**FINAL PROJECT**

**Title:** Live twitter data analysis using Cloud services

**Team members:**

1. Dhanunjaya Lakshman, Koneru (Student ID: 16336828)



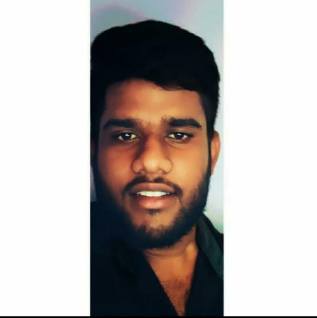
1. Lakshmi Durga Likhitha, Tellakula (Student ID: 16335481)



1. Sushmitha Reddy, Pollapalli (Student ID: 16336844)



1. Siddartha Rao, Damarla (Student ID: 16337125)



**Motivation and Purpose:**

The motivation behind the "Live Twitter Data Analysis using Cloud Services" project is to harness the power of cloud computing to extract, process, and analyze real-time data from Twitter. Twitter is one of the most popular social media platforms with millions of users tweeting every day. By leveraging cloud services, the project aims to provide a scalable, efficient, and cost-effective solution to analyze tweets and gain insights into trending topics, sentiment analysis, and social media behavior.

The purpose of this project is to create a pipeline that can extract tweets from Twitter, store them in an S3 bucket, perform data transformations and pre-processing using pandas in Lambda, generate visualizations using plotly, and store the results back in S3. This pipeline will be triggered using CloudWatch, which will monitor for incoming tweets and trigger the processing pipeline when new tweets are detected. An interactive dashboard will be hosted using Elastic Beanstalk (EBS), which will allow users to explore the data and gain insights. EC2 will be used to install additional libraries and dependencies that may be required for processing and analysis.

In summary, the primary objective of this project is to create a scalable, cost-effective, and efficient solution for real-time data analysis using cloud services. By leveraging the power of cloud computing, the project aims to extract valuable insights from Twitter data and enable users to make data-driven decisions.

