Serverless Sentiment Analysis of Social Media Data Using AWS Comprehend

A Project Based Learning Report Submitted in partial fulfilment of the requirements for the award of the degree

of

Bachelor of Technology

in the Department of Computer Science & Engineering

Cloud Based AI/ML Speciality (22SDCS07A)

Submitted by

2210030417: C Shashank

Under the guidance of

Ms. P. Sree Lakshmi

Department of Computer Science and Engineering

Koneru Lakshmaiah Education Foundation, Aziz Nagar

Aziz Nagar – 500075

March - 2025.

Introduction

Social media platforms like Twitter, Facebook, and Instagram generate a lot of text data every day. Understanding public opinion from this data is important for businesses and researchers. In this project, we use AWS Comprehend to analyze the sentiment of social media posts. The system collects text, processes it through AWS services, and shows if the sentiment is positive, negative, neutral, or mixed. Using serverless technology helps handle large amounts of data without needing to manage servers.

AWS Services Used

API Gateway: Manages incoming requests and directs them to the right service.

AWS Lambda: Runs code only when needed, saving costs.

AWS Comprehend: Analyzes text and finds the sentiment.

DynamoDB/Amazon RDS: Stores the sentiment results for future use.

These services work well together, creating a scalable, reliable, and cost-effective solution.

Project Purpose and Expected Outcome

The purpose of this project is to build a system that automatically analyzes social media text and determines its sentiment. The expected outcome is a scalable, serverless solution that provides real-time insights into public opinion, helping businesses make better decisions.

Methodology

Architecture and Workflow

1. The frontend collects social media text and sends it to the API Gateway.
2. API Gateway sends the requests to AWS Lambda.
3. AWS Lambda processes the text and sends it to AWS Comprehend.
4. AWS Comprehend analyzes the text and finds the sentiment (positive, negative, neutral, or mixed).
5. The sentiment results are stored in DynamoDB or Amazon RDS for future analysis.

This architecture is simple, efficient, and easy to scale.

AWS Service Interaction

The frontend sends text data to the API Gateway, which acts as the entry point.

API Gateway passes the data to AWS Lambda.

AWS Lambda runs the processing code only when a request is made, ensuring cost efficiency.

Lambda sends the text to AWS Comprehend, which uses natural language processing (NLP) to detect the sentiment.

Once the sentiment is determined, Lambda stores the result in DynamoDB or Amazon RDS.

Justification for Service Selection

AWS services were chosen because they offer scalability, reliability, and cost-effectiveness. API Gateway handles incoming requests easily, AWS Lambda removes the need to manage servers, AWS Comprehend offers powerful NLP tools, and DynamoDB/Amazon RDS stores data efficiently. These services ensure smooth operation without infrastructure management.

Implementation Steps

AWS Infrastructure Setup

1. Set up API Gateway:
   - Create an API in API Gateway.
   - Configure routes to send requests to AWS Lambda.
2. Create Lambda Functions:
   - Write a Python function to process the text and call AWS Comprehend.
   - Set permissions so Lambda can access API Gateway and DynamoDB/RDS.
3. Configure AWS Comprehend:
   - Set up the Comprehend service and integrate it with the Lambda function.
4. Set up Database:
   - Create a DynamoDB table or Amazon RDS instance to store sentiment results.

Security Policies, IAM Roles, and Access Controls

1. IAM Roles for Lambda:
   - Create an IAM role that allows Lambda to access Comprehend and the database.
2. API Gateway Access:
   - Restrict API Gateway access using API keys or authentication.
3. Database Security:
   - Set access policies to control who can read and write data.
4. Encryption:
   - Encrypt sensitive data in transit and at rest.

5. Monitoring and Alerts:

   - Set up CloudWatch to monitor the system and send alerts if anything goes wrong.