CS 470 Final Reflection

Lucas Kopaskie

October 20, 2022

https://youtu.be/rN1P_XfFrqc

During this course, I have learned a great deal about cloud computer, specifically with regards to AWS. I have learned about what containers are and used Docker to create some. I also learned how to setup and host a static website using S3, DynamoDB, Lambda, and API Gateway. With how prevalent cloud computing is becoming in the world these are all incredibly useful skills to have. I would say that my strengths as a software developer are my problem-solving skills and my ability to understand new concepts and technologies. When presented with a problem I think that I do a good job of working through it and looking at it from different angles to find the best possible solution. I believe that I would do well in roles where I could allow my problem-solving skills to be used. Roles such as software development and software design are more my strengths than working on UI and UX design.

Cloud computing offers many different benefits when it comes to improving efficiency and reducing costs of web applications. A serverless design such as AWS makes scale a near non-factor given how much of the headaches associated with it, such as hardware and staff, and handled by AWS. The cost of hosting a web-application depends largely on what manner you choose to host it. With AWS you can have a largely predictable cost based on your resource usage, though spike and dips can cause this to fluctuate. Using containers on the other hand means that you have to handle a bit more of the management yourself and deal with spinning up and winding down additional containers as needed which can affect costs. In the end I think that serverless is probably more predictable, because there is less that you are directly responsible for so there are fewer variables going into the pricing calculations.

When planning to expand a web application there are several procs and cons to each method. Serverless with AWS comes with a full ecosystem of different services and tools that all work well together and are supported, this is both a pro and a con because while it can be very useful it also locks you into Amazon’s system and makes it harder to pull in elements from outside of it. Containers on the other hand offer far more flexibility and freedom at the cost of having more to manage and having to deal with compatibility between services yourself. Elasticity and pay-for-service are both major factors in how easily and quickly growth can occur. With a highly elastic service such as AWS you can grow more easily with additional servers and other hardware automatically assigned and deployed when needed. This comes with the cost of pay-for-service where your costs will rise as usage does, however. With containers you must manage this yourself which makes it a bit more difficult to scale, though it is still easier than using a traditional server or even something like a virtual machine.