**System Architecture:**

**A picture containing text, diagram, design

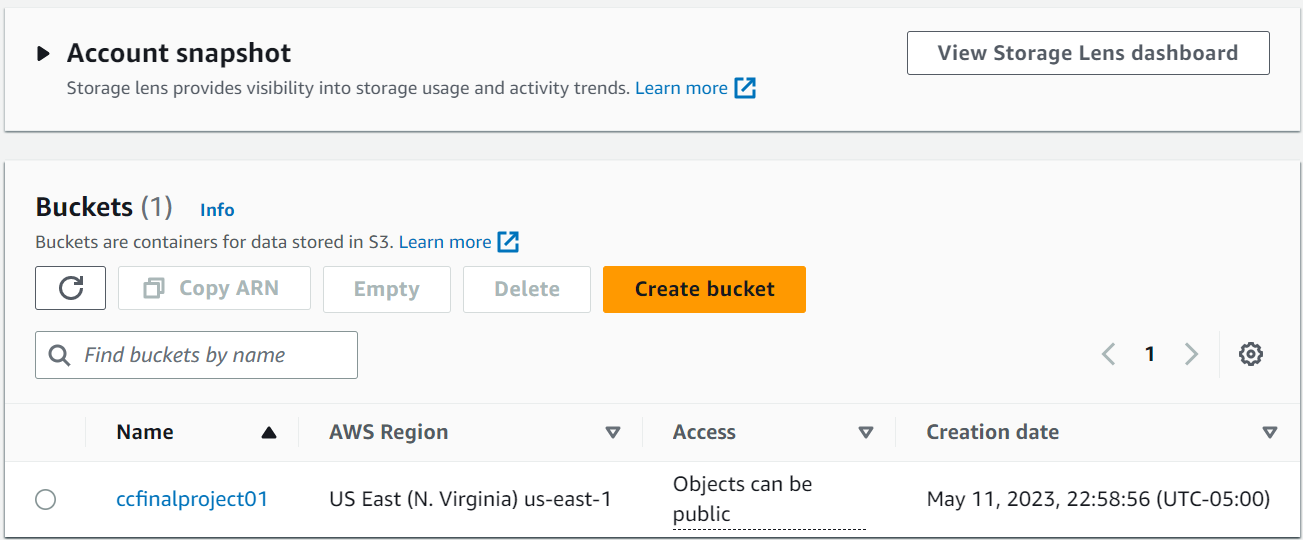
Description automatically generated**

**Features:**

1. The project aims to analyze real-time data from Twitter using cloud computing.
2. The project uses Amazon S3 for storing and retrieving data, AWS Lambda for data transformation and pre-processing, and Plotly for generating visualizations.
3. Amazon CloudWatch is used for monitoring resources and triggering the processing pipeline, while Amazon Elastic Beanstalk is used for hosting an interactive dashboard.
4. Amazon EC2 is used to install additional libraries and dependencies for processing and analysis.
5. The primary objective of the project is to create a scalable, cost-effective, and efficient solution for real-time data analysis.
6. The project leverages the power of cloud computing to extract valuable insights from Twitter data and enable users to make data-driven decisions.

**Steps:**

1. Created an S3 bucket for the storage of results in objects and made it public.



1. Installed all the libraries required for the project.

Graphical user interface, application

Description automatically generated

1. Created the Lambda Layer where we uploaded the packages as a zip folder so to use in the functions with the compatible runtime as Python 3.9 version.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

1. Created a Lambda Function **cars.**

Graphical user interface, text, application, email

Description automatically generated

1. Added the above created layer to the cars function.

Graphical user interface, text, application, email, Teams

Description automatically generated

1. Created access key to access the AWS services programmatically.

Graphical user interface, text, application, email

Description automatically generated

1. Created the twitter developer account and got the bearer token and connected the account with the tweepy package.

Graphical user interface, text

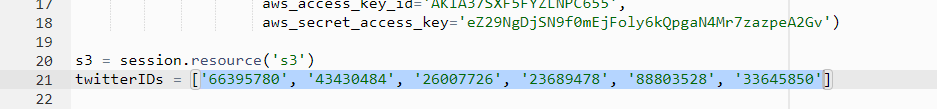
Description automatically generated

1. Created the session using the boto3 and the created access keys.

Text

Description automatically generated with low confidence

1. Got the twitter IDs of the car companies through the twitter developer account created.



1. We wrote the lambda\_handler function in which data is extracted from the car companies twitter accounts and activity of each company in the last 6 months. This data is presented visually in the bar charts where the data is stored in the S3 object, and the object is made publicly available. This whole process is automated and written in the lambda\_handler.

Graphical user interface, text, application

Description automatically generated

1. Creation of objects and updating them is automated.

Text

Description automatically generated

1. While testing the code we faced some errors where we rectified the errors and made it successful.

Graphical user interface, application, Word

Description automatically generated

1. As mentioned, the objects are created in the bucket where result is stored in JSON format and visualization in html file where it can be viewed.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. Storing the data visualization in the html object and making it publicly accessible URL.

