Brian Modzelewski
4/22/2024
CS470 Final Reflection

- 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?

      I've enhanced my proficiency in Docker and acquired the ability to transfer applications from local environments to the cloud. These competencies enhance my attractiveness as a job candidate, especially in light of the high demand for individuals knowledgeable in cloud infrastructure. To further bolster my qualifications, I aim to expand my understanding of additional cloud services and obtain certifications, such as those offered by AWS.

    - Describe your strengths as a software developer.

      I consider my greatest asset as a software developer to be my aptitude for swiftly resolving issues by reviewing code and error messages. Additionally, I excel at elucidating complex code concepts in easily understandable terms, making them accessible to individuals of varying technical backgrounds.

    - Identify the types of roles you are prepared to assume in a new job.

I am prepared to take on roles related to cloud technology, including positions such as solutions architect, cloud security engineer, or cloud security architect.

- 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?

          For scaling microservices, my approach involves implementing load balancers to evenly distribute traffic across instances and facilitating dynamic communication among services. To effectively manage errors, I rely on logging and monitoring systems to promptly detect and address any issues as they arise.

          For scaling with serverless architecture, I would leverage AWS Lambda's auto-scaling features and design the application using stateless functions capable of handling increased requests seamlessly. Regarding error management, I would employ cloud monitoring tools to promptly detect and resolve any issues that arise.

        - How would you predict the cost?

To forecast costs, I'd establish monitoring and alerts to monitor usage and expenses closely. Additionally, I'd utilize AWS's cost estimation tools to simulate various load scenarios and their associated costs. Analyzing usage patterns and fine-tuning resource allocation to optimize efficiency would further aid in predicting expenses accurately.

- What is more cost predictable, containers or serverless?

  I perceive serverless as more predictable since costs correlate directly with the number of executions. Conversely, containers may offer predictability when dealing with a stable and consistent workload.

- Explain several pros and cons that would be deciding factors in plans for expansion.

Microservices are advantageous for agile development and support continuous deployment. However, improper orchestration of services can lead to drawbacks such as increased latency, highlighting potential downsides of microservices architecture.

Serverless computing offers several benefits, including freedom from server management, high availability, and scalability. Moreover, its pay-as-you-go pricing model can

result in cost savings. However, drawbacks include limitations on third-party control and the potential for vendor lock-in situations.

- What roles do elasticity and pay-for-service play in decision making for planned future growth?

Elasticity automatically adjusts resource allocation to match demand without manual intervention. This capability ensures consistent performance during peak loads while minimizing resource waste during periods of reduced demand.

Paying for services as you use them aids in financial planning by eliminating the necessity for a significant upfront investment in infrastructure. This approach also mitigates financial risk by allowing costs to align more closely with usage.