Joseph Collett

CS-470

August 15, 2023

CS-470 Final Reflection

https://youtu.be/vjakwlpf-AY

https://www.youtube.com/watch?v=vjakwlpf-AY

Intro

Throughout the computer science program, we have learned the basics to several programming languages, created front-end applications, and integrated back-end logic and database administration. In particular to this course, we learned more efficient ways of packaging apps with Docker and Docker Compose and migrating services to AWS, in lieu of managing on-site hardware and servers.

Experiences and Strengths

As mentioned previously, this course focused on packaging apps with Docker and migrating applications to the cloud with AWS. Although I am no expert with Docker or AWS, I have been exposed to the material and have enough experience reading documentation to solve and requirement or problem I may run into in the future.

My strengths as a software developer include adaptability and an open-mindedness to learn, grow, and evolve as a software developer. At this point, I am quite confident that I can quickly acquire any information, experience, and skills required to perform an assigned task in regards to software development and devops. My experiences and interests will prepare me for entry-level software development and devops career paths.

Planning for Growth

From tinkering around with AWS, I discovered several ways microservices and serverless may be used to create efficiencies with managing and scaling an application deployed to AWS. First, you can view and analyze the statistics of your network traffic. This will help decide the most appropriate places to locate your services. Then, with the help of a load balancer, you can route the user's requests to the closest servers. Additionally, you can set VMs and containers to auto-scale, so they are given more compute power or additional container instances based on traffic max percentages. For error handling, there are programs like Kubernetes and Prometheus which monitor and provision container up-time and health. You can autoconfg those two to spin up more resources, if others error. Costs are predicted as projections of traffic multiplied by the number of hours for the resources on a daily, weekly, months, and yearly projection. Overall, containers are more predictable, because there are fewer variables determining their hourly cost.

With serverless and Docker containers, companies are not just limited to one cloud provider. To expand and decrease down time, a company may decide to expand services onto another cloud provider. However, this also means investing training time to acquaint employees with the other cloud provider's platform.

Elasticity and pay-for-service help companies project traffic and operating costs. If needed, they can always scale services down to reduce costs, even if it may detriment user experience in times of company economic problems.