Data Engineering Final Project

Politics & Stocks – Stocks and Politics Trends Near Real Time

Background

In today’s reality, we exposed to constant information stream from a multitude of different sources – Social network, digital news, billboards commercials, tv and so on.

Social network (“Wisdom of the crowd”), and digital news could give us many insights how to act wisely in different professional areas.

We want to focus on two types of characters, and two type of scenarios:

The investor, and the public figure.

The Investor: He want to track on real-time or near real-time for data from the social networks, and digital media to understand what people and financial reporters think (or believe) will happen about a specific company – Should he buy shares of this company, he wants to identify the moment to buy the shares before its prices go up or sell before its prices go down.

The political figure: He want to understand how the recent events affected the public opinion about him.

Current Situation or Problems We want to Solve

**The investor** – He needs to track manually in the social networks, search in relevant groups, and read relevant newspapers to get insight about a company.

**The public figure** – He probably will order a survey from a survey company. The survey company will call people or send digital forms to fill-out. Until the final analysis arrives from the survey company, in today's reality it may no longer be relevant.

Project Overview

The purpose of the project is to solve the mentioned above problems.

We want to provide prepare high-quality aggregated data with insights to assist the investor and the politician to make smart decisions which based on data but not by a ‘gut feeling’.

Requirements Overview

In this project we thought will be better to use serverless services in order to avoid local maintenance configurations.

Project Solution Parts

1. Get input from the user.
2. Get data from twitter and digital news which related to the user input.
3. Cleansing and enriching data.
4. Analyze the data and present insights in dashboard.

**Project Tech Components**

**Overview**

In this project, the main concept to choose tools and services was “Value for Money”.

We might find more powerful or better services than the below, since we are founding our own project.

**Cloud Platforms and Cloud Services**

**AWS**

We decide to work with AWS services since it is a popular cloud platform in startups and Hitech companies which this is the industry we are working or want to work in the future.

Also, the AWS services were choosing over other services since it easier and more convenient to work in AWS platform with the AWS services.



**S3** – To store all the data files from twitter and news, also functioning as NoSQL DB. In addition, the main advantage in S3 is the ease of analyzing the data that store in S3 with AWS tools.

**Kinesis** – Functioning as our data streamer. We chose Kinesis over Kafka to learn another streaming AWS tool (we learnt Kafka in the course). Also, as mentioned above, this is a SaaS.

**Glue (Crawler component)** – For ETLs to convert the data from S3 (a non-relational structure( to DB.

**Athena –** To query the data in s3 and present it as a dashboard via the QuichSight.

**DynamoDB** – AWS NoSQL DB. Any exception in the pipeline is sent as a record to this DB (errors log).

**QuickSight** – For BI dashboard to present the analyzed data.

**AWS Lambda** – Lambda is an event-driven, serverless computing platform that can runs code in response to events in AWS (i.e., new file in S3).

**GCP**

Only as a machine to run scripts and trigger services.

In our GCP machine we already have properly installed relevant packages and libraries which save time to install and configure it all in a new machine in EC2 (AWS machine).

**Tech components – Frameworks, Libraries, Platforms**

**APIs**

API Twitter - Tweepy

To collect tweets from twitter to analyze them later.

We chose twitter as the social network over Facebook for two main reasons:

1. twitter provide convenient API to developers to collect tweets data.
2. Tweets are limited to 280 characters (as today). Analyze data from Facebook posts could be more complicated since posts are limited to **63,206** characters.
3. Analyze long posts from Facebook demands much more complexity and ML technologies which we want to avoid. In this project we are focusing more on Data Engineering technologies.

Tweepy is a popular python package library, and well-documented to use the twitter API.

Because of the above reasons we decided to choose this library. For almost any issue we encountered we could find a solution in StackOverflow.

Digital News - newsdata.io

An API which provides data from digital news website such ‘The Sun’, ‘National Post’ and ‘Yahoo News!’.

We selected this service since his convenient pricing and his ease of use.

