Diego Bez Zambiazzi

December 14, 2021

CS 470 Final Reflection

Presentation URL: https://youtu.be/3o2_tFdv08s

This course will help me in reaching my professional goals by providing the fundamental knowledge about cloud services, their benefits, and trade-offs. I have learned how to use containers to run applications and how to deploy full-stack applications to the cloud. Furthermore, I also learned the benefits of a serverless model and single-table databases like DynamoDB. From my experience working on academic assignments, personal projects, and interview preparation, I find my strengths as a software developer to be on web development, data structures, and algorithms. Moreover, I am always eager to learn new technologies, which is crucial in this career. I feel prepared and willing to assume entry level roles as a software engineer or web developer, to which cloud computing is a valuable skill.

Cloud services provide a solution to the challenge of capacity planning, which can lead to waste of resources and customer dissatisfaction. They offer an abstraction of infrastructure and businesses only need to pay for the resources used. This makes it easier to manage capacity and reduces cost, as hardware investment is not needed upfront. Auto-scaling features also make scalability seamless. Microservices and serverless architectures are also approaches that can be used to produce efficiencies of management and scale in web applications. While microservices mean that applications are divided into smaller services, serverless means that the cloud service provider manages the creation and provisioning of servers for the applications. Handling errors can be challenging and time-consuming as applications scale. For instance, configuring error

handling for multiple Lambda functions can be difficult to maintain. Tools such as AWS's Step Functions can be used to create a serverless error handling workflow that is easier to manage.

To predict costs, I would use tools like AWS's Cost Explorer, which is used to create forecasts, and AWS's Budgets to define a cost threshold for the services being used. When we compare containers and serverless applications, the most predictable option is containers. That is because containers are always available, while serverless are charged based on the number of code executions. Moreover, serverless applications tend to be more costly because the servers used are being managed by the cloud service provider, which, on the other hand, reduces time and resources used to manage servers. To plan for expansion and growth, it is important to use AWS's monitoring tools, such as CloudTrail, which records event and activity logs, and CloudWatch, which is used to analyze service metrics. To plan for regional expansion, for instance, we could use CloudTrail to analyze user identities, traffic origin IPs, and timestamps. Another important factor to consider regional expansion is regional-specific compliance standards. To plan for growth, it is also important to consider elasticity and the pay-for-service model. For instance, since for Lambdas we pay for the number of invocations, their duration, and memory used, they may not be the best choice for heavy traffic. Instead, EC2 (Elastic Compute Cloud) or ECS (Elastic Container Service) could be better options, since the pay model only depends on the instances running, which can be reserved for a predictable time and cost.