

Isaac Jang

12/20/2024

## CS 470 Final Reflection

**YouTube Link:** <https://youtu.be/vyBYjdBZOVQ>

### Overview

This project involved migrating a full stack web application to the AWS cloud environment using serverless architecture. The application leverages services such as AWS Lambda, API Gateway, DynamoDB, and S3. Below, I detail my experiences, strengths, and plans for growth as a software developer, along with how this course has helped me achieve my professional goals.

---

### Technologies Used

- AWS Services: API Gateway, Lambda, S3, DynamoDB
  - Frontend: Angular, HTML/CSS
  - Programming: JavaScript, JSON
  - Tools: GitHub, Docker
- 

### Experiences and Strengths

#### How This Course Helps My Professional Goals

This course gave me hands-on experience with cloud development, serverless architecture, and microservices. These skills are essential for modern software development and align with my goal of becoming a cloud-native software developer. By learning to manage AWS services and troubleshooting serverless solutions, I have gained valuable, industry-relevant expertise.

### Skills Gained

- Serverless Architecture: Proficient in creating and deploying AWS Lambda functions, API Gateway configurations, and integrating with DynamoDB.
- Cloud Security: Understanding of IAM roles and policies to secure applications.
- Full Stack Integration: Connecting the frontend hosted on S3 with the backend APIs.
- CORS Implementation: Manually adding CORS configurations to enable cross-origin requests.

### My Strengths as a Software Developer

- Strong problem-solving skills and the ability to troubleshoot complex cloud issues.

- Proficiency in modern technologies like AWS services, serverless architecture, and database management.
- Attention to detail in deploying secure and scalable solutions.

### **Roles I Am Prepared For**

- Cloud Developer: Focused on building scalable cloud-native applications.
  - Backend Developer: Designing APIs and managing data integration using serverless architectures.
  - DevOps Engineer: Deploying and managing infrastructure using cloud services.
  - Software Developer: Creating full-stack applications with modern tools and frameworks.
- 

### **Planning for Growth**

#### **Microservices and Serverless for Efficiency**

Microservices and serverless architectures allow for efficient management and scaling of web applications. Future plans include:

- Scaling: Use AWS Auto Scaling and DynamoDB's on-demand capacity mode for dynamic scalability.
- Error Handling: Implement structured logging and AWS CloudWatch for monitoring and alerting on errors.
- Cost Prediction: Leverage AWS Cost Explorer to monitor usage and estimate expenses.

#### **Cost Predictability: Containers vs. Serverless**

- Serverless: Costs are directly tied to usage, offering predictable billing for sporadic workloads.
- Containers: Offer better cost control for sustained, predictable workloads but require more management effort.

### **Pros and Cons for Expansion**

- Serverless Pros: Automatic scaling, lower operational overhead, and pay-for-use pricing.
- Serverless Cons: Cold start latency, vendor lock-in, and potential costs for high traffic.
- Containers Pros: Portability and consistent performance for long-running applications.
- Containers Cons: Requires managing clusters and provisioning resources.

### **Elasticity and Pay-for-Service**

Elasticity allows applications to handle fluctuating traffic by scaling resources up or down automatically. Pay-for-service ensures that costs align with actual usage, making it easier to manage budgets during planned growth.

## **Final Thoughts**

This project demonstrated the power and flexibility of serverless architectures and cloud services. I am confident in my ability to design, deploy, and manage modern cloud applications, making me a strong candidate for roles in cloud development and beyond. I look forward to applying these skills to real-world scenarios and continuing to grow as a developer.