Boto3

Allow us to trigger AWS services from a python script - store file to S3, read files from S3, open Kinesis streaming channel, use AWS sentiment analysis.

AWS Sentiment Analysis

Amazon service which analyzes tweets text and titles from the digital news – The service provide indication about the text – is it negative or positive, and score – how much it is negative or positive.

Google Sentimental Analysis

Google service which works like the AWS Sentiment Analysis, but we preferred to get another “opinion” to improve the insights results.

**Frameworks, Libraries, Platform**

Flask

A web framework that lets you as a web developer to use Python as the backend.

We used Flask to build an HTML page with JS to get the end-user input.

Panadas

Pandas is a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool.

We use pandas object to prepare and refine the data we collected from our sources (twitter and news.io).

PySpark

PySpark is the Python API for Apache Spark, an open source, distributed computing framework and set of libraries for real-time, large-scale data processing.

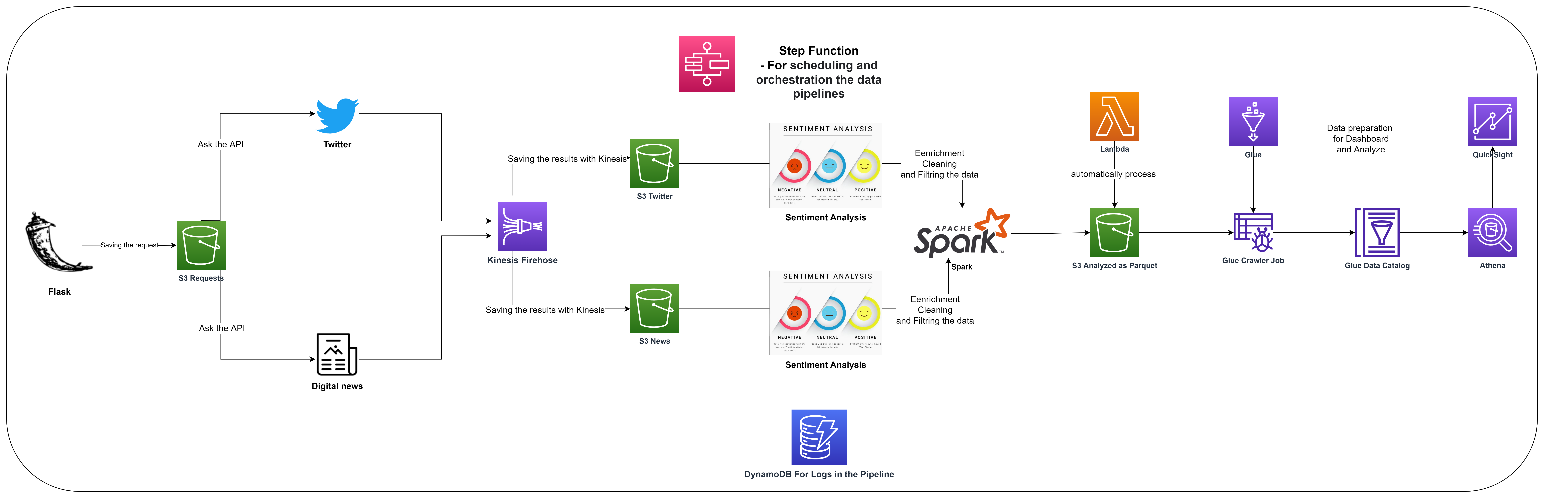
We use PySpark to handle with large scale of data.

Airflow

Workflow management platform. Since the project is one stream pipe, we use Airflow only to run the Flask when machine is started.

