ROAD MAP

1. Road Map
2. Write application description and POC intent
3. Design Services Diagram
4. Research APIs with benefits to the app
5. Define technologies
6. Build basic flow of platform
7. Define responsibility for each service
8. Define API and their behavior
9. Architect Application
10. Define data models
11. Get API keys
12. Create each individual service
13. Setup local cluster

STOCK ANALYZER

POC

The purpose of the service is to pull stock information about certain symbols and persist in to ElasticSearch.

Then it pulls from the web a series of resources such as social media data, news articles, weather, and other data that could potentially be useful to this project. Clean and curate the data, and run it through machine learning pipelines to find different scores. These sources with their corresponding scores will be persisted into ElasticSearch.

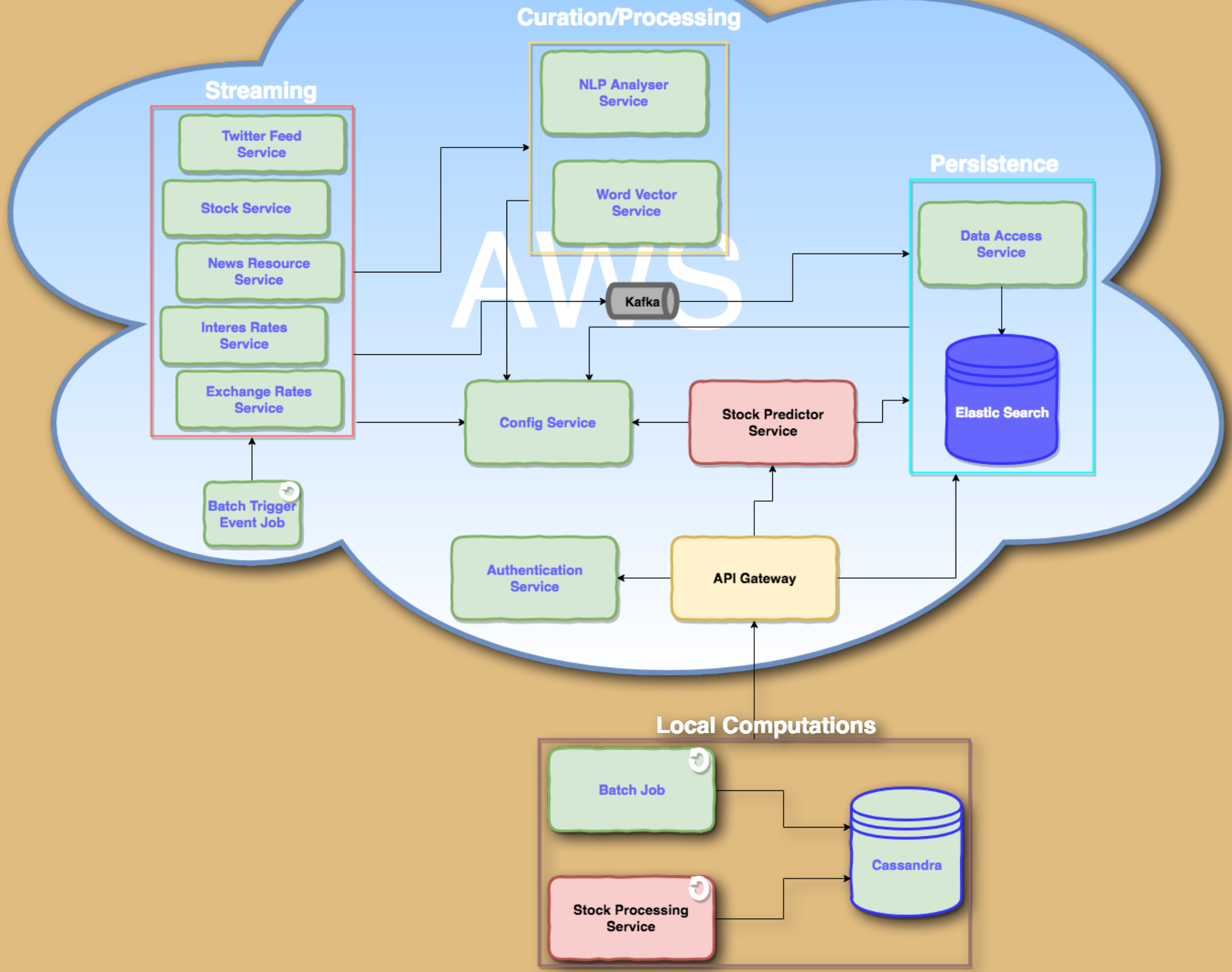
There will be a local dedicated batch job to pull the data from ElasticSearch every week and clean ElasticSearch after its done.