Chart, bar chart

Description automatically generated

Chart, bar chart

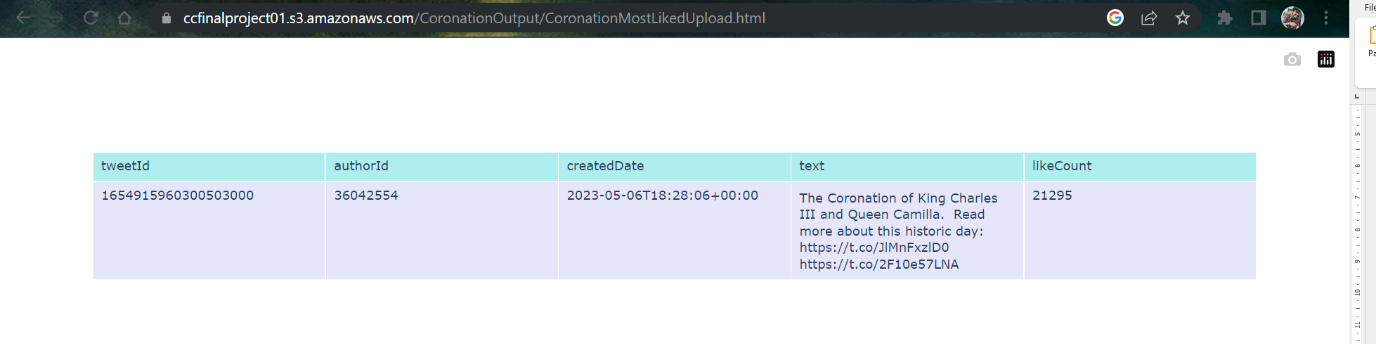
Description automatically generated

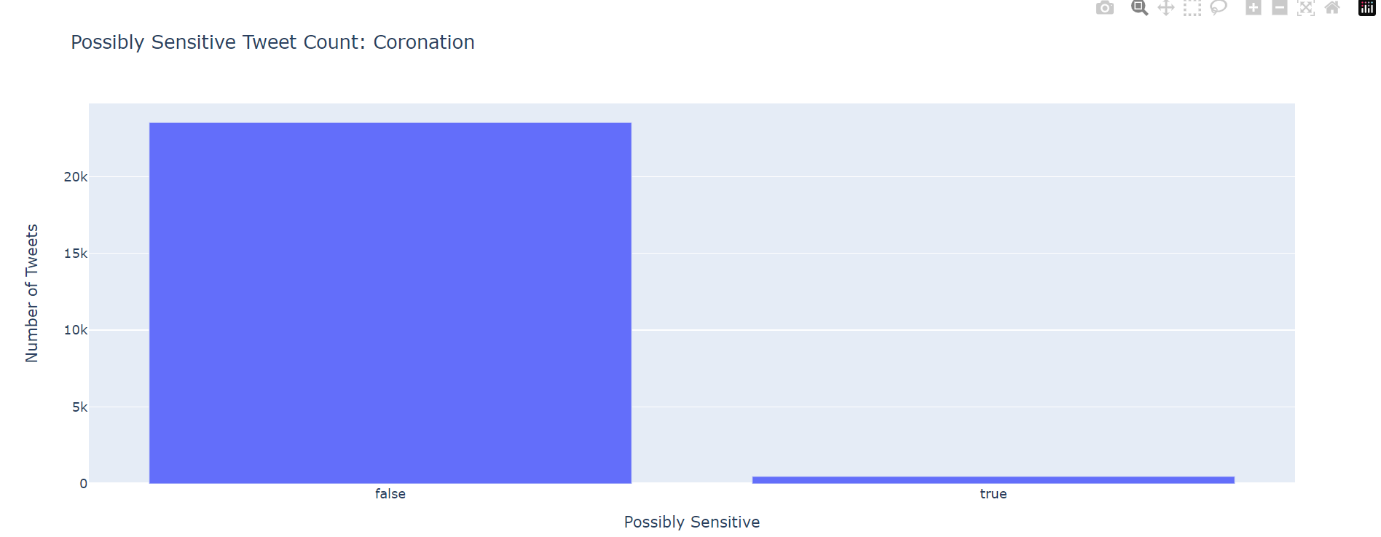
1. Like the same way, we have done visualization on SVB (Silicon Valley Bank) where we have taken coronation at some period at a particular day.

