AWS Sentiment Analysis Project

Daniel Mehta n01753264

Overview

This project is a **sentiment analysis web application** deployed on an **AWS EC2 instance**. It uses a **Flask web server** to provide an interface for **users to input text and receive sentiment predictions** using AWS AI services.

This implementation returns only the **overall sentiment classification (e.g., Positive, Negative, Neutral),** rather than showing all individual sentiment scores together.

The application consists of:

- **Front-end UI (index.html)** for user input.
- **Flask back-end (main.py)** to process text and interact with AWS services.
- **Helper script (comprehend.py)** for sentiment analysis (AWS Comprehend for sentiment analysis).

Setting Up the Application in EC2

Step 1: Connect to the EC2 Instance

Use SSH to connect:

ssh -i your-key.pem ec2-user@your-ec2-ip

Step 2: Install Required Dependencies

Ensure your system is updated and install the necessary Python libraries:

sudo yum update -y
sudo yum install python3-pip -y
pip3 install flask boto3

Step 3: Transfer Project Files

You can either create the files manually using nano:

nano main.py # Copy & paste the Python backend code
nano comprehend.py # Copy & paste the AWS AI logic
nano index.html # Copy & paste the front-end HTML

Or

Transfer them using SCP

scp -i your-key.pem main.py comprehend.py index.html ec2-user@your-ec2-ip:~/

Step 4: Run the Application

Start the Flask application:

sudo python3 main.py --host=0.0.0.0 --port=5000

Once the application is running, open your browser and enter:

http://your-ec2-public-ip/

Expected Output

If a user enters 'I love this product,' the application will return a ‘POSITIVE’ sentiment.

Link to Demo Video

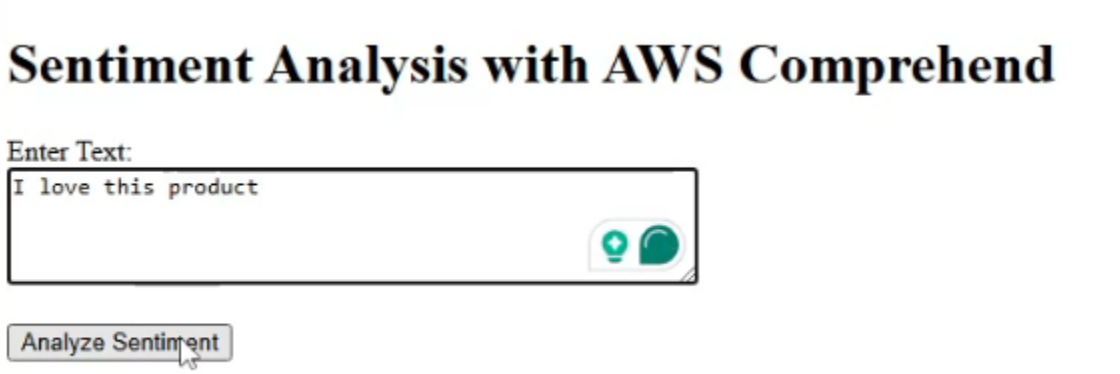
YouTube video of the working application: https://youtu.be/taQoIowwERg

How the Application Works

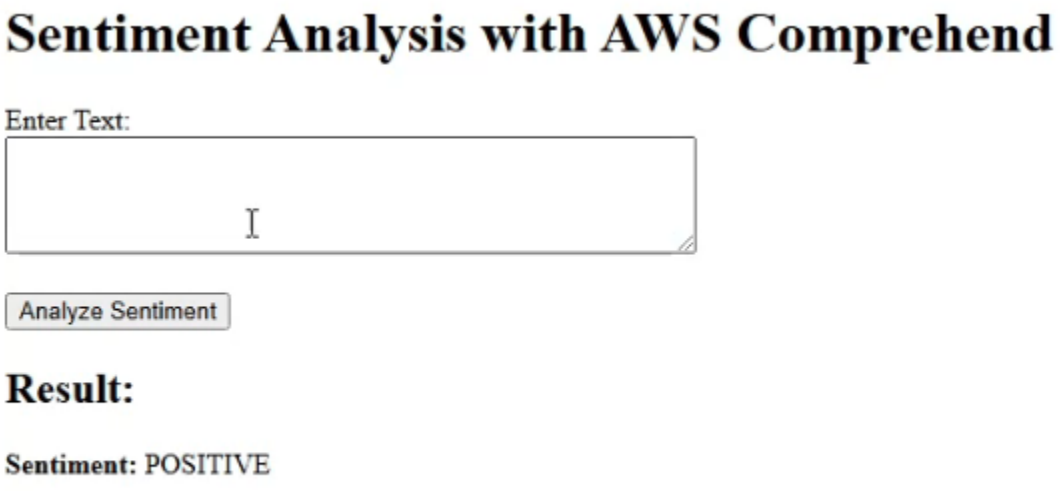
1. **User Input**:
    - The user enters text into the input text box on the web page.
2. **Processing**:
    - When the user clicks the submit button, the application sends the text to the backend.
    - The backend uses AWS AI services to analyze the sentiment of the text.
3. **Output Display**:

- The sentiment result is **only updated after the button is clicked and the analysis is complete**.
- The webpage refreshes to show the determined sentiment (e.g., **Positive, Negative, or Neutral**).

Screenshots of the Application

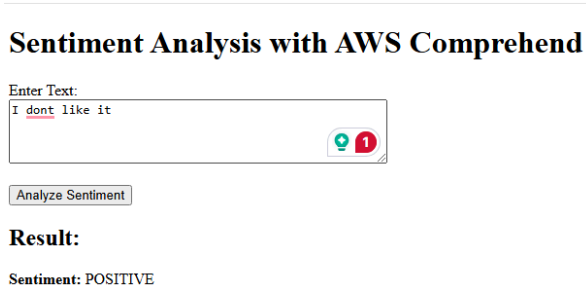


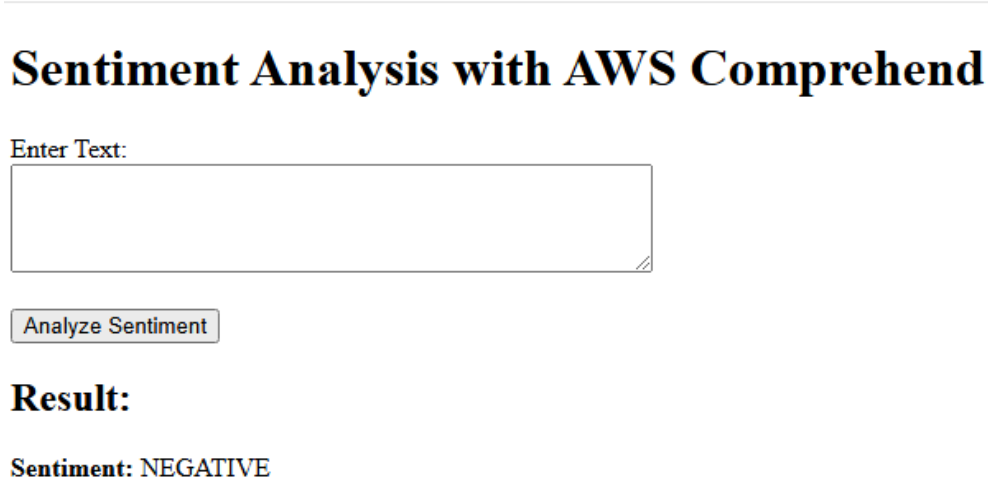
First, enter the text then click ‘Analyze Sentiment’. (Says Positive since the previous test was a positive comment and it won’t be updated until the button is clicked)



The text box will clear and the sentiment will update at the bottom.

The following are other examples:





Sentiment Analysis with AWS Comprehend

Enter Text:

it is soso

Analyze Sentiment

Result:

Sentiment: NEGATIVE

Sentiment Analysis with AWS Comprehend

Enter Text:

Analyze Sentiment

Result:

Sentiment: NEUTRAL

Sentiment Analysis with AWS Comprehend

Enter Text:

I love it

Analyze Sentiment

Result:

Sentiment: NEUTRAL

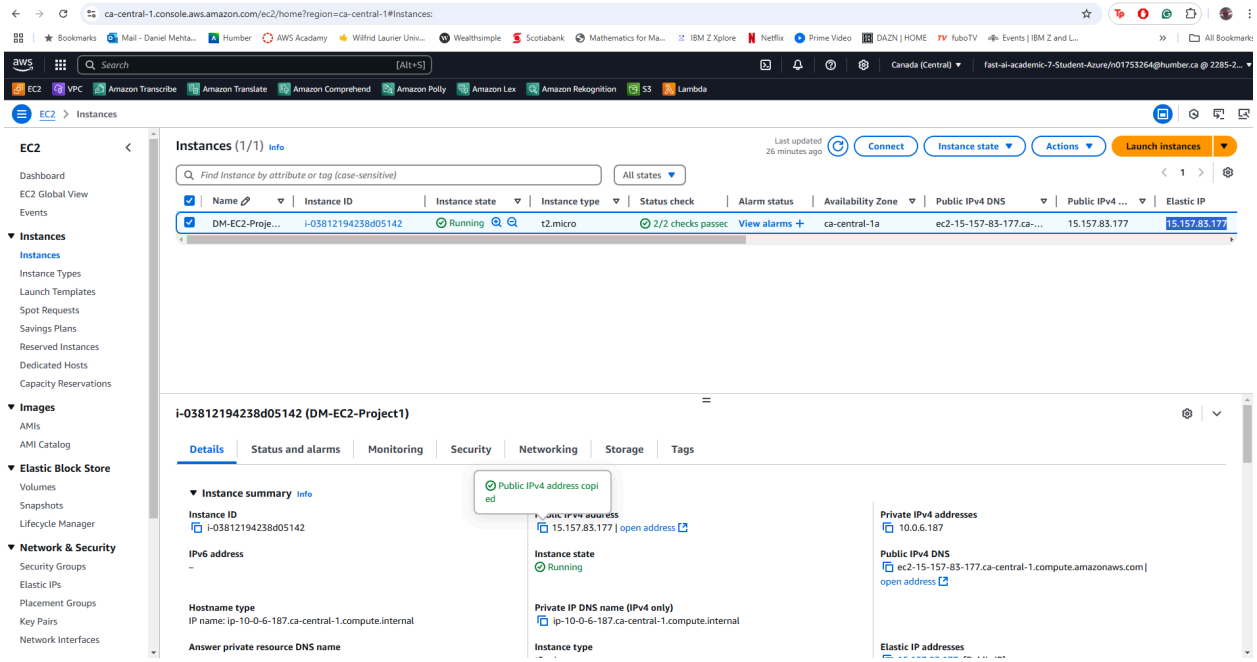
Sentiment Analysis with AWS Comprehend

Enter Text:

