CS321 Team Big Data Proposal

A group project proposal

Connor Baker

2019-08-25, Compiled on August 25, 2019 at 11:43pm

Contents

1 Background **2**

2 Proposal **2**

2.1 Offering 2

2.2 Proposed Technology Stack 3

2.2.1 NiFi 4

2.2.2 Elasticsearch 6

2.2.3 Kibana 7

2.3 Development Methodology 7

2.3.1 Delivery Cadence 8

2.3.2 Collaborative Tools 8

1 Background

> You want to create something that people will use… Your product must have unique characteristics that set it apart from any competing products that are already in the market.

Scientia potentia est: knowledge is power. It is those which control knowledge, information itself that are truly powerful.

More so than at any point in the history of the world, information has become the market's scarcest resource. This claim seems to be paradoxically at odds with the near-ubiquity of information technology; indeed, never before have so many had the ability to gather so much information. However gathering information and gathering *pertinent* information are two very different things.

A company which produces smartphones does not benefit from knowing the market price of laundry detergent. Likewise, a company which manufactures and sells laundry detergent would not benefit from having information about the specifications of the next generation of smartphones. However, both companies would benefit from knowing information about their competitors and the demographics of their markets. Increasingly, the companies that succeed in the free market are those that have the largest capacity to ingest pertinent information, enrich it, and analyze it to produce yet more information.

Facebook and Google, the "F" and "G" of FANG, are two of the market's four best-performing tech stocks. Facebook and Google's primary sources of revenue comes from selling ads through their respective platforms. Part of why Facebook and Google have excelled where their competitors have stalled is their ability to acquire information about their users. With every post, click, or search users of these platforms tell the monolithic companies behind own them their secrets. This in turn allows them to more effectively advertise toward these users.

Information in and of itself has become a market of sorts. Numerous companies have sprung up in the information market, each purporting to deliver analytics packages to businesses which would perform some miracle like doubling their marketing reach or click-through rate. These offerings are particularly enticing to businesses without the fortune of controlling a platform which passively aggregates their target market's information.

It is these businesses that Team Big Data wants to serve.

2 Proposal

2.1 Offering

Team Big Data (TBD) seeks to fill a perceived gap in the market. Most companies lack a platform which passively aggregates information about their target market (à la Google and Facebook). These companies stand to benefit from analytics about the audiences they hope to reach with their advertisements.

TBD's proposed product is an analytics package which performs sentiment analysis of some number of users tweets. The sentiment analysis of a user's tweet can be thought of as representing the user's emotional state (negative/neutral/positive) which can be used to further target advertisements.

TBD's proposed analytics package consists of three parts:

1. A NiFi pipeline which fetches tweets from potential customers and performs sentiment analysis
2. An Elasticsearch backend which ingests data from NiFi
3. A Kibana frontend which provides visualizations, metrics, and analysis

2.2 Proposed Technology Stack

TBD's technology stack consists of three main components:

- NiFi
- Elasticsearch
- Kibana

The following section breaks down the general setup and dependencies of each technology.

