Andreya Colvin

Date: December 16th, 2024

Assignment Name: CS 470 Final Reflection

Experiences and Strengths

Throughout CS 470, I learned and applied skills essential for developing a full stack web application in the cloud. This course helped me master technologies like microservices, containerization, serverless architecture, and cloud deployment. These skills have strengthened my understanding of cloud services and made me a more marketable candidate in my career field.

Skills Developed

- Cloud Deployment: I successfully deployed a MEAN stack application using AWS services, demonstrating my ability to leverage cloud platforms for real-world projects.
- Containerization: Using Docker and Docker Compose, I learned how to package and deploy microservices efficiently, ensuring scalability and easier management.
- Error Handling and Monitoring: Implemented robust error-handling mechanisms and learned to use monitoring tools like AWS CloudWatch to maintain application reliability.
- Scalability: Designed the application to handle future growth, applying concepts of elasticity and load balancing.

Strengths as a Software Developer

- Adaptability: My ability to quickly learn and apply new technologies, such as containerization and serverless computing, is a key strength.
- Problem Solving: I excel at breaking down complex problems and delivering practical, scalable solutions, as seen in my web application project.
- Collaboration: I effectively collaborated on tasks like API integration and testing, showcasing strong teamwork and communication skills.

Roles I Am Prepared For

With my experience in this course, I am well-prepared for roles such as:

- Cloud Developer: Leveraging cloud services to design and deploy applications.
- DevOps Engineer: Building automated pipelines for application deployment and maintenance.
- Full Stack Developer: Creating dynamic and scalable web applications using technologies like the MEAN stack.

Planning for Growth

Future Use of Microservices and Serverless

To ensure the application's efficiency and scalability in the future:

1. Scale and Error Handling**:**

- Implement auto-scaling features using AWS Lambda for serverless functions or Kubernetes for containerized microservices.
- Use distributed tracing tools to monitor and debug system-wide errors efficiently.

2. Cost Prediction:

- Monitor usage patterns with AWS Cost Explorer or Azure Cost Management to estimate future expenses.
- Serverless solutions offer cost predictability for sporadic workloads, while containerization is more predictable for consistent, high-demand workloads.

3. Pros and Cons of Containers vs. Serverless:

- Containers:

Pros: Greater control, compatibility with legacy systems, predictable costs for high usage.

Cons: Requires more management and oversight infrastructure.

- Serverless:

Pros: Lower maintenance, cost-efficient for sporadic tasks, easier scalability.

Cons: Unpredictable costs for high-traffic applications, potential vendor lock-in.

Elasticity and Pay-for-Service

- Elasticity: Ensures that resources scale up or down dynamically based on demand, optimizing performance while minimizing costs.

- Pay-for-Service Model: Offers cost efficiency by charging only for resources used, making it easier to budget for expansion.

Final Considerations

For planned growth, I would recommend adopting a hybrid approach:

- Use serverless for low-traffic, infrequent tasks to save costs.
- Employ containerized microservices for high-demand, critical components requiring more control and customization.

Presentation Link

https://youtu.be/dbUmnUbKDl4