Evan Bush

December 17, 2023

Assignment name: CS 470 Final Reflection

https://youtu.be/ahV9sQhKWDE

Experience and Strengths

Working through this course has taught me numerous skills. For one, it allowed me to work with and better understand Docker and Docker Compose, an essential tool for maintaining application consistency across various environments. Another significant benefit of this course was the skills taught when working with AWS and the many microservices involved. Gaining in-depth knowledge of AWS Lambda, API Gateway, and DynamoDB is highly beneficial when working in cloud environments. Finally, how to set up IAM roles and policies is an invaluable skill to build, ensuring I am always prepared in cloud security management.

My strengths as a software developer are focused on my adaptability, problem-solving techniques, security-conscious mindset, and project management capabilities. Adapting and overcoming problems is crucial when dealing with the evolving landscape of technology. I am committed to always finding the most efficient path forward that produces results. After all courses, I have been able to implement a security-focused mindset while tackling a challenge, ensuring I am constantly improving my work. Lastly, I believe in collaboration and sticking to a set schedule of tasks and specified workloads, allowing me and others to work together and complete objectives without being overwhelmed or burnt out.

Based on this course, the roles that I would be prepared to assume would be full-stack developers, DevOps engineers, or cloud architects. Granted, these would be the junior version of each role, but this course has prepared me to have the needed competencies to excel in these positions.

Planning for Growth

Many approaches to scaling the application and predicting costs regarding microservices or serverless technologies within cloud services exist. With microservices, you can scale each

service independently, reducing the overhead of managing monolithic applications. This allows you to estimate costs since you have more control over the resources each microservice would use. With serverless technologies, developers are freed from server management tasks due to the hosting service scaling and provisioning resources automatically based on the application's needs. However, when it comes to estimating costs, serverless routes can cause a bit more unpredictability. Since costs are associated with actual compute time and resource usage, sporadic spikes in usage can unexpectedly increase the bill significantly.

Selecting the right option between microservices and serverless is always challenging in cloud expansion. Both allow for considerable scalability, and the serverless pay-for-what-you-use model is beneficial and cost-effective when traffic and resource utilization are known ahead of time. Not to mention, the deployment speed with serverless functions is a significant benefit. One thing to consider while deciding is the complexity of managing a distributed system with multiple microservices. There is also the idea of being locked into a single vendor like AWS. Utilizing proprietary services from cloud providers can lead you to rely on them for all needs instead of building the solution with different third-party ones.

Finally, speaking on the roles of elasticity and pay-for-service with cloud services, they assist in decision-making for planned future growth. Elasticity allows resources to be dynamically adjusted based on the current load, allowing the application to handle service peaks without over-provisioning. This reduces the risk of degrading performance during unexpected demand spikes, upholding user satisfaction and service reliability. The pay-for-service model ensures costs directly correlate with resource consumption, allowing a more transparent and predictable approach to billing. This model gives more financial flexibility in planning for growth since costs align with application development and utilization.