Another local spark job will be dedicated to analyze the data and find possible correlations between the two.

The service will be comprise of 4 stages:

1. Real time streaming of resource data (Cloud)
2. Data processing and scoring (Cloud)
3. Data persistence (Cloud)
4. Stock analyzer jobs (Local)

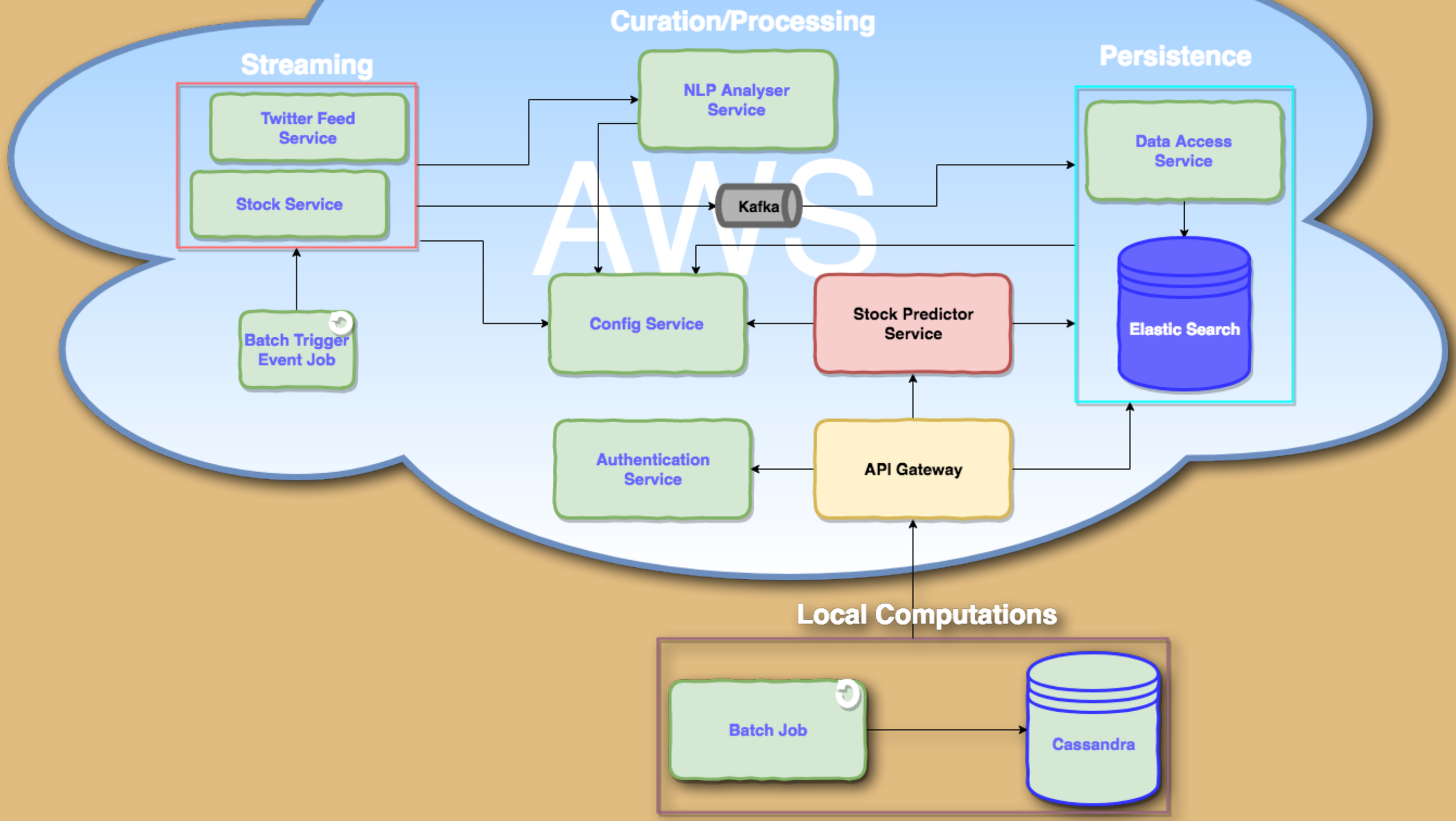
ARCHITECTURE DIAGRAM



RESOURCES

Yahoo/Weather API
https://github.com/Java-Techie-jt/yahoo-stock-web-api
Twitter API
https://developer.twitter.com/en/apps/15924304
News API
https://newsapi.org/pricing
Machine Learning
NLP
ElasticSearch
Intrinio: Stock Data
https://intrinio.com/
https://github.com/intrinio/intrinio-realtime-java-sdk
IBM Predictive Market
https://console.bluemix.net/apidocs/predictive-market-scenarios?
https://github.com/IBM/Predictive-Market-Stress-Testing/blob/master/README.mdlanguage=java
https://github.com/IBM/Predictive-Market-Stress-Testing
Exchange Rates
https://fixer.io/quickstart
QuanDL: Stock Data
https://blog.quandl.com/category/api-usage-guides?utm_source=google&utm_medium=organic&utm_campaign=&utm_content=api-for-interest-rate-data
Bonzai
https://bonsai.io/

BASIC FLOW


Curation/Processing
Streaming
Persistence
Twitter Feed Service
Stock Service
NLP Analyser Service
Kafka
Data Access Service
Elastic Search
Batch Trigger Event Job
Config Service
Stock Predictor Service
AWS
Authentication Service
API Gateway
Local Computations
Batch Job
Cassandra

TECHNOLOGIES

Java 8
SpringBoot
Microservices
AWS
Cassandra
Kafka
Maven
Git
Apache Spark
Yahoo/Weather API