2.2.1 NiFi

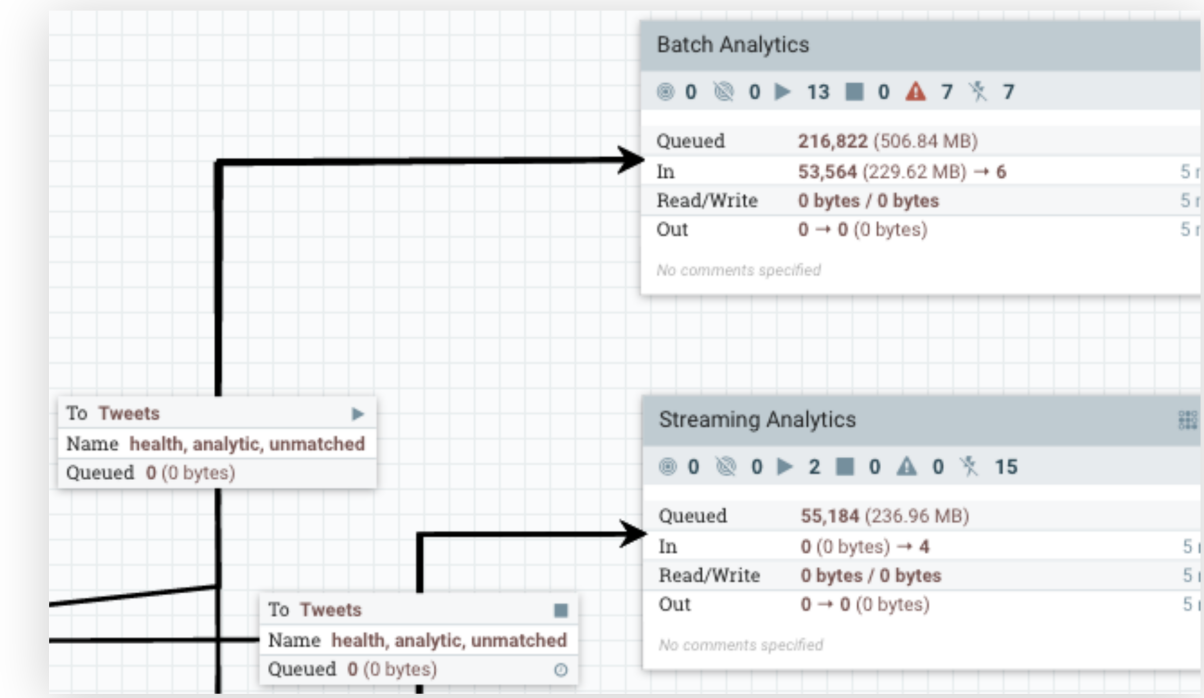


Figure 1: A portion of a NiFi flow taken from the Apache NiFi homepage, https://nifi.apache.org/.

Apache NiFi is a flow-based programming tool mean to automate the flow of data between systems.

The atom of the NiFi Flow is that of the Processor. A Processor performs some function, be it modifying the content that passes through it or simply redirecting it to a new destination. Processors in turn can be grouped together into Processor Groups. Processor Groups are typically used to isolate different functional portions of the Flow from one another.

We selected NiFi for two main reasons:

1. NiFi is open source (which makes security audits easy) and has a great deal of documentation, and
2. NiFi provides an API one can use to write new processors (in Java) to add functionality.

The NiFi portion of TBD's technology stack takes care of three main tasks:

1. Fetching data from the edge
2. Transforming and enriching data
3. Ingesting data into a data store

Each is described in more detail in the following subsections.

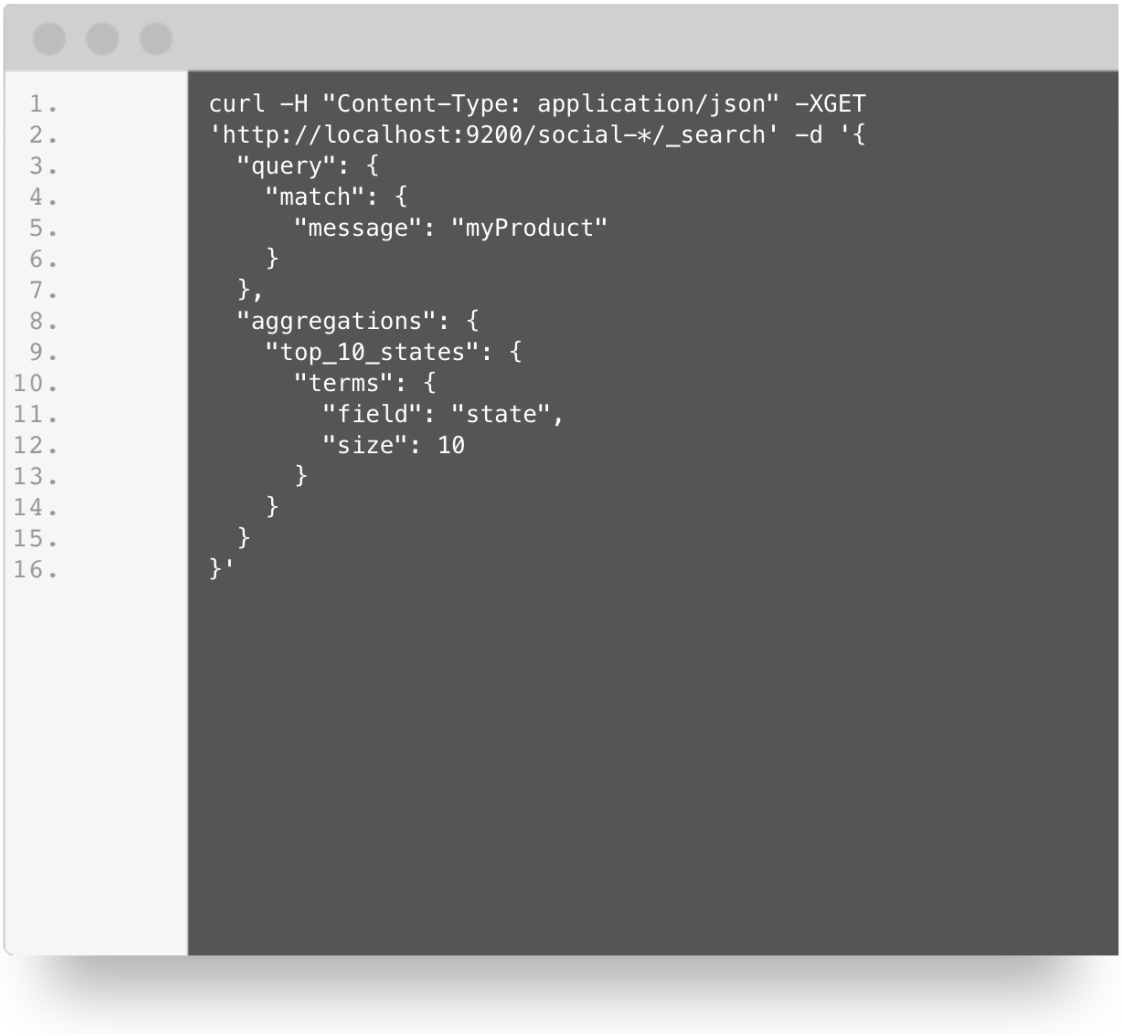
Fetching data from the edge NiFi comes with a Processor which, when configured, is able to fetch data from Twitter as a JSON object.