**Solution**

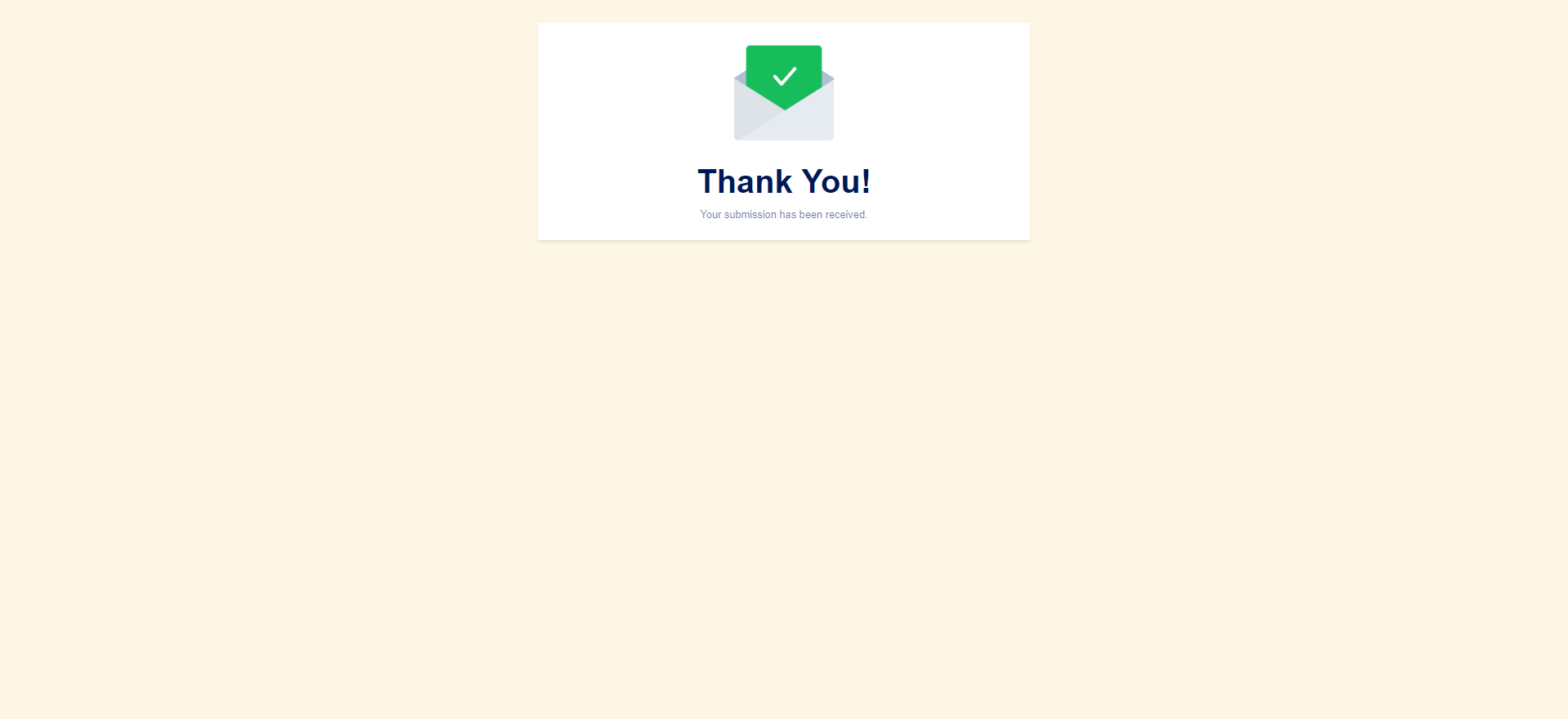
Data Pipeline



Get End-User Input Data (Flask)

We used Flask to get the user inputs.

