CS 470 Final Reflection: Cloud Development

Christian Henshaw - April 28, 2024

Link To Cloud Development Presentation:

https://www.youtube.com/watch?v=5KJPgwHqYd0

This course and the development of this application have helped me grow as a developer in many ways. Importantly, this course provided exposure to specialized knowledge of developing an application in the cloud. Within this foundation is the knowledge of cloud-based deployment tools such as Docker to leverage a containerization migration approach and Docker Compose to streamline the management of a multi-container application. This course also reinforced key development skills such as security and compliance with the intricately configured AWS IAM roles to implement the principle of least privilege as well as continuous integration and continuous delivery (CI/CD) methodologies that are prevalent in today's development environments. In retrospection, my strengths as a developer now include a broad knowledge of specialized tools, adaptability to develop new technologies from web to cloud development, familiarity with development operations (DevOps) procedures, strong problem-solving and critical thinking abilities, effective time management, and a desire to continuously learn. Subsequently, this application has prepared me to perform the duties required of a full-stack software developer as well as a cloud-based software engineer.

Within AWS, we can leverage microservices or other serverless concepts to produce efficiency and scalability throughout our application stack. A microservices approach breaks the application into containers with specific functionalities and can provide a wide range of benefits. As the physical infrastructure requirements are abstracted within a serverless environment, each microservice can be scaled independently to effectively direct resources without affecting other parts of the application. This also benefits error handling as a faulty microservice is isolated and can be easily identified, which enables efficient troubleshooting and application reliability. As these benefits are ingrained in the AWS serverless architecture, each of our application services, such as S3 capacity, Lambda and API Gateway requests, and DynamoDB table sizes, will automatically scale resources to meet user demand. It is also necessary to leverage AWS monitoring tools to understand usage variations across the application and any potential vulnerabilities.

To predict the cost of operating the application, it is necessary to highlight that a cloud-based application in AWS utilizes a pay-per-use model where the cost is directly related to the resources used. To predict the cost of our cloud-based application, we would utilize a tool such as the AWS Pricing Calculator, which can be approximated based on expected service usage metrics such as the amount of Lambda processing. It is also necessary to incorporate applicable scaling strategies such as load balancers and other tools such as persistent monitoring, as these will contribute to our operational costs. While not pertinent to application costs, contingency costs must be considered to cover downtimes, requirement changes, compliance requirements, and so on. Serverless computing utilizes a pay-per-use scheme with auto-scaling resources, whereas certain container environments may have a fixed cost system, resource allocation costs, and resource utilization optimization issues. As a result, serverless computing is generally more cost-predictable than containers, but it is highly dependent on application, scalability, performance, and complexity characteristics.

The prospect of expanding an application brings many pros and cons that must be considered to implement a successful strategy. Importantly, expansion is generally a necessary measure to cultivate user interest and compete effectively in new markets. This also has the potential to streamline existing processes, enhance operational processes, and minimize potential points of failure. Subsequently, expansion can raise awareness, increase customer satisfaction, and gain a cutting edge against competition. However, expansion also requires financial investments, especially upfront, which could potentially be difficult to justify. Additionally, adding new requirements or processes can bring operational challenges to ensure new developments can coexist with existing procedures. Other concerns, such as complying with now-applicable regulations, increased application complexity, market disruptions, and so on, can also be experienced during expansion.

The two key cloud-based development principles of elasticity and pay-per-use are critical for the planned future growth of an application. Elasticity refers to the ability to dynamically allocate resources up or down to meet user demands. Importantly, this characteristic ensures peak traffic hours will perform the same as low traffic hours, eliminating network degradation while maintaining user experiences. For future growth, the implication of elasticity is consistently meeting the expected growth in traffic, which is nearly impossible to match with native hardware. Pay-per-use, as previously mentioned, means a company will only pay for the resources they have used. Each service used within AWS will have a separate cost scheme related to this principle to create the cost evaluation plan for cloud-based hosting. For future growth, this payment model eliminates upfront investments for upgrading physical infrastructure or reoccurring maintenance costs and removes concerns over idle or overloaded resources. Subsequently, these principles ensure an application has the flexibility and freedom to grow as many critical concerns are explicitly addressed.