



# **CNG-495 Capstone**

## **Project Stage 2: Progress**

### **Report & Updated Proposal**

**Cem KARASU**

**2315398**

**Metehan YILMAZ**

**2315638**

**11/16/23**

## Table of Contents

Project Topic .....	3
Project Description .....	3
Cloud Delivery Models.....	3
Architectural Overview.....	3
User Authentication: .....	3
Static and Dynamic Content Delivery: .....	3
Microservices: .....	3
Workflows:.....	3
Image Recognition Workflow:.....	4
Data Storage and Processing: .....	4
Search MicroService:.....	4
Diagrams/Figures .....	5
Expected Contribution .....	6
Front-End Swift Mobile Application Development: .....	6
Report Writing:.....	6
Cloud Computing Tasks: .....	6
References .....	6
Milestones Achieved: .....	7
Milestones Remained:.....	8
List of Futures to be delivered: .....	8

## **Project Topic**

Developing a Swift-based e-commerce application on the AWS cloud infrastructure with integrated machine learning capabilities.

## **Project Description**

This project aims to create an e-commerce application that users can list their products by adding new items. AWS Cognito manages the authentications of each user and customer. A customer can search and add products to their cart. Payment microservice is not included in this project.

Image recognition of Amazon Rekognition is used when an image is uploaded to add an item. The main purpose of this functionality is to help through the filling item details by automatically filling some fields based on previously specified configurations of the identified object.

## **Cloud Delivery Models**

In our project, we plan to adopt serverless computing as a fundamental part of our architecture. We aim to leverage AWS Lambda for its remarkable ability to automatically scale and handle event-driven tasks. Our focus will be on the simplicity and scalability provided by Lambda and AWS Cognito for serverless user management. We will provide software developed in swift.

- **SaaS (Software as a Service):** We will provide an e-commerce application that users can list products.

## **Architectural Overview**

### **User Authentication:**

AWS Cognito for serverless user management, ensuring secure and scalable authentication.

### **Static and Dynamic Content Delivery:**

CloudFront for routing static requests to S3 bucket holding static data (e.g., images).

API Gateway as a secure single point of access for dynamic requests, routing to microservices.

### **Microservices:**

Four microservices - Inventory, Cart, Order, and Search - implemented using AWS Lambda.

DynamoDB for data storage used by Inventory, Cart, and Order microservices.

### **Workflows:**

Three workflows: New Order Workflow, Cancel Order Workflow (handled by Order MicroService),

and Image Recognition Workflow (handled by Inventory MicroService using Lambda Step Functions).

### **Image Recognition Workflow:**

Triggered when an image is uploaded to S3.

Inventory MicroService Step Function invokes Amazon Rekognition.

Recognized labels stored in DynamoDB, enhancing product information.

### **Data Storage and Processing:**

DynamoDB used by Inventory, Cart, and Order microservices.