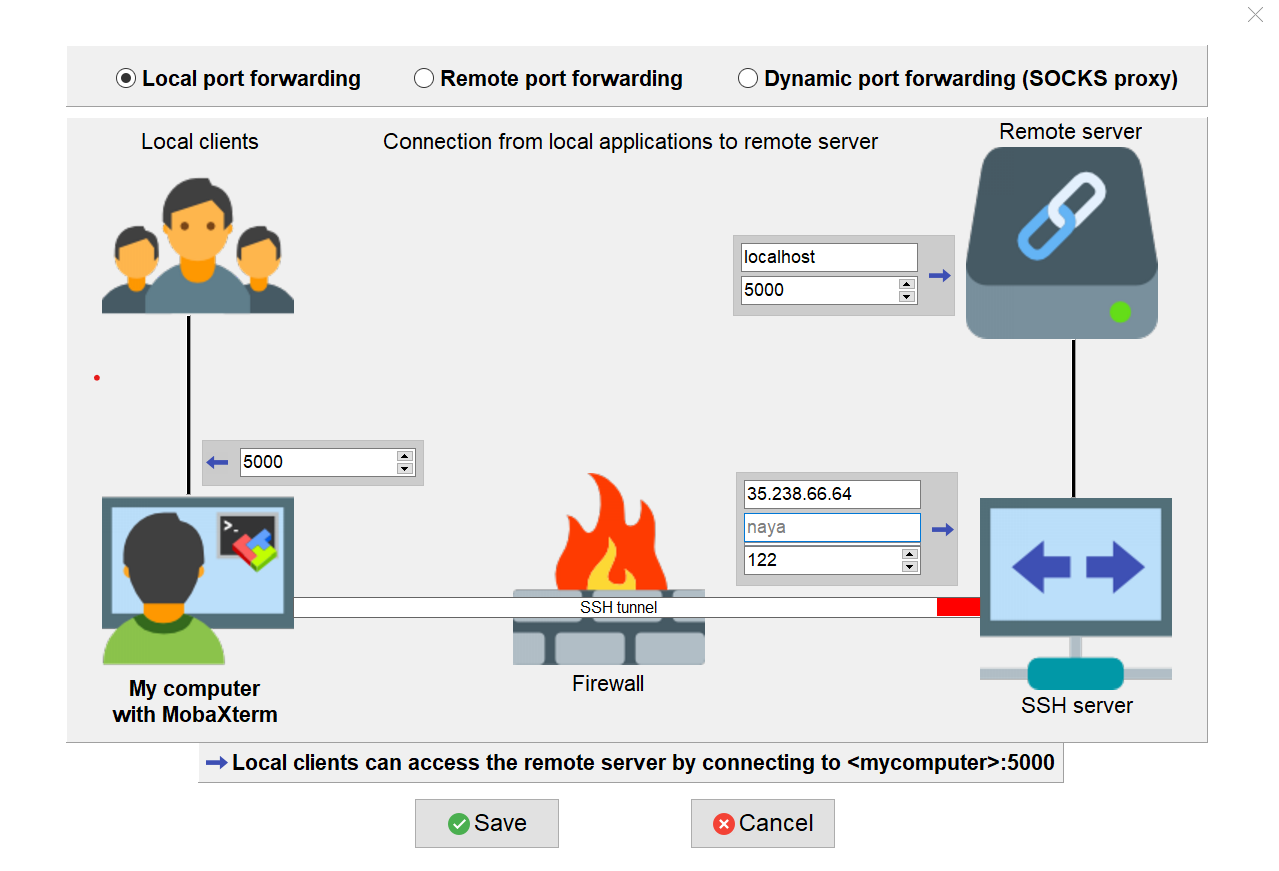




The user fills out keywords to search, from which date he wants to search, the category of the search (politics or stocks).

The fields ‘To Date’ and ‘Country’ will not use us in this project but just to present more ‘realistic’ form and for POC, how can we develop the process/project in the future.

Configure tunneling to load the web server from the GCP (remote machine)



Key Request

We are creating a unique key for the request. The key request is concatenated string of the following user inputs: keywords to search, from date, category, and country.

This key request will use us all along the flow of the data pipeline.

Store Request as a Json File in S3

The user input is saved in S3 as a json file for failover and backups in a specific bucket.

Later on, the process will take the relevant data from the request file to the next steps in our data pipeline.

For this step we used direct put with the boto3 to store the data.