A screenshot of a computer

Description automatically generated with low confidence

1. The output of the SVB is shown below for the most liked tweets and possible sensitive tweets.

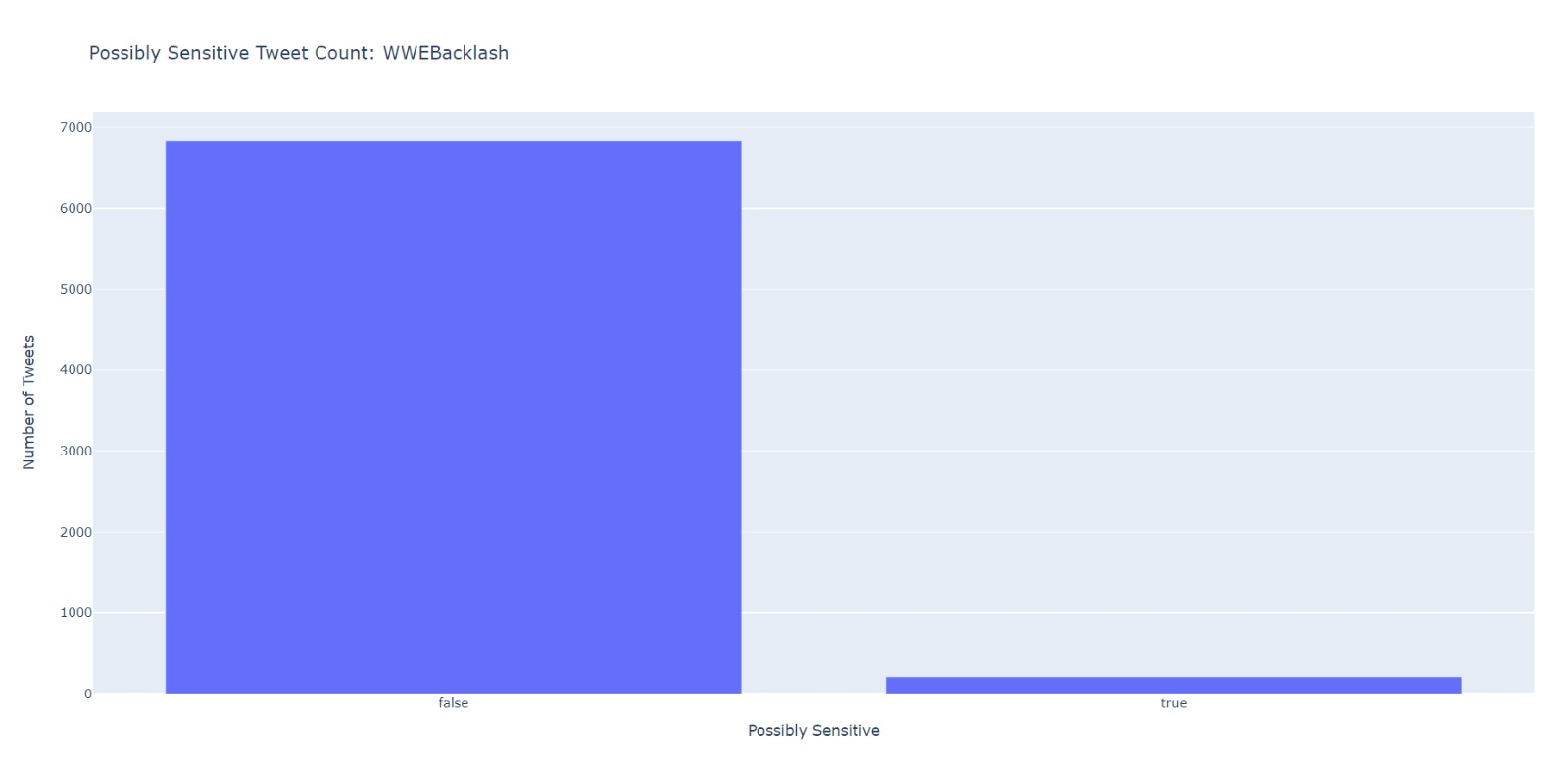




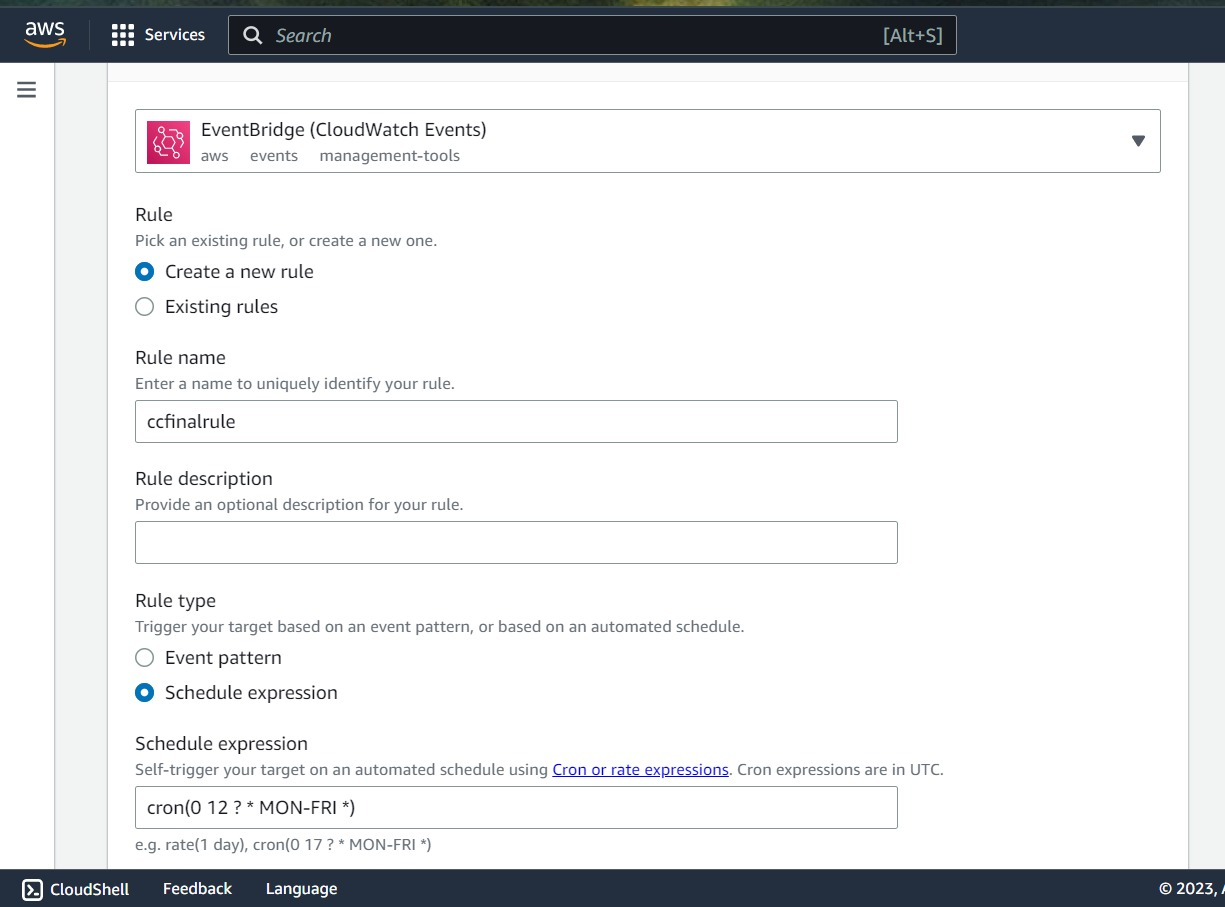
1. The same way we have extracted the data of the WWEBacklash and made the visualizations for the most liked and the possible sensitive tweets.

