Seth Reeise

8/14/2022

CS 470 Final Reflection

**Cloud presentation video link:** https://youtu.be/u-9Z7CqqBSE

- **Experiences and Strengths:** Explain how this course will help you in reaching your professional goals.

    - What skills have you learned, developed, or mastered in this course to help you become a more marketable candidate in your career field?
    - Describe your strengths as a software developer.
    - Identify the types of roles you are prepared to assume in a new job.

The skills I have learned throughout this course include Angular, JavaScript/TypeScript, HTML, CSS, Docker, mongo, and a multitude of AWS including API Gateway, Lambdas, S3, and DynamoDB. This includes migrating a full stack application to the cloud.

My strengths as a software developer includes back end and front-end development and my vast knowledge of AWS from work and throughout this course.

The types of roles I am prepared to assume in a new job include backend, frontend developer or full stack developer.

- **Planning for Growth:** Synthesize the knowledge you have gathered about cloud services.

    - Identify various ways that microservices or serverless may be used to produce efficiencies of management and scale in your web application in the future. Consider the following:
        - How would you handle scale and error handling?
        - How would you predict the cost?
        - What is more cost predictable, containers or serverless?
    - Explain several pros and cons that would be deciding factors in plans for expansion.
    - What roles do elasticity and pay-for-service play in decision making for planned future growth?

To handle scaling going forward I would use AWS to deploy on the cloud so that there is automatic scaling. For error handling I would use the AWS SDK and CloudWatch logs.

Predicting the cost of these service includes having a breakdown of all the services needed and calculating the cost of each one based on AWS's pricing. This can be done with the AWS's price calculator.

Determining the cost effectiveness of containers vs serverless depends on the size your application. Because serverless has elasticity it more cost predictable to use containers as you know exactly how much use their will be.

Pros for expansion include less overhead for on prem servers and less wasted resources during downtime.

Cons for expansion include possible higher cost initially and little knowledge of cloud services. Elasticity and pay-for-service play one of the most important factors for future growth. You can try to predict future growth and plan accordingly,  but this may not always work. Elasticity allows for your calculations to be off while not costing extra. The pay-for-service is also equally as important, if your application is small in terms of usage the benefits of pay-for-service far outweigh any cons. If you application is not being used, you are not being charged for downtime.