Figure 3 main script - boto3 - upload to s3

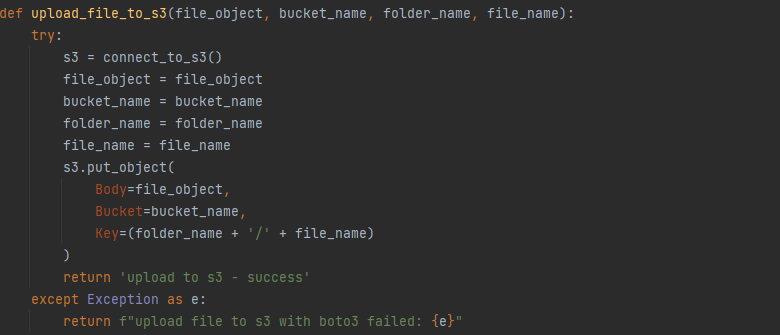


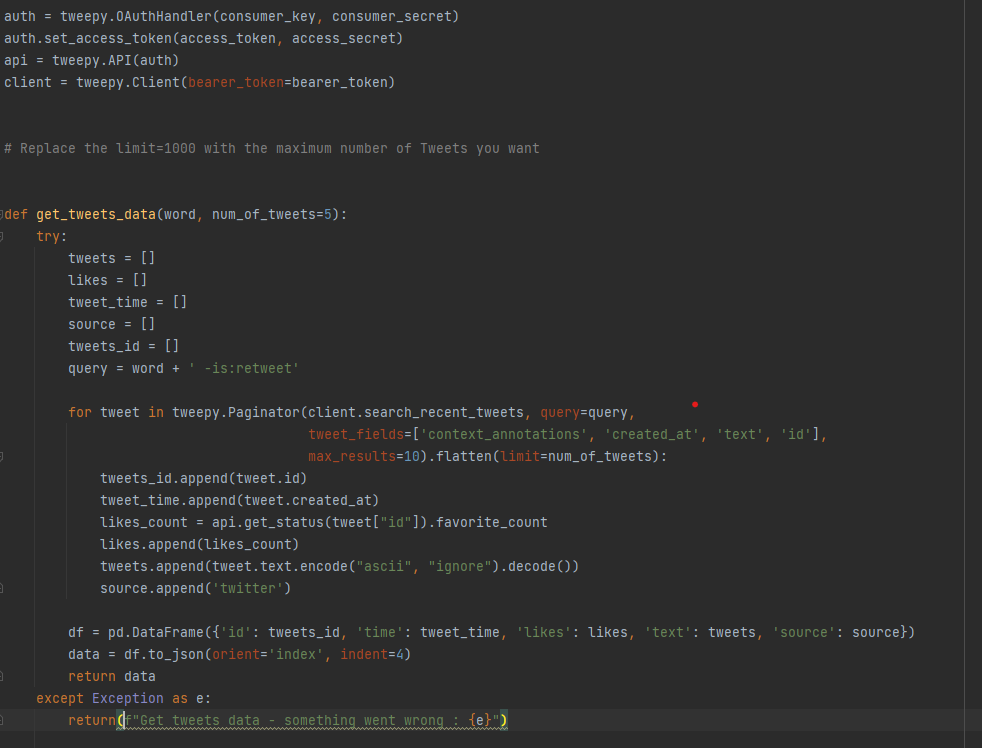
Figure 4 - Function to upload to S3 with Boto3

Collect Relevant Data

Twitter

Take the user input keyword and search them in Twitter.

The data will be saved in S3 Bucket ‘twitter-data-api’ under a dynamic folder as will be described below



newsData.IO

Take the user input keyword and From Date and search them in the API news.

The data will be saved in S3 Bucket ‘news-data-api’ under a dynamic folder as will be described below.

For this project, the relevant text that analyzed is only the title of the article but not the whole article. Analyze a full article demands much more complexity and ML technologies which we want to focus less in Data Engineering’s project.