Transforming and enriching data The JSON object Twitter yields contains a great deal of data that is irrelevant to our needs. To avoid bloat in our data store due to maintaining information which is unneeded, we extract the relevant fields. This extraction is performed by `jq`, a tool like `sed` built for JSON (https://stedolan.github.io/jq/).

The body of the tweet is then transformed into a JSON object containing the results of the text's sentiment analysis. The sentiment analysis is performed by Vader Sentiment, a tool written in python (https://github.com/cjhutto/vader Sentiment).

Ingesting data into a data store NiFi ships with the capability to ingest into some Elasticsearch index.

2.2.2 Elasticsearch



```
curl -H "Content-Type: application/json" -XGET
'http://localhost:9200/social-*/_search' -d '{
  "query": {
    "match": {
      "message": "myProduct"
    }
  },
  "aggregations": {
    "top_10_states": {
      "terms": {
        "field": "state",
        "size": 10
      }
    }
  }
}'
```

Figure 2: An example of interacting with Elasticsearch's RESTful API, taken from the Elasticsearch homepage, https://www.elastic.co/products/elasticsearch

Elasticsearch is a fast, resilient, and distributed search and analytics engine. Elasticsearch uses a RESTful API and JSON, but also has clients for a number of different languages.

Elasticsearch was selected as the data store due to its performance and tight integration with its companion application, Kibana, which provides insight on the data stored with Elasticsearch.

