

# Comprehensive Big Data Solution for Stock Prices and Tweets Collection

Paul Adams

Rikel Djoko

Stuart Miller



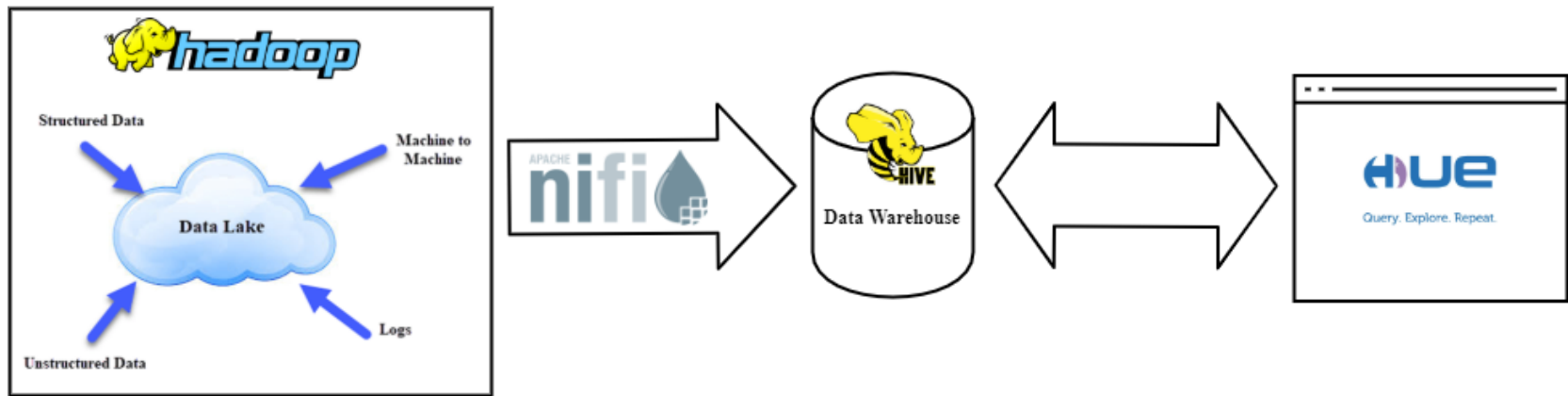
# Problem Statement

With the proliferation of data across the internet we want to use raw and unstructured data to build a large-scale data framework that will enable us to store and analyze financial market data and drive future predictions for investment.



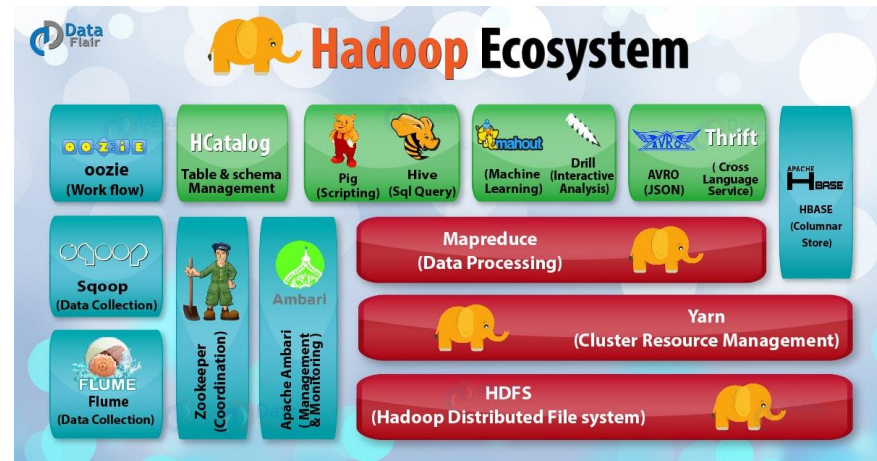
# Big Data Solution

- 1) Data loaded into data lake, Hadoop
- 2) Processed via data flow automation, NiFi
- 3) Loaded into data warehouse, Hive
- 4) Data queried using web-based Hue



# Apache Hadoop “Data Lake” Ecosystem

- Hadoop Distributed File System (HDFS)
  - rapid data transfer between server-nodes in cluster
    - data-parallelism
    - task-parallelism



