Final Reflection

J. F. Kiddy

Southern New Hampshire University

19 Nov 23

https://youtu.be/vaNqOqBvgPI

While it has been a long journey writ with many hurdles to overcome in the pursuit of my degree, it has, in the end, been a worthwhile struggle. Throughout this course, in particular, there were several struggles, traveling the width of the nation several times along with my inability to correct my Node mismatch from week three and the follow-on assignments. However, through these struggles, I have found a better understanding, more knowledge and developed the skills necessary to develop full-stack web applications in the cloud, so long as I don't have to use Node. Learning I about Amazons Web Service, and its vast array of tools and Docker's abilities would be potential bullet points to make me a more marketable candidate in the IT field.

I find that the greatest strength I have in regards to software development is that I understand the theory and principles of how all this should work together. Coupled with my years of experience in the Navy and as a Defense Contractor my skills of troubleshooting, learning new system and technologies quickly, ability to work within a team, and attention to detail would benefit from the things I learned here and vice versa. I would imagine that these skills would be ideally suited to being a software developer. However, I am not actively pursuing a job in software development, perhaps in the future, but currently it's my hope to move to a job that requires less travel and more managerial tasks.

If I were to take on the mantal of responsibility as a software developer and the tasks of managing software development fell upon me, some of the tools that we learned about in AWS would be a possible solution. In handling scaling and error handling, AWS analytics like Amazons CloudWatch, coupled with AWS APIs would easily provide help to identify any errors and performance issues. Cost analysis would be a much simpler task, most cloud providers provide pricing calculations allowing me to accurately estimate the cost of running potential applications. And with cost prediction, serverless applications are typically cheaper to run vise container-based applications, because you are only paying for the time your application is actually being used rather than having servers running 24/7. Serverless costs are also usually linear with usage, meaning it usually gets less expensive per unit of measurement the more you use it like buying things in bulk.

For future expansion, the constraints and requirements will have to be considered. If we are developing an app that will be designed to work on multiple OS's, are we developing the application in an environment similar to production and use, are working with inconsistent computer environments, are we working over great distances, or am I working at a startup or a major corporation? These questions will decide whether the we need the security of onsite storage, because you have to be comfortable with relinquishing a certain amount of control of your software. Or the flexibility and scalability of cloud services. But it is hard to ignore the benefits of going serverless such as reduced management overhead, scalability, cost-effectiveness, and elasticity.