SentiStock+

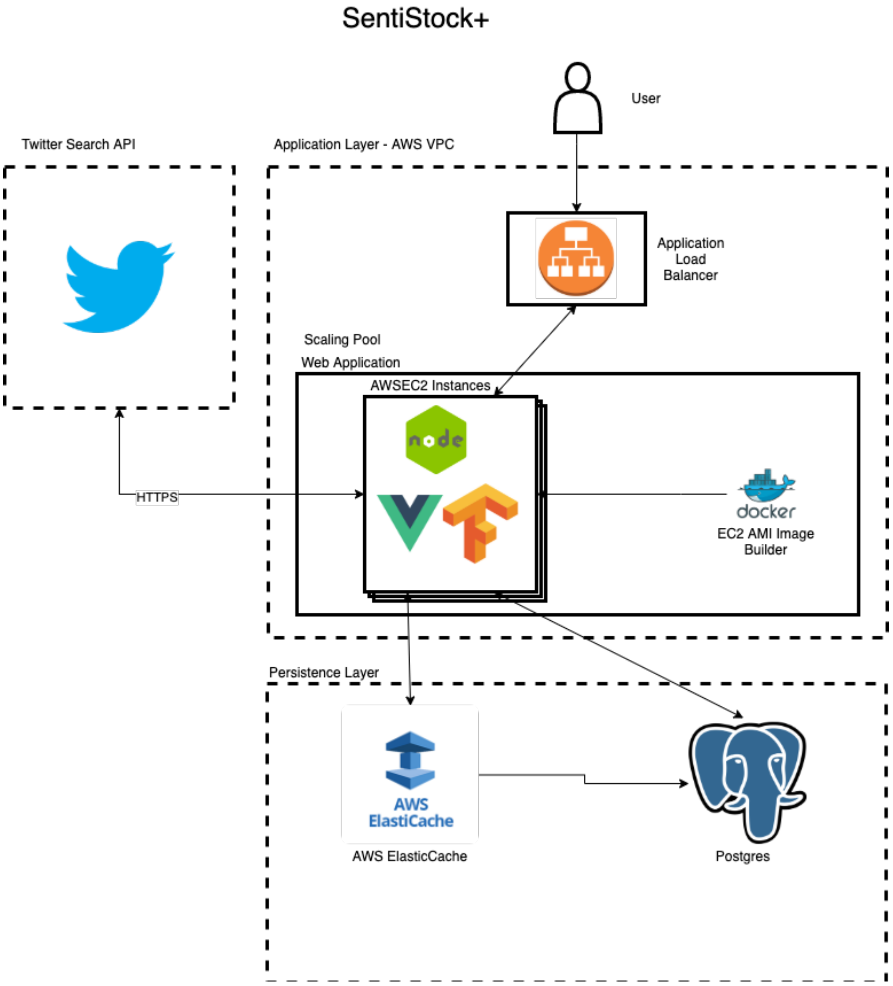
Assignment 2 Proposal

Purpose & Description

The purpose of this application is for the user to be able to quickly gauge the consensus of a certain stock within a public forum. This is beneficial to the user as it is an easy way to aggregate a lot of data into one user friendly interface. In addition, the user should be able to view the price of the stock within the application and look at the chart of a stock. They should also be able to see the current tweets about the stock.

The application should have an intuitive interface, and it should be modern in its appeal. In addition, the user should not have troubles navigating around the application. All of the information should be there. This includes the charting tool, the current tweets on the stock, the lookup bar for the stock and the current sentiment about the stock. In addition, they should be able to see their past histories of stocks which they have looked up before.

Architectural diagram



Persistence

- For this application, the persistence will be in the form of a cache + SQL service. This application will use these services to perform the following
    - Store previous search queries
    - Store previous analysis on search queries
    - Store previous data obtain from twitter for a certain stock ticker
    - Store the user's information regarding login

Scalability & Load

- For this application, the scalability will come in the form of the scaling pool. This scaling pool will be scaled using a CPU usage policy. As the number of tweets (data) is request from the Twitter API, and hence processed by the server using the tensorflow.js library, so too will the load on the server, and hence the CPU usage.

Phases of implementation

1. Develop the application locally (SQL database, redis caching, server, client)
2. Deploy the application to a container via docker, and create an AMI
3. Run the application in the cloud with one instance, test a moderate load (50 tweets)
4. Create a scaling pool, and scaling policy
5. Test the application using multiple requests with a high amount of tweets

Use Cases

1. As a user, I want the system to display the sentiment of a stock ticker, so that I can gauge the sentiment of a stock in a public forum
2. As a user, I want the system to be able to track my previous stock ticker searches, so that I can look back at past searches
3. As a user, I want the system to be able to compare my current stock ticker search to a previous stock ticker search, so that I can see the difference in sentiment across different time frames
4. As a user, I want the system to display analytics on the tweets which are used to give me sentiment on a stock, so that I can make informed decisions on the stock
5. As a user, I want to be able to login to my account, so that I can view all of my previous data and information

Service & Data APIs

1. Twitter Search API

Description: This will be used to fetch the data required for sentiment analysis on each of the stock tickers. This can be required as the API which will obtain the data for the application. The load of the application will change with the amount of data requested from this API.

2. TradingView

Description: This API is used to validate the stock tickers which are entered on the loading screen. Therefore, the user will not be able to enter any invalid tickers on the loading screen.

Client

Tech Stack

- Quasar → Frontend component library based on the material UI standard
- Vue → frontend framework
- Chart.js → Charting library used to perform analytics

The client will be responsible for rendering the charting for the analytics of the tweets, as well as the sentiment of the tweets. In addition, it will perform some validation on the input fields for the query parameters before sending it to the server.

Server

Tech Stack

- Node → Backend javascript runtime environment
- Express → Backend web application framework
- Swagger → Documentation for the APIs
- Tensorflow.js → Sentiment analysis library for natural language processing
- Elasticache → Managed caching service from AWS
- Postgres → SQL database for persistence

The server will receive data from the Twitter search API based on the user's query parameters. This data will then be used by the server to perform sentiment analysis. The tensorflow.js library will be used to perform this analysis. This analysis, as well as the data will be persisted into an SQL database, and will be stored in the cache to be used at a later date from the user.