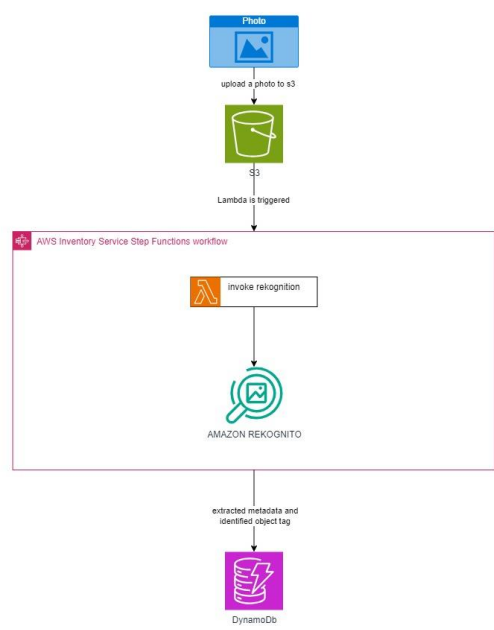
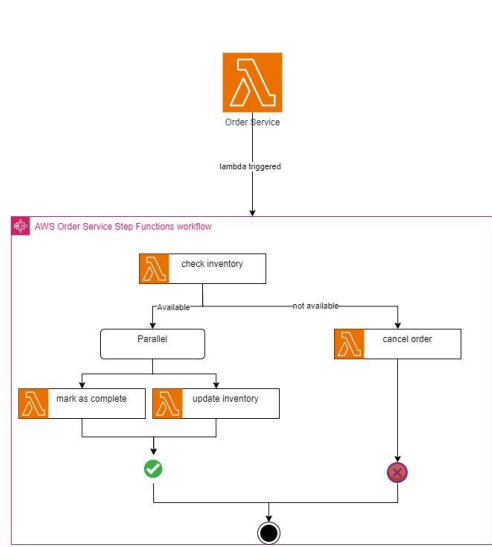
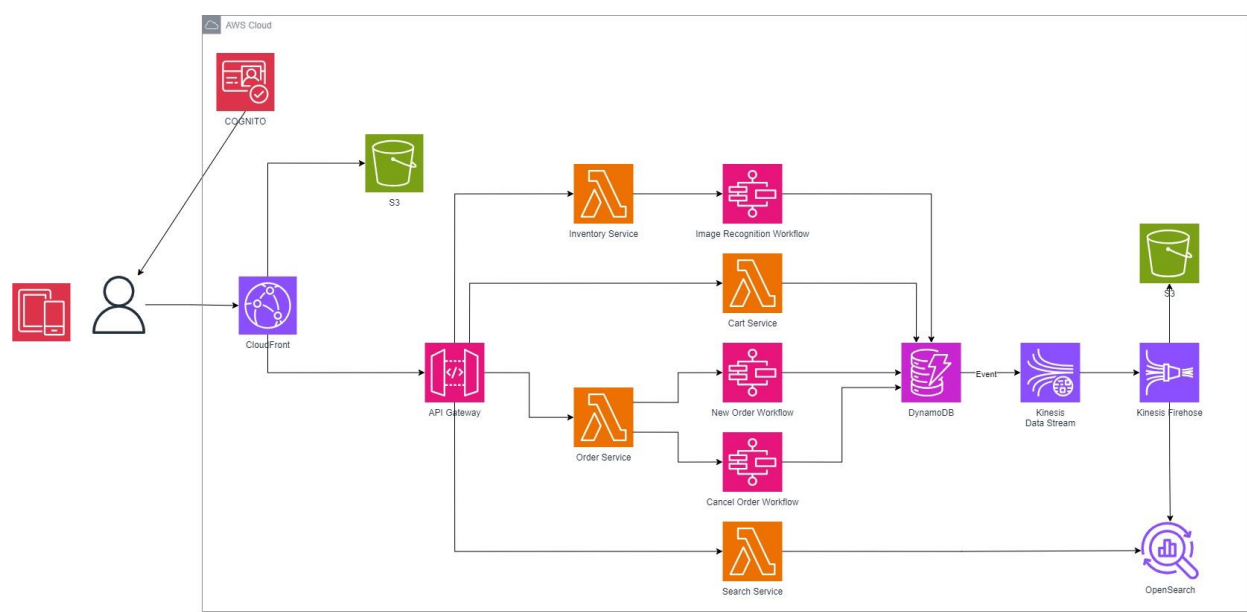
Kinesis Data Stream for data streaming.

Kinesis Firehose for efficient data delivery to S3 Bucket and OpenSearch.

### **Search MicroService:**

Utilizes OpenSearch for efficient and scalable search functionality.

Diagrams/Figures



## Expected Contribution

We've outlined how each team member will contribute to our project, considering our individual strengths and areas of expertise:

### Front-End Swift Mobile Application Development:

Cem will be responsible for developing 100% of the front-end Swift mobile application. This includes creating the look and feel of the app to ensure a smooth and easy shopping experience.

### Report Writing:

Metehan will handle most of the report writing, covering 90% of this task. Cem will contribute 10% to the report writing. We'll keep communication open throughout to make sure the report is well-rounded.

### Cloud Computing Tasks:

To make sure the workload is balanced, we're dividing cloud computing tasks into 60% and 40%.

Metehan will take on 60%, and Cem will manage the remaining 40%. Both of us will work closely on all cloud computing tasks to understand and implement them effectively.

We understand the importance of staying in touch and working together closely. This plan is designed to make the most of each team member's skills, ensuring our platform on the AWS cloud comes together successfully.

