Austin Fuchs
October 23, 2022
CS 470 Final Reflection

YouTube Link to Presentation: https://www.youtube.com/watch?v=V2Rqna_CghM

The skills I have developed throughout Full Stack Development II will allow me to migrate and maintain a full stack application in the AWS cloud. I've learned about cloud principles and new technologies such as containerization/virtual machines, container orchestration, hosting static websites on cloud-based storage, serverless architecture, deploying a RESTful API in the cloud, modifying an application's technology stack to use cloud technologies, and securing a cloud application with IAM roles and policies. My strengths as a software developer are that I'm a great communicator, creative and critical thinker, and that I'm passionate about creating quality and secure code and gaining a holistic understanding of a business needs to help implement the best solution possible. This course has prepared me for assuming roles that involve architecting a cloud solution, administering and ensuring the solution operates as intended in the cloud, and developing a solution in a new job.

There are many things that you may need to consider when you need to produce efficiencies and scale in your web application. One thing to consider is scaling the application. As the number of requests grow, API Gateway and our Lambda functions will scale with the application, but it's important to also be aware of how scaling will impact your other downstream services. In our application the only other downstream service is DynamoDB. If you've architected your application using a single-table design, then this is where DynamoDB shines as you'll be making one or two requests per Lambda function call. As our application was not designed with a single table, our database won't scale that well. Using another AWS service like Amazon Kinesis can solve this issue. Kinesis can accept a high number of requests, store the requests, and process them at a speed that our Lambda functions a database can accept.

Another thing to consider is error handling for our Lambda functions. Occasionally, Lambda functions will fail. For example, they may fail when an unhandled exception is thrown or when they run

longer than the configured timeout. To work around this, you can use Lambda Step Functions. These functions allow you to create a workflow that supports error handling. You can configure step functions to respond to errors with conditional logic. Another thing to consider is predicting cost. We didn't touch in this during the course, but this can be done with the Cost Explorer tool in the AWS console. Cost Explorer can create forecasts for your costs based on your prior usage. Cost Explorer will provide a forecast with an 80% prediction interval. However, when it comes to considering another architecture model, such as containers over serverless, it still is more cost effective to use serverless. Containers are more costly as you'll still pay for servers even when they're idle and not taking requests. Serverless is much more elastic, as we'll still only pay for the server time we use.

When it comes to expanding our cloud infrastructure one thing to consider to region deployment. Deploying to another region means having access to another market, but there are a few things to consider when reviewing a region for deployment. We'll need to consider if the way our workload manages data is compliant within the countries of the new region. We'll also need to consider the latency our customers will experience and whether that will necessitate using or adding a different region. Cost will also be a major factor, as AWS services are priced differently from one region to another. Also, as our business grows and we add or upgrade to newer AWS services and functionality in one region, we'll need to consider how that will affect other regions as new AWS services are gradually rolled out. This means that we won't be able to use these services in smaller regions until AWS implements them in their regions.

Regardless of what expansion will look like in the future, I believe that elasticity and the elastic price model that results from our usage will still be how we operate our product. Outsourcing our server management to AWS and migrating to a serverless architecture means that we only pay for the compute

and storage that we use, which will always be more cost effective than purchasing and operating servers

ourselves.