

CAB432 Cloud Computing

Assignment 2 - Cloud Project (Focusing on Scalability)

Overall Application Purpose and Description

This application is a cloud-based query processor and sentiment analyser based on live tweets. Multiple hashtags can be submitted to the application and will become a streaming filter against incoming Twitter messages. Any tweet that passes through the filter will be analysed and the result will be displayed on the screen, including the filtered messages.

Architecture Diagram and Proposed Phases of Implementation

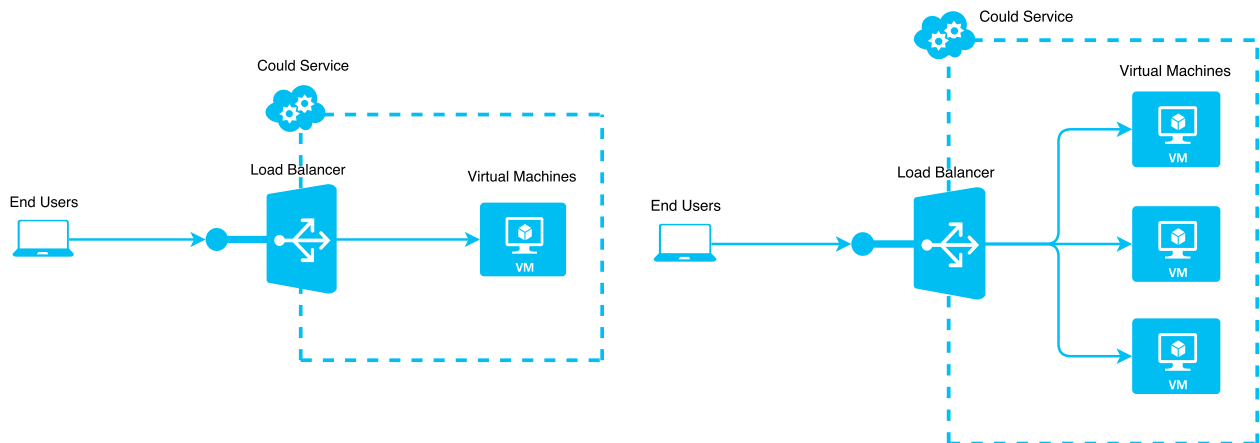


Figure 1: The architecture diagrams (before and after scaling).

The application will be implemented on top of the Microsoft Azure VM (IaaS Cloud). The Azure load balancer will be used as a service to randomly distribute the load of web request traffic across multiple web roles or instances when automated scaling is needed.

In case of deployment, the application will be deployed via a Docker container on top of Ubuntu Server VM on Azure.

Example Use Cases

This application can be used to detect emotions, evaluations and opinions in the content that people tweet specified by a collection of hashtags. It can qualitatively investigate people's thoughts and opinions on a particular subject using quantitative scale.

In terms of real usage, streaming inflow Twitter messages filtered by a set of hashtags can be used to get opinions or feedback on a particular topic. For instance, live Twitter streaming at any event is a good way to collect feedback and questions. By referring back to the collected data and performing sentiment analysis, it can give some useful insides of what should be improved and what are particularly appreciated.

List of Service and Data APIs to be utilised

- Twitter Streaming API - This API allows developers to access the global stream of tweet data. This application will use the public streams end point, which streams the public data flowing through Twitter. The service will be implemented on the server side using a Twitter client library (twitter, ntwitter or twit).

Server Side and Client Side Processing

Server Side

- Node.js - Node.js will be used to build the server side components.
- Express - Express will be used as a web application framework for Node.js.
- Socket.io - Real-time bidirectional event-based communication between server and client.

Client Side

- JavaScript - Client side scripting language
- HTML + CSS - Twitter Bootstrap will be the default web page layout of the application.
- AngularJS - A JavaScript framework for single-page web applications.

Essential Modules

- Sentiment analysis modules - One of the following modules will be used as a natural language processor (NLP). These modules need to be tested first in terms of its capability to analyse data (others are provided as alternatives).
 - Natural - A natural language processing library for Node.js. It can be used for tokenising, stemming, classification, phonetics, tf-idf, WordNet, string similarity and inflections.
 - Sentiment - This module uses the AFINN-111 word list to perform sentiment analysis.
- Twitter - The following modules are almost identical in terms of usage. One of these modules will be used to stream inflow Twitter messages (others are provided as alternatives).
 - twitter - An asynchronous client library for the Twitter REST and Streaming APIs.
 - ntwitter - An improved version of the node-twitter module, it is also a library for Twitter REST and Streaming APIs.
 - twit - A Twitter API client which supports both the REST and Streaming APIs.

Data Storage

- MongoDB (Mongoose module) - A NoSQL database, it stores data as JSON-like documents with dynamic schemas, which is called the format BSON. This will be used to store the result of sentiment analysis of a collection of hashtags.

Visualisation

- d3js - This library will be used to produce interactive data visualisation in the application using the result from sentiment analysis.