Shane Griffith

10/19/2022

CS 470 Final Reflection

Final Reflection

https://youtu.be/zyrFBd6KPbs

This was the second part to our full stack development course here at SNHU. Throughout both courses, I learned how to build a website from scratch, implement it using a localized server system and then how to take a pre-existing website and upload it into the cloud using the AWS suite of features. Before this course, I had very little experience with JavaScript, HTML, or CSS as web development is not necessarily the route I plan to take after graduation. However, the tools that I have learned over the last 16 weeks have been invaluable to me in learning how development tends to work in the "real world" outside of a classroom environment and there were a lot of standards that I learned that will stick with me throughout my career as a software developer.

As a software developer with no real professional experience yet, I view that my strengths lie with being a quick learner (something that I quickly learned is an important skill in this field) the want to learn as much about programming as I possibly can. I'm looking to make a career change slightly later in life and am just trying to absorb as much information as possible to try and make that transition a reality.

As an entry level software engineer, I'd be happy in any role as long as I'd be in a situation where I can learn as much as possible. I plan to move into the aerospace industry and an embedded software engineer. My goal is to assist in building space craft that lands humans back on the moon and eventually onto Mars. I plan to make my first job out after graduating a steppingstone to get me closer to that goal. If this remains true, I'd do anything for any company.

The scalability of using AWS as a hosting platform is almost inherently built in. AWS allows the project only to use the resources that is currently necessary for the performance and traffic that the project is currently seeing. Previously, an internal team would be necessary to manage and upgrade services as they become necessary for the system. There might be a time where the benefits of hosting the project in the cloud becomes too expensive and doesn't make sense for the growth or scheme of the project moving forward. However, utilizing the tools that AWS has available is a huge advantage for many projects. Especially ones for smaller companies who only need the functionality of the website and don't ever plan on growing into something that would require an on-premise infrastructure.

Predicting cost of an implementation over the course of time can be tough. Especially in the beginning. While the revenue and traffic can be easily tracked, it can be difficult to estimate if a company is just starting out due to the fact that there wont be anything to go off of. However, there are somethings that can be done to estimate growth. Then everything else will fall into place things such as

costs and revenue especially. For example, if you take marketing team and start posting ads on social media for example, you can make an assumption that each new user on the platform or website will cost a certain amount of marketing dollars based on marketing trends in the domain in which the company resides. If this same growth is assumed over x number of time then x number of marketing dollars will equal x number of new users. Plan that out over the course of 3 years, 5 years, and 10 years. Then, all of the sudden you have a plan for how not only for the amount of resources the company might need to pay for through the hosting platform, but also if you take how much each user spends on average then you can estimate revenue over the course of that same time frame as well. Ultimately, having access to the analytics behind how the platform is being interacted with is invaluable in this space especially if you estimate over time your platform will have enough traffic to perhaps one day be able to sell ads on your own website which could help offset the hosting costs or even generate extra income to then put back into marketing to gain more yours and they cycle continues in that way. Based on this, I would say that serverless would be much easier to predict the costs on.

The pro's and con's are simple in my opinion. If you are a small local company who doesn't foresee much traffic (relatively) then serverless is by far the way to go due to the lower cost associated opposed to running something on premise that will require a vast infrastructure to implement and keep running.