

Dewayne Staton

April 28, 2024

CS-470 Final Reflection

Throughout the course, I've learned the ins and outs of developing a full-stack application that runs in the cloud with a variety of tools such as containerization and container orchestration, cloud storage, Lambda functions, the Amazon API Gateway, MongoDB to DynamoDB migration, and IAM roles and policies. I was introduced to containerization and container orchestration using Docker. These are essential skills for building and deploying scalable applications in the cloud. This aligns with my goal of becoming a software engineer, where the ability to manage microservices may be required. My understanding of cloud storage solutions like S3 in AWS equips me to handle large datasets and object storage efficiently. My knowledge of other AWS services like Lambda functions and API Gateway also helps demonstrate my proficiency in serverless architecture. Understanding the transition from MongoDB to DynamoDB showcases my ability to migrate data between the NoSQL databases. Learning about IAM roles and policies for cloud security best practices is essential for ensuring secure access and resource management in a cloud environment.

Cloud services provide many tools to develop web applications that are scalable and manageable. Two ideal approaches in cases like this are microservices and serverless architectures. Microservices break down your application into smaller, self-contained services. This allows each service to be scaled independently depending on the workload. In other words, if one service experiences increased traffic, it can be scaled up without affecting the others. Error handling is much easier because if one service fails the rest of the application will keep running. This helps to keep downtime to a minimum. With serverless architectures, you write the functions that automatically execute in response to events, eliminating the need to manage servers. These functions scale automatically based on demand, which can be cost-efficient. Since serverless

functions have the same output for a given input every time, simplifying error handling and retries when failures occur.

Microservices depend on an infrastructure like a virtual machine (VM), and the cost depends on how many resources each service uses. Predicting this cost requires estimating resource usage for each service under various load conditions. Serverless offers more predictable costs. You typically pay based on the execution time and memory used by your functions. Many cloud providers even offer free tiers and pay-per-use options, making cost estimation significantly easier.