Twitter API
News API
Linux

Machine Learning
NLP
ElasticSearch
Intrinio API
IBM Predictive Market
https://fixer.io/quickstart
QuanDL

STREAMING SERVICES

Twitter Feed Service

- Get twits based on topics.
- Rates each twit with a sentiment value score.
- Persist the twit to ElasticSearch

Stock Service

- Get stock price every 5 minutes.
- Persist the stock into ElasticSearch

News Resource Service

- Get news resources.
- Filters news, finding paragraphs related to topics.
- Rates filtered sources with a sentiment value.
- Persist title and location of paragraph with score.
- Get current weather and persist.

DATA MANIPULATION SERVICES

NLP Analyser Service

- Calculates the sentiment value of text
- Separates documents into paragraphs.
- Removes stop words from text.

Word Vector Service

- Calculates synonyms for a list of words.

PERSISTENCE SERVICES

Data Access Service

- Dedicated API to persist/retrieve data into ElasticSearch.

ElasticSearch

- Stores all data related to other services.

UTILITY SERVICES

API Gateway

- All possible REST endpoints should be defined here.

Authentication Service

- Will most likely have an in-memory authentication system.

Config Service

- All environment variables, keys, and important constants should be defined here.

LOCAL SERVICES

Datastore Cassandra

- Will be setup at the LAN.

Batch Job

- Job will most-likely run 2-3 times a day for when the market opens and closes.
- Pulls data from DAO Service to persist in Cassandra

Stock Processing Service

- Process home data.