Assignment 2 – Project proposal

“Twitter Enalyst”

|  |
| --- |
| Tutor feedback:   * Are you using the Twitter search endpoint or the streaming endpoint or both to retrieve the tweets? This decision will impact your architecture choices * What is your proposed scaling policy? * Just doing sentimental analysis may not generate enough load. This is something that you will need to explore |

1. Description

Twitter Enalyst is an app helping user to quickly look up the trending keywords on the Internet and analyze the sentiment level of a keyword that users choose.

2. Architecture diagram

Diagram, text

Description automatically generated

3. Phases of implementation

App features -> Storage (Database + Redis) -> scaling

4. Persitence choices, scaling metrics, application threshold

- Scaling policy: Network trafficking / cpu usage

- Using EC2 as VMs to host the app, Redis is utilized to store short-term data

- DynamoDB: stores queried results (twitter posts). Serve 2 purposes:

- Calculate the top10 trending searched word on the app in the last 7 days (update daily, ec2 will query from dynomoDB and calculate). DynamoDB needs Keyword, timestamp,

- Use existing data to display on chart (summary analysis of the last 100 posts if they are less than 1h old – write to DynamoDB with a timestamp)

**User searches ‘dog’**

->

stream: stream posts

search: 100 posts summary,

trending: server/DynamoDB calculate the trending keyword and send to client

->

server: get the keyword, timestamp (rodo will label timestamp himself) from both api and 100 posts summary from search

**Next user searches ‘dog’**

->

stream: stream posts as normal (don’t care if there’s no new posts)

search: get from database if the key word is searched less than 1h ago or get directly from search api otherwise

trending: same as above

->

server: get the keyword, timestamp (rodo will label timestamp himself) from both api and 100 posts summary from search if query from search api

Search api: Keyword, summary (of newest 100 posts), timestamp

Stream api: Key word, timestamp

5. Use cases

User story 1

|  |  |
| --- | --- |
| As a | User |
| I want | see what keywords are trending right now in Australia or worldwide on google search engine |
| So that | I can catch up with the trending news/events |

User story 2

|  |  |
| --- | --- |
| As a | User |
| I want | To search for a keyword of my choice |
| So that | The app can return the sentiment score of posts containing that keyword |

User story 3

|  |  |
| --- | --- |
| As a | User |
| I want | The app to display the sentiment score of **newly-posted** twitter posts from that moment visually in a chart |
| So that | I can view the sentiment scores more intuitively |

User story 3

|  |  |
| --- | --- |
| As a | User |
| I want | The app to display the **summary** sentiment score of **newly-posted** twitter post visually in a chart |
| So that | I can view the overall sentiment scores of the keyword |

User story 4

|  |  |
| --- | --- |
| As a | User |
| I want | The app to display the sentiment score of the keyword **from the last 7 days/ 24h** in a chart |
| So that | I can view the overall sentiment scores from the past as well |

6. APIs

Google Trends API – User story 1

<https://www.npmjs.com/package/google-trends-api>

Twitter API – User story 2

<https://developer.twitter.com/en/docs/twitter-api/early-access>

Sentiment – User story 3

<https://www.npmjs.com/package/sentiment>