

Vitalie Cucuta

02/20/2022

CS 470 Final Reflection

Presentation Link: https://youtu.be/BCdbm_wZ7kI

Final Reflection

Prior to this course, I took the AWS Academy Cloud Foundations course and I strongly believe that it had helped me a lot throughout creating this website. This CS course was a lot more in depth with specific AWS services like Amazon S3, Amazon EC2, DynamoDB, Amazon API Gateway, Amazon IAM, and Lambda. We had also used Docker Compose for containerization of our application development. There was a great deal of knowledge that I had gathered through learning to traverse and utilize these AWS services, especially how they all worked together to bring about a final product that worked flawlessly. Understanding how to properly use the IAM permissions and policies to create a more secure application was probably the number one skill that I am most grateful for because of how important security is. With so many websites and application moving to a cloud architecture, and many of them being through AWS services, these skills I have gained will show my expertise in utilizing AWS for web and application development.

I found understanding how these services work to be quite simple and that's one of my strengths as a developer. I can pick up and retain this knowledge because of the passion I have for software development and my drive to better understand how everything works. Although I may not be completely fluent in my chosen developmental languages, I am always looking to hone my craft. I view this as another one of my strengths as a software developer; knowing your weaknesses and turning them into strengths creates a much stronger foundation and command of software development.

For my future position as a software developer, I am looking to take on any role honestly. This isn't meant to sound desperate, instead, I want to learn more than just one aspect of software development. I want to create new software programs and systems, debug and

maintain existing ones, produce efficient code, improve existing code, and so much more. I believe that for someone to fully refine a skill is to learn as much about it and understand all aspects of that craft. It is just the kind of person that I am and in my career as a healthcare professional I had this same belief as well.

AWS offered a multitude of serverless services for web and application development that brings no surprise as to why they are so popular among cloud development. Not having to manage these services and being able to focus on strictly the development is such a huge benefit for developers. Even out of the ones we used in this course, we didn't fully utilize all the capabilities. For example, in Amazon S3 we can create access point and multi-region access points. These access points simplify data access for any AWS service that stores data in S3. We can create access points for applications as well as users that require access to the bucket, which is a great resource when you build applications with large, shared data sets. A service that we didn't get to use, which I personally think is indispensable in cloud development, is CloudTrail. It allows for logging and tracking of user activity and API usage and that's not only good for security purposes but also to monitor and log any errors that may have occurred. Microservices, on the other hand, require more overhead than serverless cloud computing.

They are typically in house run operations and are handled by IT professionals for support of the underlying infrastructures. This also makes them much more complex and requires a more cross-functional team when it comes to development. Microservices are standalone so they don't depend on each other. A good example of this is how we used Docker Compose for containerization and MongoDB for our database in the beginning. In terms of scalability, serverless computing is exponentially better than microservices because you can scale your needs instantaneously with serverless computing. Since microservices rely on

hardware, it takes much more time to scale them. AWS serverless computing also has a lot of services which help in catching errors, such as CloudTrail, and log these errors when they occur so they can be fixed much faster. In microservices, most of the error handling is done manually.

Compared to serverless computing, microservices can end up significantly more expensive while AWS services are on a pay-as-you-go model. Even if you're not utilizing a container in microservices, you have to pay for it. In terms of cost prediction, containers are technically more predictable because you pay for what you have. Regardless of the demand, you either have or don't have the resources to meet the demand but you can still add resources later. Serverless allows for a lot more flexibility at the price of cost prediction because you can add and remove services as needed. With that being said, AWS Cost Explorer and Cost Profiler can help give you a better understanding of what to expect through cost and usage reports. It also allows for creating budgets and gives you a forecast of the budget. You inevitably have much more knowledge and control over your application through a serverless service rather than through microservices. The ability to dynamically allocate resources is essential when running a large software application in a cloud architecture. Especially with how seamless the services work together in a serverless architecture through AWS.

Overall, using serverless cloud architecture is much more beneficial, in my opinion, than microservices. Even though you can have a much more diverse team through microservice usage, you must allocate a lot of time to the overhead work that comes with the upkeep of the hardware and other resources. Serverless allows for a more direct and application focused development process that produces faster application delivery.