Save the Twitter and News Results in S3 With Kinesis

All the above data are saved as json files in S3 with Kinesis Firehose.

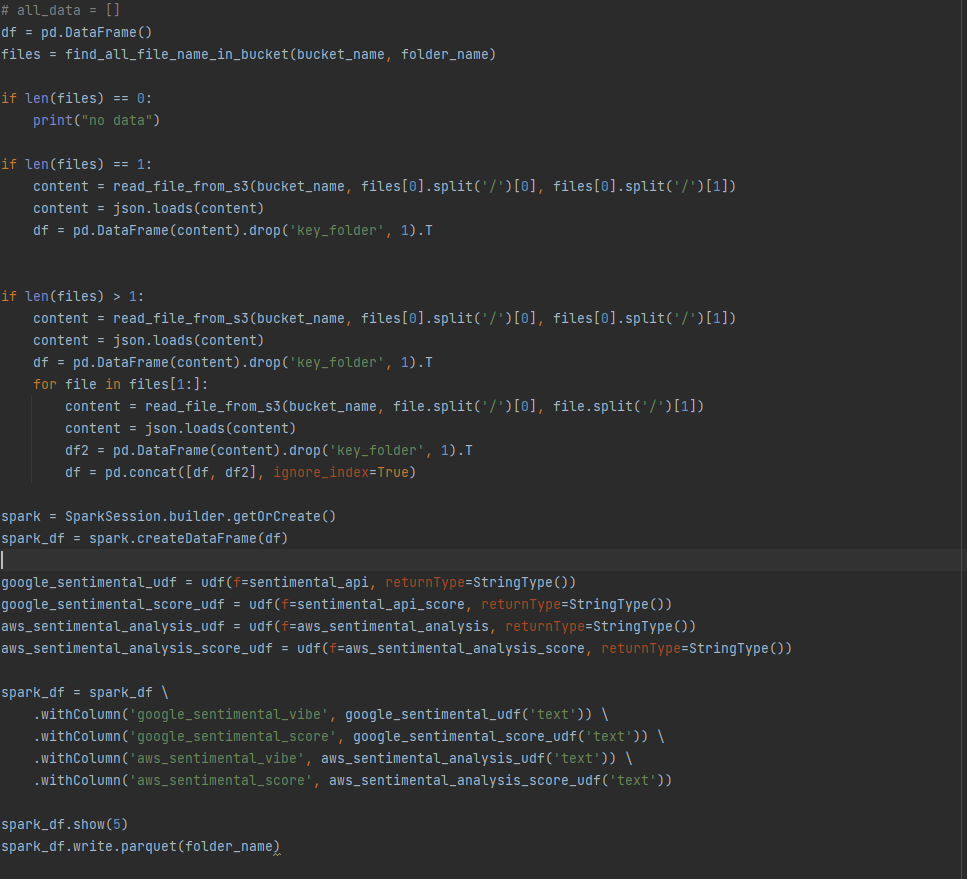
The firehose waiting for the data, create an S3 folder under the relevant bucket (‘twitter-data-api’ or ‘news-data-api’) with a dynamic name which related to the key request.

The next steps will be to enrich the data with sentimental analysis (google and aws) by using Spark

Spark

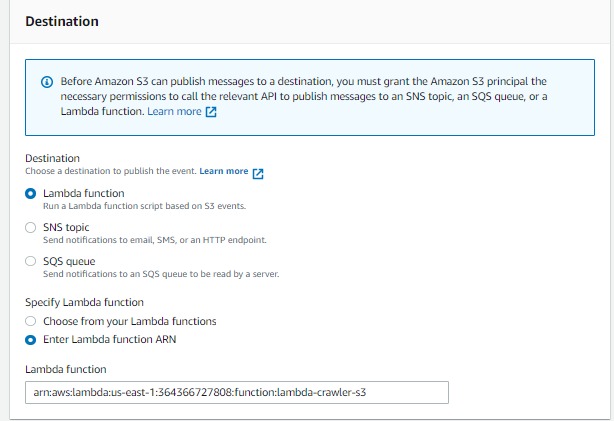
Collect the data from the above files with SparkDF, create udfs (external customize functions) which send the texts to sentimental analysis API, remove duplicated tweets and news title, and convert it to Spark DataFrame object.

