Mariam Haji

15th Dec 2023

CS 470 Final Reflection

YouTube link

Learning AWS has always been a goal of mine. As a data analyst, I was eager to understand how AWS works and how to integrate with it. The course has been instrumental in helping me work with and comprehend AWS web services, including setting up endpoints and getting a website operational. This foundational knowledge has given me the confidence to delve deeper into exploring the AWS cloud system and its offerings.

One of my strengths as a developer lies in research and exploring new technology. Through this course, I have honed my research skills, particularly in learning new tech disciplines, whether it's a coding language or a principle. Post-course, I am prepared to advance my role in data analysis, potentially transitioning into a data scientist or delving deeper into business intelligence (BI).

In planning for growth using cloud services, especially microservices and serverless architectures, I've realized that a strategic approach is essential. Microservices segment an application into smaller, independent services, enabling easier scaling and updates of individual components without impacting the entire system. In contrast, serverless architecture emphasizes running backend code without server management, offering inherent scalability and cost-effectiveness since it charges only for the resources used.

Handling scale and errors, microservices allow for the independent scaling of services and isolate failures to prevent system-wide crashes. Serverless architectures automatically adjust to the number of requests and incorporate built-in mechanisms for failure handling. Furthermore, in terms of cost, microservices often rely on containerized environments and incur costs associated with compute resources. Meanwhile, serverless pricing is based on actual usage, which can be more unpredictable with variable workloads.

Both architectures present pros and cons for expansion: microservices offer improved scalability and faster deployment but increase management complexity; serverless provides high scalability without the need for server management but may face higher latency and runtime limitations. Elasticity and pay-for-service models are vital in these architectures for efficiently handling varying loads and aligning costs with usage. Decision-making for future growth involves assessing specific application requirements, technical expertise, and cost implications. Choosing between microservices and serverless should align with an organization's goals, resource availability, and expected application demands, as each offers unique advantages and challenges in web application development.