Janai Cano

April 27, 2024

Final Reflection: CS 470

https://youtu.be/9em4SCE_Thk

CS 470 was my first experience using AWS and its microservices. In today's technical landscape, the cloud and being a full-stack certified developer is highly marketable. Without knowledge of cloud-based technology, and specifically AWS, I could not apply to roles as a full-stack developer or DevOps Engineer. This course opened the door for my continued learning in this field.

My strengths as a software developer is that I now have knowledge of the software development lifecycle, from planning and system design to cloud integration and CI/CD processes and agile methods. I could assume a job as a junior web developer, a full stack engineer, a DevOps engineer, or a data analyst.

Cloud services allow developers to focus on the code and the application itself rather than resource allocation. Using a pay-for-use model, the cost efficiency is unbeatable because you never over pay for unused resources. Any request or response errors would be handled via API gateways. For every response, a different endpoint could be shown to the user, depending on what response code was received. Scaling issues are virtually nonexistent in cloud-based applications, but if more resources were required, we would contact our cloud provider. Predicting upfront cost would also require looking over pay-plans and resource usage with our cloud provider. Containers provide an easier cost prediction, as you know what resources are needed to run containers, so predicting number of users, you could predict a cost. However, if user-demand is unknown, serverless may be easier to predict because you only pay for necessary resources and pay for what you use.

Janai Cano

April 27, 2024

Final Reflection: CS 470

https://youtu.be/9em4SCE_Thk

Some pros in serverless include the elasticity of resources, allowing them to expand and grow as the application grows, or shrinks as it shrinks. This allows developers to continue developing and deploying new features to users, and updating system code without worrying if there are enough resources. Allocating resources would simply be a discussion of cost and budget. Cons could be the cost of storage as the application does grow. Storage can get costly as resources grow, but breaking the application up and across different cloud providers could be a potential solution.