, but extends to the work itself. As I will often sacrifice my personal time to exceed necessary bounds of an assignment, out of an enjoyment in partaking in the work. This is not to say I am immune to the inherent difficulty of a long and hard day of work, but rather than just presenting stubborn dedication, I also bring a dedication fueled by enjoyment, which is a much more potent aspect to my continued love of development.

Another of my strengths as a developer, that has benefitted the prior greatly, is a lack of unhealthy egotistical entwining with my work. By this, I refer to an active cognisance and consideration of my own personal beliefs, opinions, and capabilities, and do not let this interfere

with my, or team members work. Although this is the case, I also maintain a healthy pride in the work I take on, this has led me to be a very adaptable individual as well as developer. I am able to function efficiently and professionally across a variety of peer environments, to the extent I find team corroboration to be one of the great joys of development, in addition to being an exemplary learning opportunity.

Ultimately, a dedication to, enjoyment with, and lack of egotistical involvement in my work has allowed me to become an adaptable, hardworking, team-forward developer that puts their best foot forward, regardless of circumstance. Due to the technical skills honed in this, and the prior course, I am prepared to enter a multitude of web-development and full stack roles I had not been prior. In addition to more traditional roles in software development and engineering, I am now capable of delving into each stage of the full stack, and have experience doing so, in traditional and cloud-native environments. This positions me well for a multitude of roles across a variety of systems, including assisting prospective transitions to cloud-native environments. This course has provided the skill and confidence to approach an entirely new sector of software development.

Planning for Growth: Synthesize the knowledge you have gathered about cloud services.

Due to the skills honed in this course, expanding upon and scaling the serverless application built throughout the course would not be overly daunting, as AWS has become much more familiar. Thanks to the inherently elastic and flexible nature of AWS, handling scaling and error handling would not be overly difficult. As we saw with the migrated application, AWS offers services that can be leveraged to build and deploy microservice-based applications, like with the Lambdas created independently to run code pertaining to a portion of the applications functionality without provisioning server resources. This microservice architecture allows for horizontal scaling, where the individual services can be scaled independently based on demand. Similarly, centralized error handling and logging can be implemented to monitor and manage errors across microservices. A more detailed explanation of these concepts can be found in the accompanying presentation to the applications migration.

When predicting the cost of an application, there are a large multitude of factors that contribute. In relation to containerization versus serverless, serverless is often more predictable in terms of cost, as you can compare and contrast the various “tiers” of payment across the

platform's services based on need, and adjust accordingly. This also allows one to map out a variety of potential use-cases, each with an estimated cost associated, as AWS provides a pricing model for each tier and a pricing calculator that can assist. Containerization by contrast can be more variable depending on resource utilization, and as a result can be more difficult to predict the cost of.

In terms of growth for the Angular application we migrated, maintaining a cloud-native environment within AWS services would be recommended, as the platform is inherently elastic. Elasticity refers to the ability to acquire resources as you need them and release resources when you no longer need them. Many AWS services do this automatically as part of the service, and this capability is essential for scalability, allowing systems to handle variable workloads efficiently. An understanding of the elastic nature of the platform, combined with an analysis of projected growth and demand, AWS could be configured to provide the most efficient cost to resource utilization possible. Although this method is more efficient than traditional methods of server architecture, careful consideration of use-case, and projected usage is important for maintaining an efficient cost-resource ratio within AWS. A more detailed explanation and consideration of these concepts can be found in the accompanying presentation about the applications migration.