A close-up of a computer screen

Description automatically generated with low confidence



1. Using CLOUD WATCH Event Bridge to trigger the Car, Coronation and WWE Backlash application at a particular time on scheduled days in our case it is Monday to Friday at 12pm as below.



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

1. Created an EBS application with the aws-elasticbeanstalk-ec2-role for the visualization of the car companies with their features like speed, safety, comfort, and security. Written the python code for the above where it extracts the data from the CARData.json from the bucket and produces the visualization. In this visualization we have made access to exclude a company by clicking on the company name color box on the right to get more details.

A picture containing diagram, text, screenshot, design

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

A picture containing screenshot, diagram, circle, text

Description automatically generated

**Challenges:**

* Inability to use flume to ingest data.
* Limitations while using tweepy.
* Structure of ingested data.
* Data types of ingested data.
* Developing and hosting the dash application.
* Structuring the data for developing dash application.
* Installing third party libraries to use them in lambda.
* Deploying the dash application.

**Accessible Links:**

1. Object URL of car data: <https://ccfinalproject01.s3.amazonaws.com/CAROutput/CarCompanyActivity.html>
2. Object URL for coronation most liked tweets of SVB: <https://ccfinalproject01.s3.amazonaws.com/CoronationOutput/CoronationMostLikedUpload.html>
3. Object URL for publicly sensitive tweets from SVB: <https://ccfinalproject01.s3.amazonaws.com/CoronationOutput/PossiblySensitiveUpload.html>
4. Object URL for coronation most liked tweets of WWE Backlash: <https://ccfinalproject01.s3.amazonaws.com/WWEBacklashOutput/WWEBacklashMostLikedUpload.html>
5. Object URL for publicly sensitive tweets from WWE Backlash: <https://ccfinalproject01.s3.amazonaws.com/WWEBacklashOutput/PossiblySensitiveUpload.html>
6. URL for the Car features using the EBS: <http://ccfinalprojectcar-env.eba-6bjzn4tc.us-east-2.elasticbeanstalk.com/>

**GitHub Link:** [**https://github.com/DJ2803/CC-Project-Live-Twitter-Data-Analysis.git**](https://github.com/DJ2803/CC-Project-Live-Twitter-Data-Analysis.git)

**Video Link:** [**https://drive.google.com/file/d/1r2YN12F2gSThHbfh53tEnenTucSeKcby/view?usp=share\_link**](https://drive.google.com/file/d/1r2YN12F2gSThHbfh53tEnenTucSeKcby/view?usp=share_link)