Proposal CAB432 assignment-2

Student 1: Daniel Sørenes n10572465

Student 2: Anshul Jangra n10080431

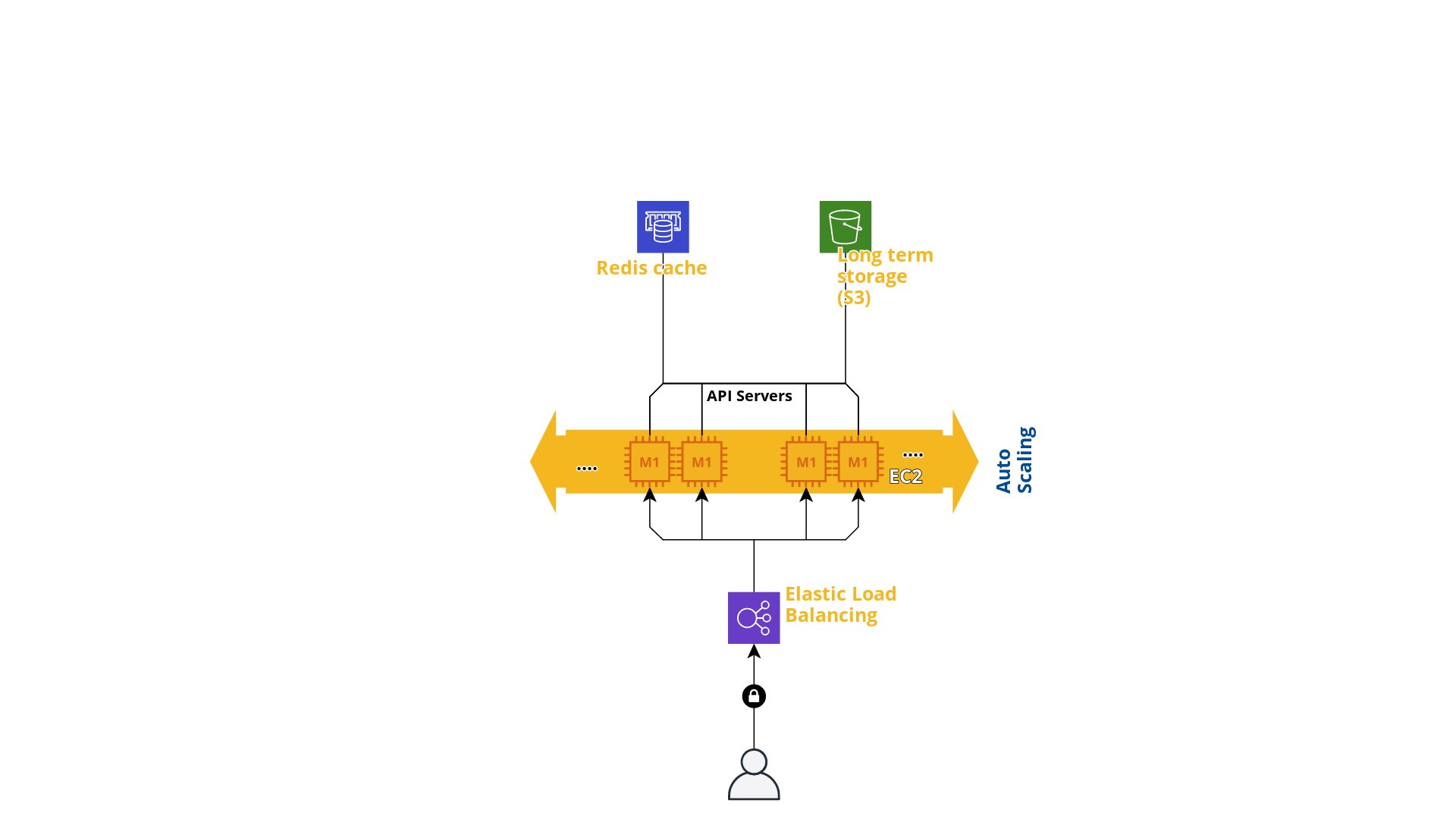
# Application purpose and description

The application takes in user queries (specified hashtags) for Twitter tweets and shows relevant information pertaining to those tweets. Such information is most important/relevant words in that context, most used hashtags and sentiment analysis. This will be displayed with D3.js and wordclouds, bar-graphs and the other visualization methods.

# Phases of implementation

1. Front-end for basic queries.
2. Back-end for basic query functionality.
3. Back-end for twitter analysis, such as sentiment analysis, word importance, frequency of words and hashtags.
4. Persistence implementation with Redis cache (ElastiCache).
5. Persistence implementation with S3
6. Implement load-balancing and scaling

* S3 for the long-term storage
* ElastiCache (redis) for cache storage
* EC2 M1 machines for the instances
* ELS (Elastic load-balancing, internet-facing) for the load balancing



# Use cases

1. I want to see what the context of the tweets are with these specific hashtags
2. I want to see which hashtags are the most popular one based on these hashtags
3. I want to see which words are most associated with these specific hashtags.
4. I want to see if the overall representation of these hashtags is negative or positive.

# Services and APIs

* Twitter API for basic search and getting the tweets
  + Fetches tweets based on a range of possible query parameters. Here the only query parameter used is hashtag specific.
  + Used for every user story to get the tweets containing those specific hashtags
* NLP
  + Used for user-story 1, 3 and 4. Used for sentiment analysis and tf-idf feature extraction (get the words with most importance/relevance to the context of the tweets).
* D3js for visualization
  + Used for every user-story to visualize the data from the analysis.
* Twitter Bootstrap
* Natural-compromise for data pre-processing
  + For pre-processing the tweets to make it usable for NLP analysis.
* ElastiCache for cache
  + For caching the tweets/tweet analysis.
* S3 for long-term storage
  + For long-term storage for the tweets/tweet analysis.
* Node.js for backend
  + For the server and tweet processing.
* React for frontend
  + For the user-interface and integration with D3js.

# The division between client-side and server-side

A clear statement of the division between server side and client-side processing, and the technologies to be deployed.

The server will get a request from the client and handle the user query with the Twitter API. It will the parse and analyze the tweets from the Twitter API response, store and cache that data, and in the end send the outcome back to the client. The server is concerned about getting and creating the data. Basically, WHAT to send to the client and how to send that data in a relevant format.

For the server-side the technologies to be deployed are:

* Natural-compromise
* NLP
* Node.js
* S3
* ElastiCache
* REST

The client is only responsible for HOW to display the data it receives from the server to the user.

Technologies to be deployed on client-side:

* React
* D3.js
* Twitter bootstrap