<https://data-flair.training/blogs/hadoop-ecosystem-components/>

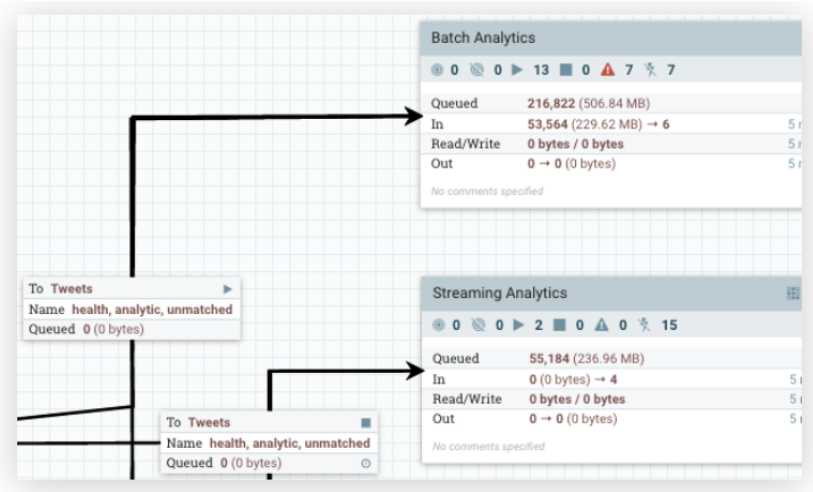


# Apache NiFi Data Flow

- Automates data flow between systems using a FlowFile
- FlowController provides threads for extensions



An easy to use, powerful, and reliable system to process and distribute data.



# Apache Hive Data Warehouse

- Stores large amounts of data from disparate sources
- Hive Query Language (HQL)
- Operates on Hadoop data lake
  - Twitter Feeds
  - Structured Stock Market Data



# Cloudera Hue

- browser-based user interface for Hive

The screenshot displays the Cloudera Hue web interface. At the top, there is a navigation bar with the Hue logo, a 'Query' dropdown menu, and a search bar for saved documents. Below this, the interface is divided into several sections:

- Left Panel:** A sidebar showing the current database 'ds7330\_term\_project' and a list of tables. The 'intraday' table is selected, showing its schema with columns like 'times', 'symbol', 'volume', 'open', 'high', 'low', 'close', 'band\_high', 'band\_mid', and 'band\_low'.
- Center Panel:** A code editor containing a Hive SQL query. The query creates a table 'intraday' if it does not exist, inserts data into it, and then selects the first two rows. The query is as follows:

```
41 create table if not exists ds7330_term_project.intraday(  
22 times string  
23 ,symbol string  
24 ,volume bigint  
25 ,open double  
26 ,high double  
27 ,low double  
28 ,close double  
29 ,band_high double  
30 ,band_mid double  
31 ,band_low double);  
32  
33 insert into ds7330_term_project.intraday  
34 select  
35 i.times as trade_time  
36 ,i.symbol as symbol  
37 ,i.volume as volume  
38 ,i.open as open_price  
39 ,i.high as high_price  
40 ,i.low as low_price  
41 ,i.close as close_price
```
- Right Panel:** A console showing the execution status and logs. It indicates that concurrency mode is disabled and that the query was executed successfully. The logs show: 'INFO : Concurrency mode is disabled, not creating a lock manager' and 'INFO : Executing command(queryId=hive\_20191018151642\_999a84d2-6b24-437f-9024-9bd461c8ab26): select\* from intraday limit 20'.
- Bottom Panel:** A table showing the results of the query. The table has columns for 'intraday.times', 'intraday.symbol', 'intraday.volume', 'intraday.open', 'intraday.high', 'intraday.low', 'intraday.close', 'intraday.band\_high', 'intraday.band\_mid', and 'intraday.band\_low'. The results are as follows:

	intraday.times	intraday.symbol	intraday.volume	intraday.open	intraday.high	intraday.low	intraday.close	intraday.band_high	intraday.band_mid	intra
1	"2019-09-09 14:30:00"	"AAPL"	592595	214.18	214.3899	213.96	214.06	216.3653	209.4339	202.5
2	"2019-09-09 14:45:00"	"AAPL"	601480	214.07	214.5	213.95	214.19	216.4122	209.4796	202.5