In the end of this process the data will be delivered to a specific bucket in S3 (name: “Reports”).

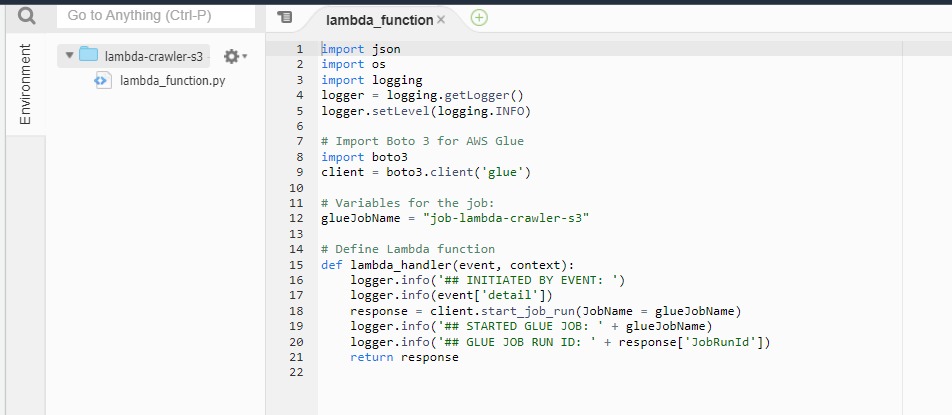


Event Notification (S3) + Lambda

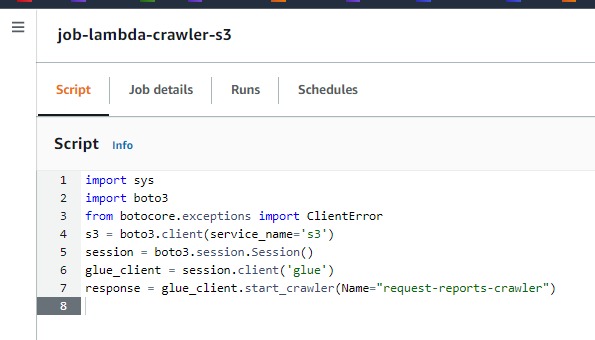
We implemented an event notification on the Reports S3 bucket. This event trigger a Lambda function every time it identifies a change in this bucket.





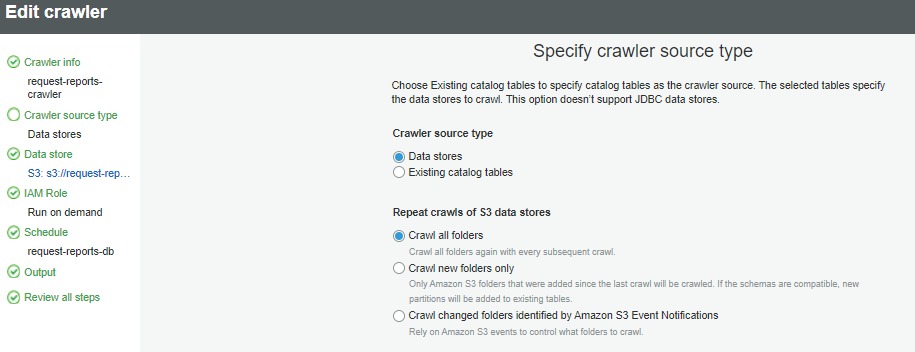


The lambda initiates a job Glue crawler.



Glue Crawler + Athena

After the Glue components initiated by the above lambda, it is identifying the deltas, and update the table which analyzed by Athena.



QuickSight

We are creating a dataset on the S3 bucket with the Athena to build a dashboard in the QuickSight.