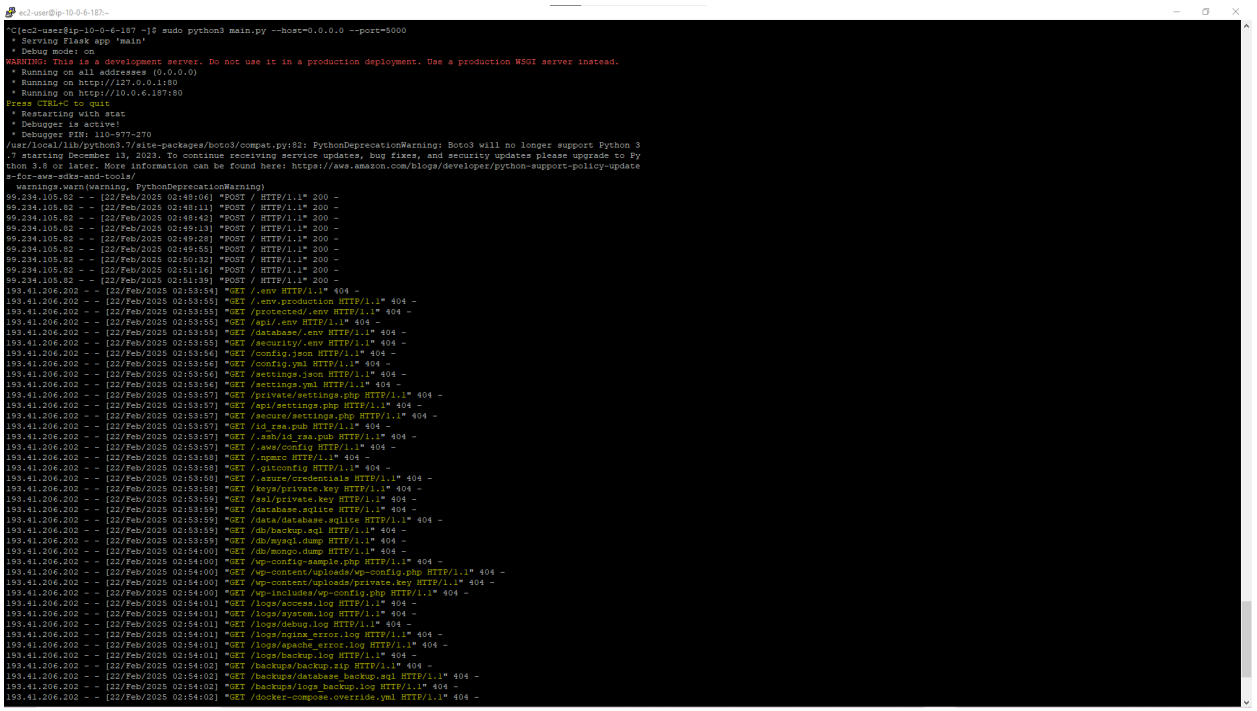
Analyze Sentiment

Result:

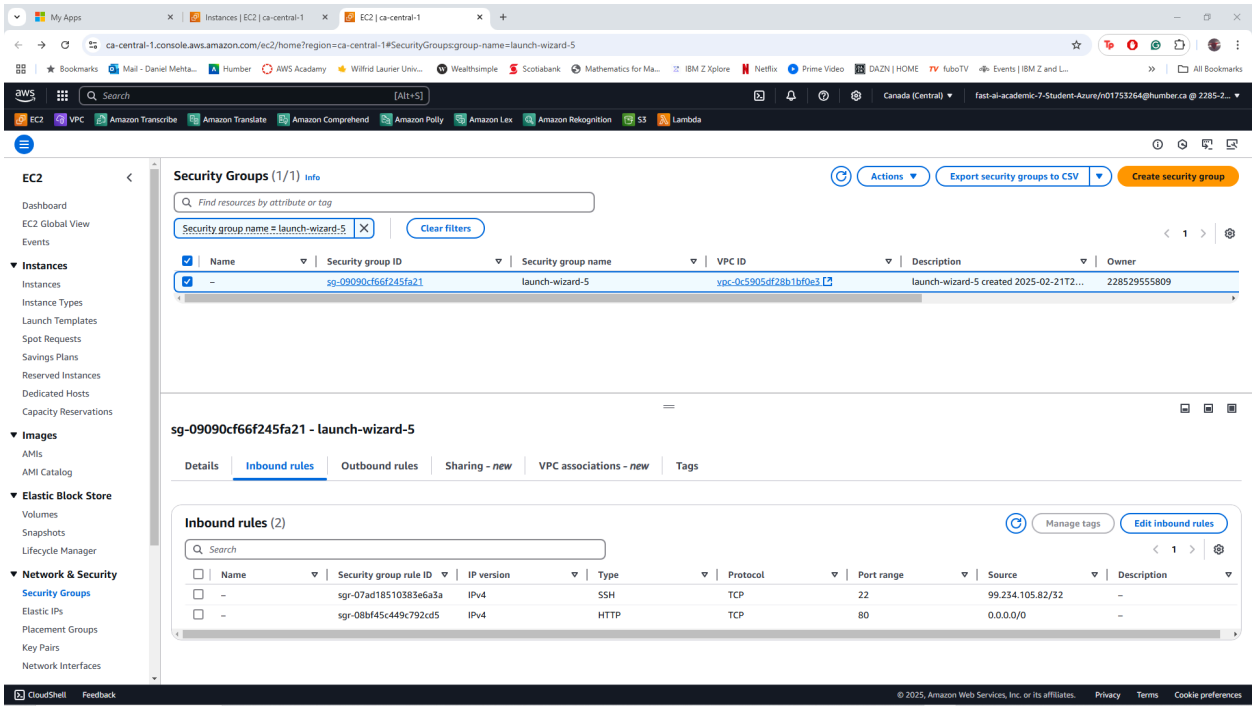
Sentiment: POSITIVE



The EC2 instance



The SSH while the application is running.



A security group rule was configured to allow inbound traffic on port 5000, ensuring external users can access the Flask application.