2.2.3 Kibana

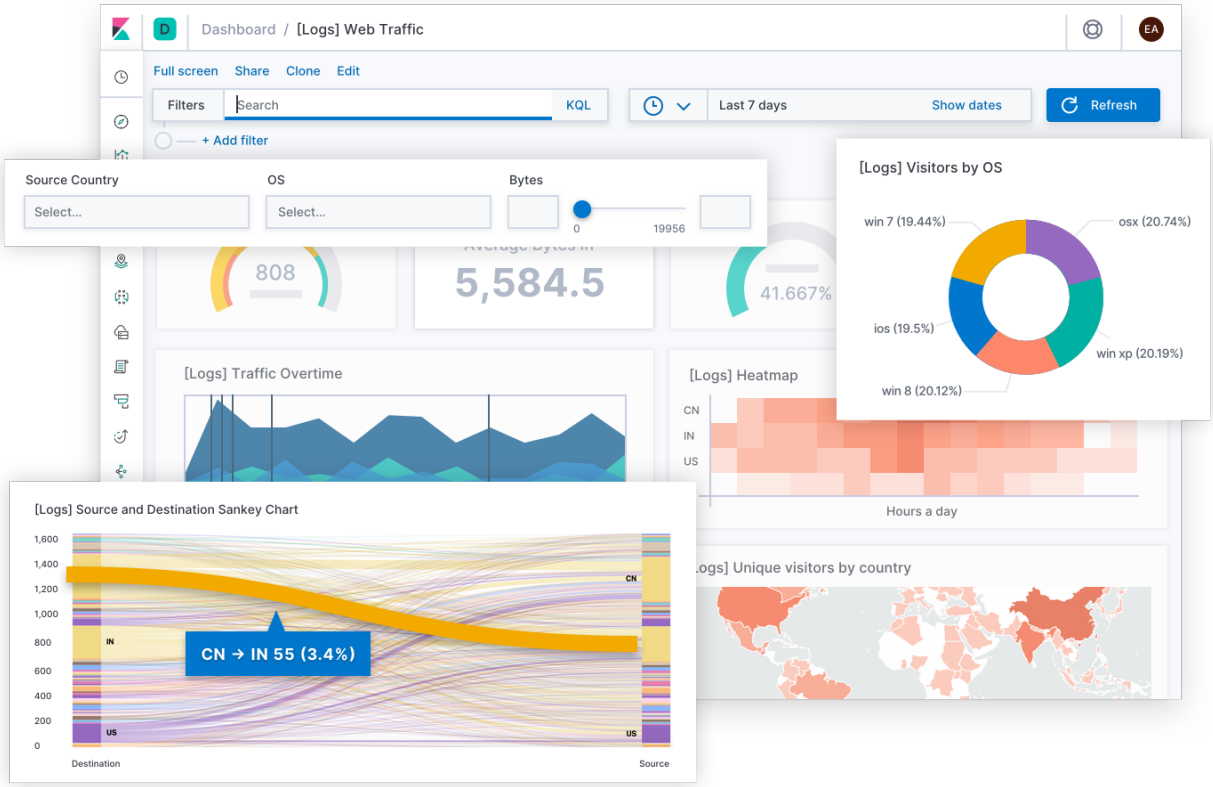


Figure 3: A Kibana dashboard taken from the Kibana homepage, https://www.elastic.co/products/kibana.

Kibana is made by the same company behind Elasticsearch and is billed as the “window” into Elasticsearch. Kibana comes with dozens of visualization templates, each of which can be expanded and built on.

Kibana is built by the same company that makes Elasticsearch – they are made to work together. Pairing the two of them yields best-in-class performance, maintainability, and ease of use.

2.3 Development Methodology

To enable high-velocity development, TBD embraces the several techniques from Extreme Programming (XP) and the Scaled Agile Framework (SAFe).

While SAFe is highly regarded, even the smallest configuration, Essential SAFe, requires a team about three times the size of TBD’s. As such, rigorously following SAFe is impractical. However, by taking techniques from XP and

customer-facing ceremonies from SAFe, TBD is able to reap the benefits of both.

2.3.1 Delivery Cadence

The semester is 16 weeks long. For this reason, TBD proposes that each sprint lasts two weeks. Every two sprints is followed by a Planning Increment, the purpose of which is to allow TBD to meet with the customer (the Professors and TAs) to better synchronize with their needs.

2.3.2 Collaborative Tools

TBD will use the following tools to enable collaboration.

Slack Slack will serve as the development team's primary means of communication. Instant messaging and the ability to easily share screenshots helps with the rate at which problems can be identified, diagnosed, and solved with distributed teams.

