



CNG-495 Capstone

Project Stage 1: Project Proposal

Cem KARASU

2315398

Metehan YILMAZ

2315638

2023/11/16

Table of Contents

Project Topic.....	3
Project Description	3
Cloud Delivery Models	3
Architectural Overview	4
User Authentication:.....	4
Static and Dynamic Content Delivery:	4
Microservices:	4
Workflows:	4
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

Project Topic

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

Project Description

We propose to develop a robust e-commerce application using Swift for mobile programming and AWS as the cloud infrastructure. The application will provide users with a seamless shopping experience, incorporating features such as user authentication, real-time inventory management, shopping cart functionality, and order processing. The application will leverage various AWS services to ensure scalability, security, and efficient management of data and workflows, including the integration of machine learning through Amazon Rekognition for image recognition.

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. Lambda will be our primary choice for swift, scalable operations without the hassle of infrastructure management.

Additionally, we are considering the use of another service called Amazon EC2, providing resizable compute capacity in the cloud. Unlike Lambda, EC2 will offer us more control over the environment, enabling us to launch virtual machines with various hardware and software options. We anticipate using EC2 for specific scenarios like continuous processing or tasks that require a more hands-on approach. However, it's essential to note that for our initial version, our focus will be on the simplicity and scalability provided by Lambda and AWS Cognito for serverless user management.

- **SaaS (Software as a Service):** AWS Cognito will manage serverless user management, ensuring secure and scalable authentication.
- **PaaS (Platform as a Service):** AWS Lambda will be our primary choice for microservices, and AWS Step Functions will seamlessly orchestrate workflows.
- **IaaS (Infrastructure as a Service - Planned for Future):** While EC2 instances will offer more control for server-side processing, we do not plan to incorporate them into the release. They might come into play in future versions as our project evolves.

It's like having a dynamic toolkit of cloud services, each uniquely positioned to play a specific role in building our application.

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.

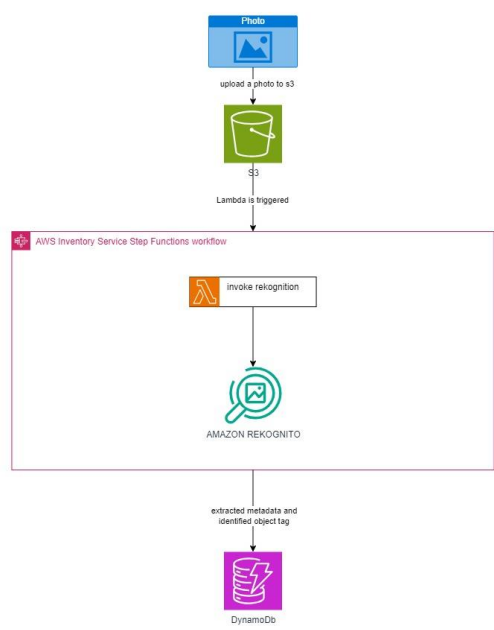
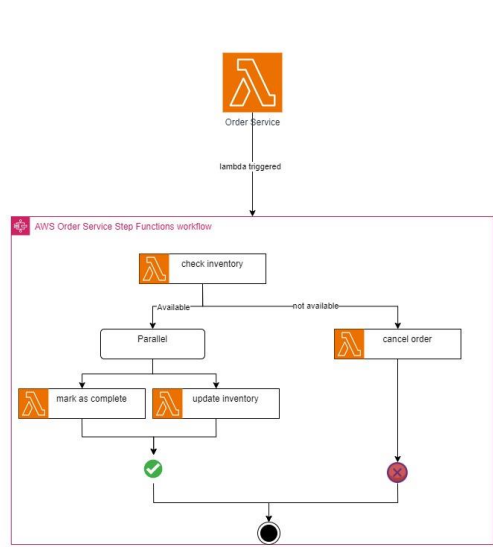
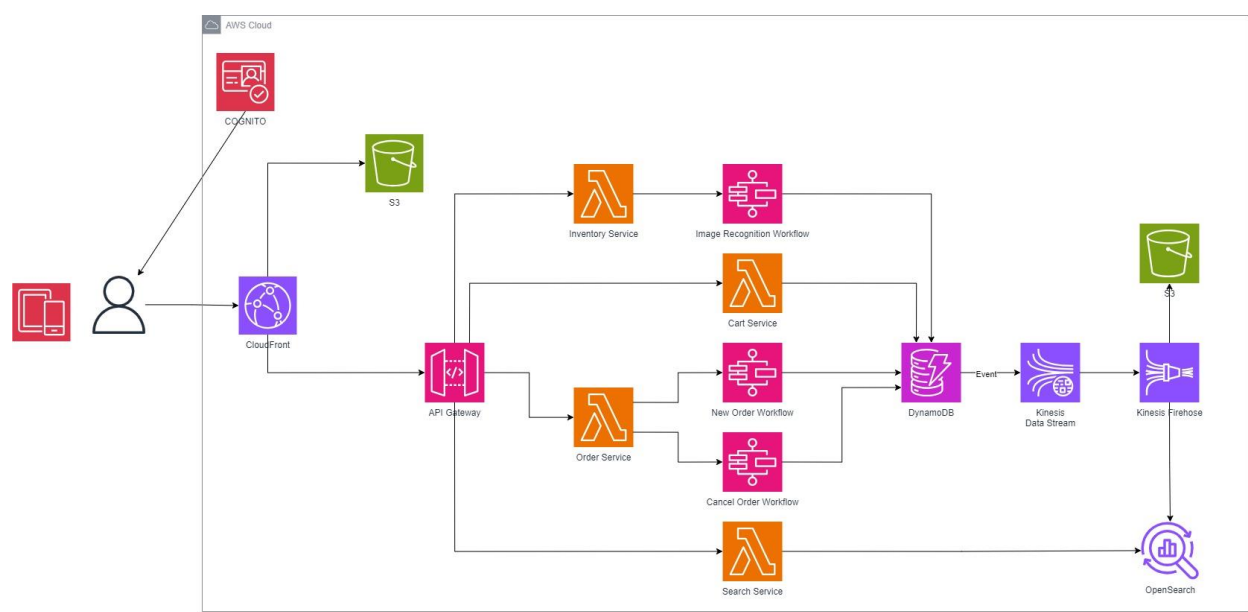
Kinesis Data Stream for data streaming.

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 in charge of 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 Swift-based e-commerce application 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*](#)