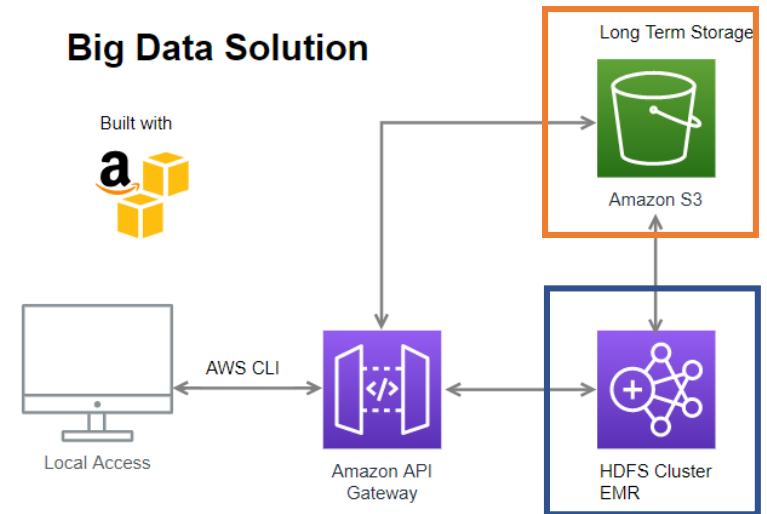
# AWS Based System

AWS Elastic MapReduce Cluster

Pre-configured Hadoop System

- Hadoop
- Hive
- Hue

S3 Storage



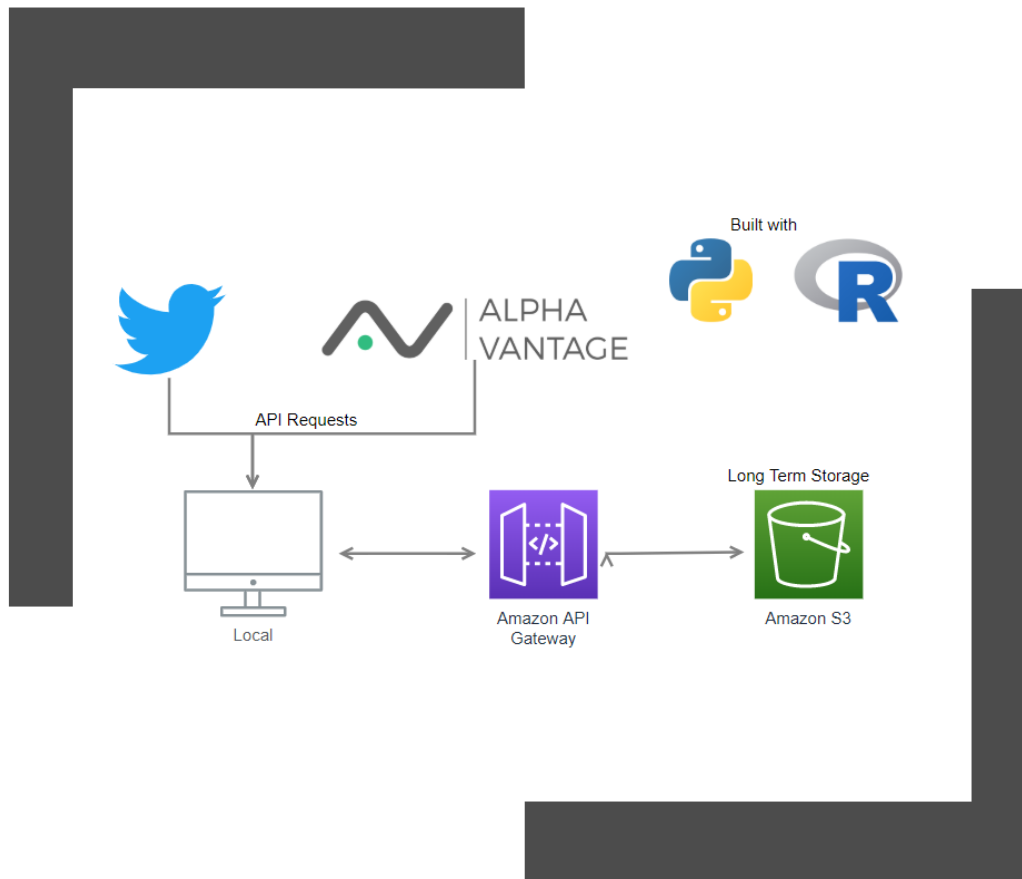


# How it Works

- Cluster created on demand
- Data loaded from S3 bucket
- Runs Hive scripts to create database
- Interact with database with
  - Hue Interface
  - Command Line



# Data Collection



- Daily Stock Prices
- Intra-day Stock Prices
- Tweets



# Financial Data

- Data gathered in full normal form
- Data gathering from API key through Alpha Vantage in R
  - NASDAQ and NYSE
    - data updates by 15-minute intervals
    - Time series prices, volumes, Bollinger bands, stochastic indicators

times	open	high	low	close	volume	symbol
8/28/2019 9:45	204.5061	204.9361	203.5148	203.9518	2166112	AAPL
8/28/2019 10:00	203.96	205.25	203.73	204.86	1171569	AAPL
8/28/2019 10:15	204.93	205.36	204.68	204.81	780524	AAPL
8/28/2019 10:30	204.8174	205.1904	204.4898	204.5774	677803	AAPL
8/28/2019 10:45	204.59	205.45	204.59	205.215	750300	AAPL
8/28/2019 11:00	205.21	205.34	204.93	205.08	429695	AAPL
8/28/2019 11:15	204.97	205.0175	204.62	204.74	529615	AAPL
8/28/2019 11:30	204.7633	205.6279	204.6721	205.4233	764153	AAPL
8/28/2019 11:45	205.5954	205.8893	205.4489	205.4684	499739	AAPL
8/28/2019 12:00	205.47	205.56	205.12	205.32	323214	AAPL



