Maya Ellison

August 19, 2023

https://youtu.be/u3k0TrNXzMw

CS 470 Final Reflection

The most crucial skills I have developed through this course are related to DevOps and deployment and that is server management and containerization. The server deployment and management tool I was able to require a knowledge of was the cloud service Amazon Web Services (AWS). I gained a better understanding of how cloud computing platforms provide a flexible and scalable environment that allows users to access computing resources on-demand. This was also the first time I touched base on containerization, specifically Docker, to learn how to package applications and their dependencies for consistent deployment. I believe my strengths that make me a good software developer are my ability to quickly learn and adapt to new tools and frameworks, my ability to manage time efficiently to deliver quality work, and my ability to analyze an issue, break it down into smaller components, and devise a solution. Even though I concentrated in software engineering while majoring in computer science, my journey throughout these three years of schooling has prepared me for additional roles. I have grown interested in full stack development and scrum and could possibly see myself pursuing a career in any of these jobs as well.

Serverless architectures offer advantages for improving management, scalability, and efficiency in web applications. The serverless platforms automatically scale based on incoming requests and most commonly offers built-in error handling features. Serverless platforms charge based on actual usage, such as the number of invocations, execution time, and memory usage. This pay-as-you-go model can make cost prediction easier as you only pay for what you use. Serverless is often considered more cost-predictable compared to containers due to this pay-as-you-go model. The model helps to eliminate overprovisioning and make cost predictions easier. For applications with varying workloads, serverless can offer better cost efficiency since you're not paying for idle resources.

Pros of Cloud Services:

1. Scalability – Its ability to scale resources up or down based on demand.
2. Cost efficiency – The pay-as-you-go model means you only pay for the resources you use, eliminating the need to invest in excess hardware.
3. Flexibility - Cloud services provide a wide range of services.
4. Speed – It facilitates rapid deployment and experimentation, enabling quicker time-to-market for new applications and features.

Cons of Cloud Services:

1. Data security/privacy - Storing data off-site raises concerns about data security.
2. Latency - Data transfer to and from cloud data centers might introduce delays.
3. Reliability - Cloud service outages can impact applications and cause downtime.

In the context of planned future growth, both elasticity and pay-for-service are key factors that influence decision-making. Elasticity ensures that the infrastructure can handle increased demand without performance degradation, while pay-for-service aligns costs with usage patterns, allowing for efficient resource planning. This also helps with cost control by ensuring that you're not paying for idle resources during periods of lower demand. Both concepts provide the flexibility needed to accommodate growth while optimizing costs and resource utilization.