Jarod Timms

10-20-2022

CS 470 Final Reflection

https://www.youtube.com/watch?v=9q7-p4YU3RE

While taking this course, I learned about the fundamentals of cloudless service migration and how to move a full stack application to AWS cloud services. The availability of these services helps me, as a developer, create programs which can become more reliably accessible and stable under any traffic loads. This means that I can continue to develop high performance code within best practices, without the need to regularly maintain hardware to host my applications. This also opens doors for me in the software development industry, especially with large companies which use services, such as AWS, to reach a larger audience. Having a in depth understanding of the AWS SDK and services will ensure that I have skills relevant to the interests of these companies.

The nice thing about these services is that they often have tools to handle error handling and testing when deploying an application to the public. The Amazon API Gateway allows for the testing of various lambdas and security policies by sending test requests through the services and checking the responses for errors. Amazon also hosts supporting features like Canary which allow you to publish new builds to a smaller sample of users alongside the previous public build. This allows for individual users to encounter an issue without ensuring that they will encounter it again as the access more of the service over time, with the trade-off being that developers get all the data from potential errors in the canary build. Scaling becomes incredibly easy to manage when migrating to a cloudless model as the service takes over the responsibilities of managing the hardware side of things while also offering software that manages hardware resources, such as Amazon’s AWS Elastic Cloud Computing which can automatically spin up more hardware as network traffic increases. Within the conversation of “cost predictability”, I would have to say that cost prediction is equally difficult for both container models and serverless models. One could estimate the cost per period-of-time versus the active traffic, however the goal is to increase that traffic over time. The cost efficiency can be predicted more easily by understanding what is desired from the models. Containers can offer more control over your hosted services though software like Docker, which can even be hosted on a cloudless platform, however there is more cost efficiency in using streamlined services like AWS and Azure which can charge on the use of individual services.

Expansion is a huge step for any company, however, can be a hugely costly step as well. The major reason for expanding your company is having confidence that your product has established reasonable demand to warrant a need to scale. A refusal to expand when a service moves into high demand means instability and poor performance which can halt your services from functioning. Another potential con to expansion would be standing up new hardware and hiring staff to maintain it. Expansion can be a lucrative opportunity, but also an inevitable one. Luckily the negative implications of expansion as mentioned above are optional ones with serverless services. Elastic computing and pay-for-service models will allow companies to make a major expansion while spending less money and offloading the responsibilities of hardware maintenance to third parties.