Nicholas Renna

06/28/2024

CS 470 Final Reflection

YouTube Presentation link: https://youtu.be/_53V4HuOCl4

Experiences and Strengths:

The full stack development course allowed me to gain new skills in cloud development with Amazon Web Services and utilizing containerization. With more companies moving operations to a cloud-based service the skills learned such as using Lambda functions, creating APIs or using containers to store parts of an application will help market me as a candidate in the Computer Science field. My new skill utilizing the cloud to host a full stack application is a useful as cloud-based services offer a business more efficiency, ability to grow with your application (scalability) and cloud technology uses the latest and most advanced features to support application development. With these newly required skills I can now work in cloud development projects, and develop secure, efficient applications that support many uses.

My Strengths as a software developers include attention to detail, software architecture, critical thinking skills and the ability to understand business operations. When creating a full stack application that uses cloud-based technology, having attention to details is important because of the many steps it takes to migrate the application to the cloud. Also, I gain new knowledge in software architecture and Amazons Web Service microservices such as API Gateway, S3 bucket storage or Lambda Function creation. The last Strength I gain from the Computer Program and the Full stack Development sources was understanding business operations that can assist teams that migrate their application or parts of their systems to a cloud-based environment. The types of roles I am prepared to assume are entry level developer jobs with web applications, cloud architect, system operations administrator and cloud security specialists.

Planning for Growth:

With cloud-based technology the developer can often use serverless or microservices (Lambda, DynamoDB) that have automatic scalability to support applications. For example, if your application is growing with increased usage then serverless components can be scaled to adjust to current needs. When more API calls or other actions are being requested on your application, microservices and serverless environments can support those needs when they increase or decrease with scaling and auto scaling for Lambda functions.

Error handling will be handled with Lambda functions that use HTTP headers to return different exceptions or error messages. For example, in AWS Lambda the developer will create a Lambda function using the integration type of Mock that will work with specific headers on each REST method that is used in the application. Basically, Lambda allows the developer to route an exception or error message for specific HTTP methods or REST method calls to support error handling.

To predict costs in microservices or serverless environments there are some options which will allow you to manage your costs, predict future costs and plan for costs during peck application usage. In Amazon Web Services they have feature such as Cost Calculator that will estimate based off previous usage, Cost Explorer to help analyze services like Lambda functions, and other cost management tools (budget and alert services) that assist the developer or business with predicting costs. To consider between containers or serverless will depend on the application usage. For example, if your application is consistent with usage or has similar volume of users accessing your application each time then containers might be a more cost predictable option. But if your application usage or user volume is unpredictable or has peck seasons then a serverless environment would be able to support increased or decreased user volume at any given time.

Pros for plans of expansion include the latest and most current technology is being used with serverless and cloud application development which means you will be uses the best technology to become both efficient in use and cost effective. Some other pros for expansion mean you could now use new services for your application such as using S3 buckets, Lambda and the API Gateway to manage your application. Paring the Cloud Watch with these services allows you to complete control of your application and monitor all alerts including costs. Cons for expansion into the cloud and serverless include not having full control of resources now (third party provider now manage services), without monitoring your costs frequently or setting up alarm’s costs can increase without warning, and expansion can have a complex set up in the short run.

In planned future growth cloud technology with serverless environments provides the application with elasticity which allows the application and the services it uses and provides to increase or decrease resources as needed at any given time. When planning for future growth elasticity is a great feature that cloud development provides for the developers and business partners. Another great service that clouds development is known for is the pay-for-use model which can in the long run save application expenses. One example is that when your application uses resources such as API calls then those will be part of pay-for-use services but if your application does not use API such as sitting idle then no payment is required.

References:

Amazon CloudWatch. (n.d). Amazon Web Service. aws.amazon.com. APM Tool - Amazon CloudWatch - AWS

Forecasting with Cost Exployer. (n.d). Amazon Web Service. aws.amazon.com. Forecasting with Cost Explorer - AWS Cost Management (amazon.com)

What is AWS Billing and Cost Management? (n.d). Amazon Web Services. Aws.amazon.com What is AWS Billing and Cost Management? - AWS Cost Management (amazon.com)