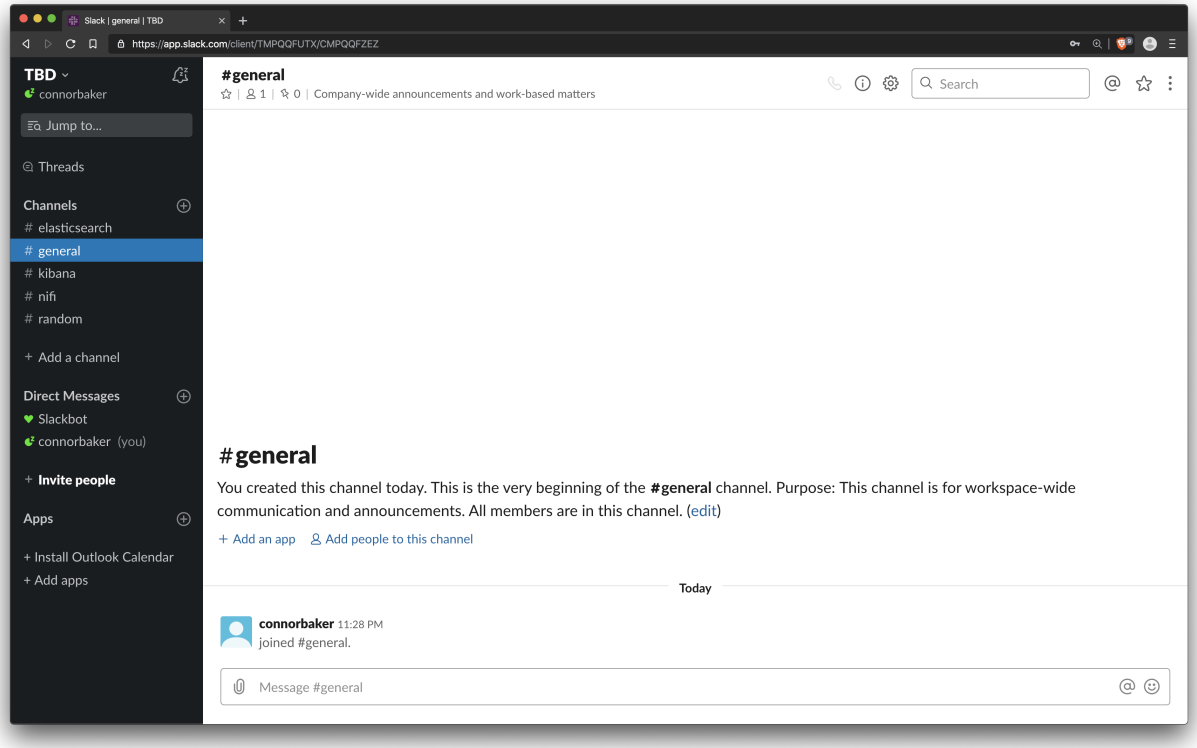


Figure 4: TBD's Slack.

Trello Trello will be the team's agile workflow. There will be one board per sprint, each serving as our backlog and Kanban board.

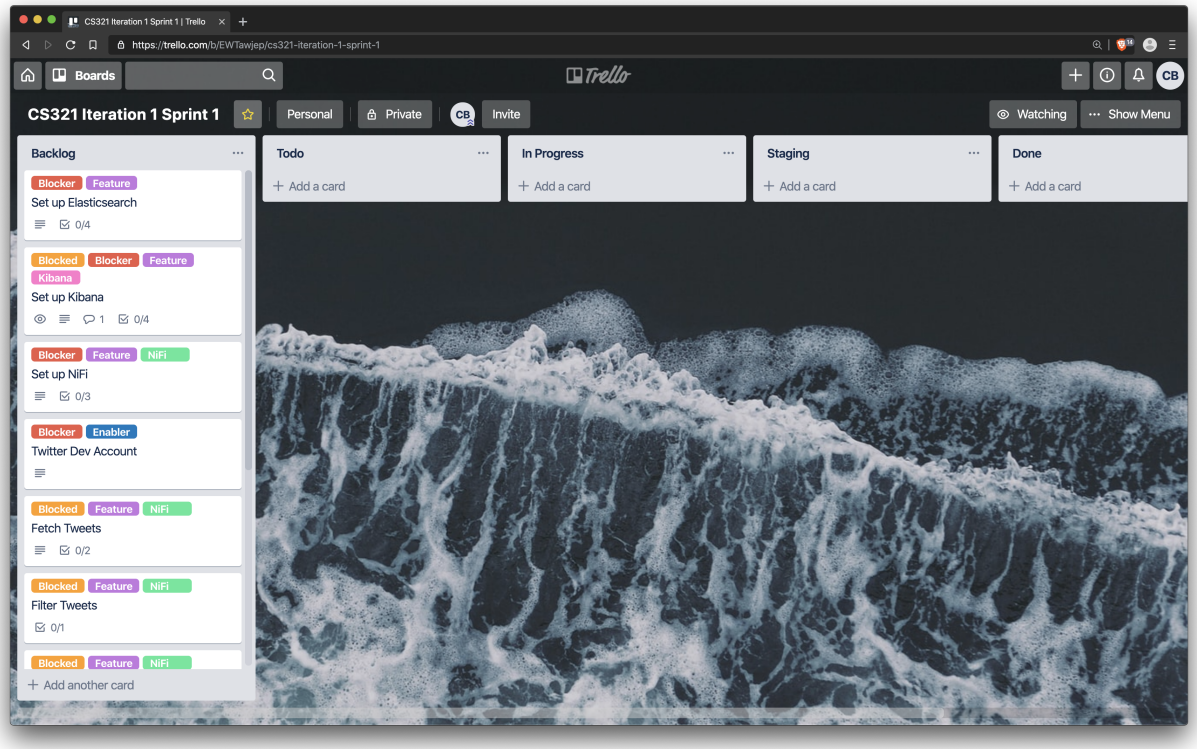


Figure 5: TBD's Iteration 1, Sprint 1 Trello Board.

GitHub GitHub will serve as our revision control system (RCS). In addition to being an RCS, it will also serve as our document repository, thanks to its Wiki functionality.