## References

[\*AWS Lambda vs. EC2: Which is Right for Your Use Case?\*](#)

[\*AWS Lambda Pricing\*](#)

[\*AWS Step Function Pricing\*](#)

[\*Amazon Cognito Pricing\*](#)

[\*Amazon CloudFront Pricing\*](#)

[\*Amazon S3 Pricing\*](#)

[\*Amazon API Gateway Pricing\*](#)

[\*Amazon SWF \(Simple Work Flow\) Pricing\*](#)

[\*Amazon DynamoDB Pricing\*](#)

# Stage 2: Progress Report

## Milestones Achieved:

Milestone Name	Description	Week	Member
Proposal	Proposal for this project is prepared to be submitted.	W7	All
Self-Study	Each group members get ready for the continuation of the project by doing self-study on the AWS topics.	W9	All
API Gateway	Triggers for the Lambda Functions and S3 bucket are created.	W10	Metehan
Lambda Functions	“items” and “inventory” Lambda Functions are implemented using API Gateway endpoints. More specifically search, add, and delete functionalities are implemented.	W10	Cem
DynamoDB	Simple item_table is created to be used by Lambda Functions to store data to DynamoDB.	W10	Cem
GitHub Commits	GitHub commits are done by a group member even though codes are written by either of the group member.	W10	All
Progress Report	Stage 2 of the capstone project.	W11	All

Figure: DynamoDB

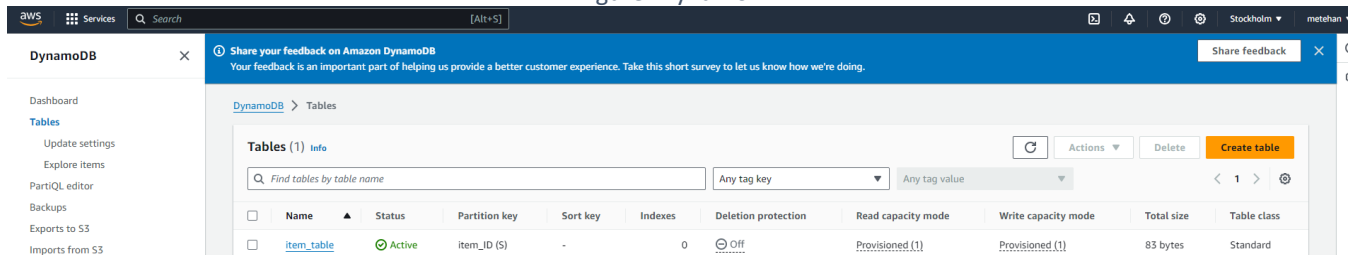
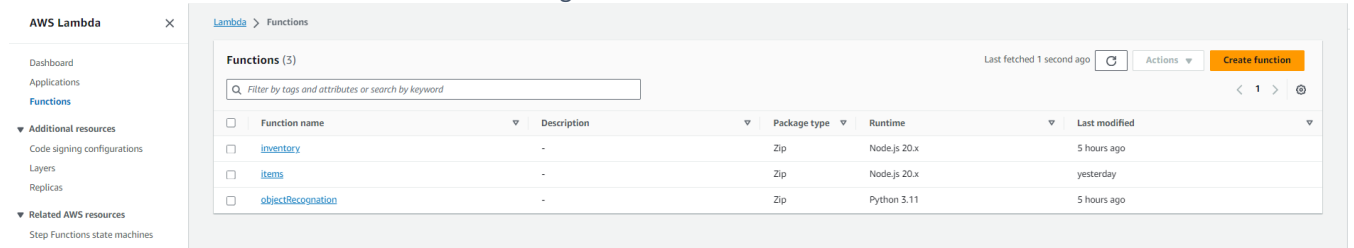


Figure: Lambda Functions



<https://github.com/metumetehan/e-commerce-CNG-495.git>

## Milestones Remained:

Milestone Name	Description	Week	Member
UI	The front-end of the project has not been completely implemented. Therefore, it is not included in this report.	W11	Cem
Image Recognition	Implementing Amazon Rekognito.	W11	Metehan
Integration of UI	Amazon Rekognito and UI integration. Add functionality to suggest and autofill information while adding an item. Uploading an image triggers Rekognito and object recognition information is retrieved to the UI.	W12	All

## List of Futures to be delivered:

- Server code.
- Database backup.
- <https://github.com/metumetehan/e-commerce-CNG-495.git>
-