# Tweet Data

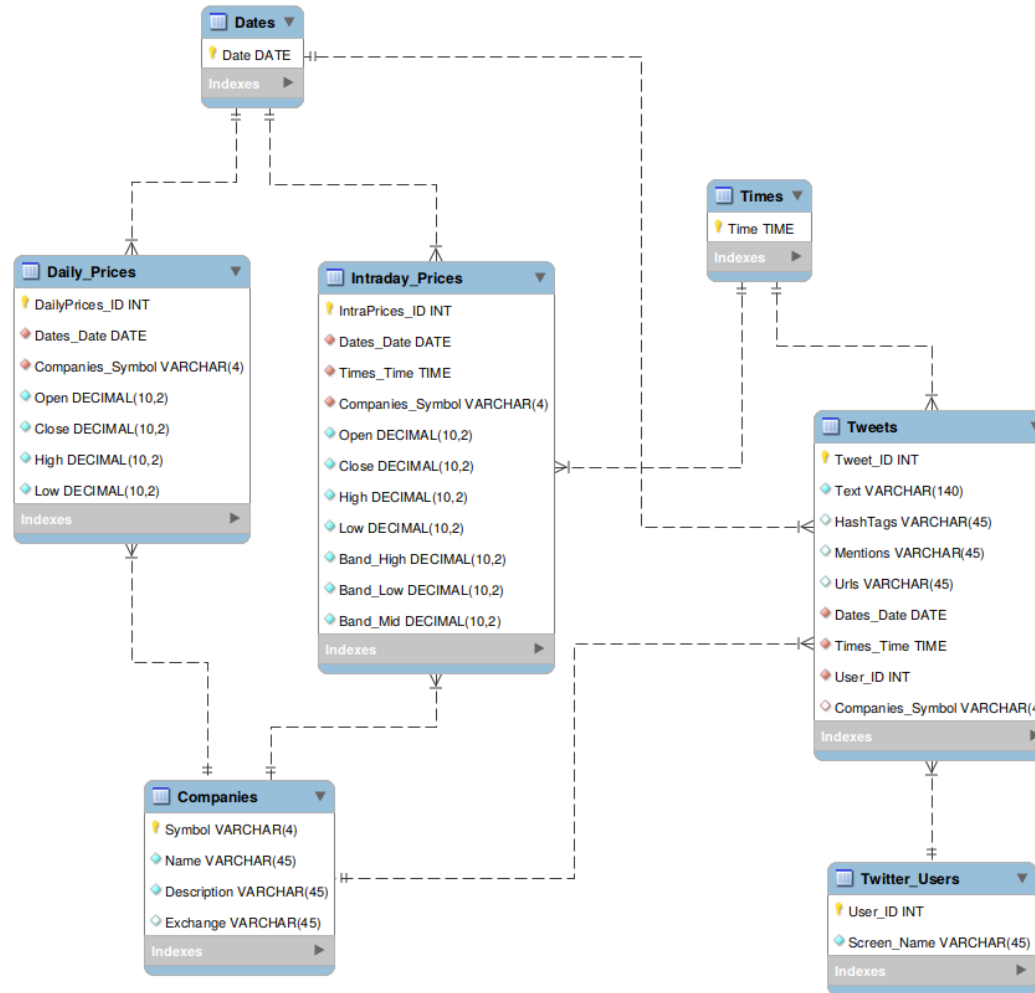
Features

300,000 json files



- Tweet Text
- Hashtags
- Urls
- Mentions
- Post time



# Current Schema Proposal



# Milestones & Status

Milestones(MS)	Expected Time of Arrival(ETA)	Actual (Trend)	Status
AWS Infra Setup	WW40	WW40	Done 
Data Collection	WW41	WW40	Done 
Integrate sample data into Hadoop	WW43	WW42	Pending 
Integrate all tools to Data Warehouse (Hive, NiFi Hue, Hadoop)	WW43		
Load data to Data Warehouse	WW44		
Transform data and build Schema	WW45		
Use HiveQL to access data for Analysis	WW46		
Documentation(final paper)	WW47		
Final presentation	WW48		

