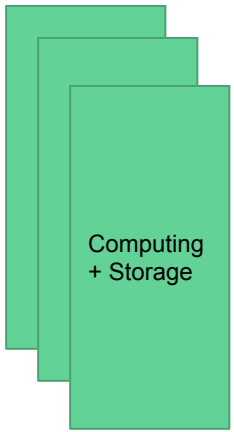
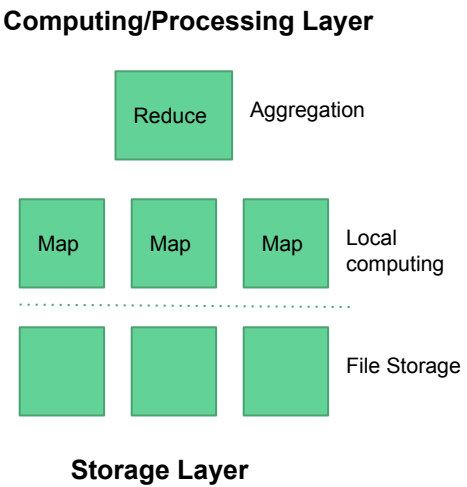
Intro Big Data Hadoop & Architecture Evolution

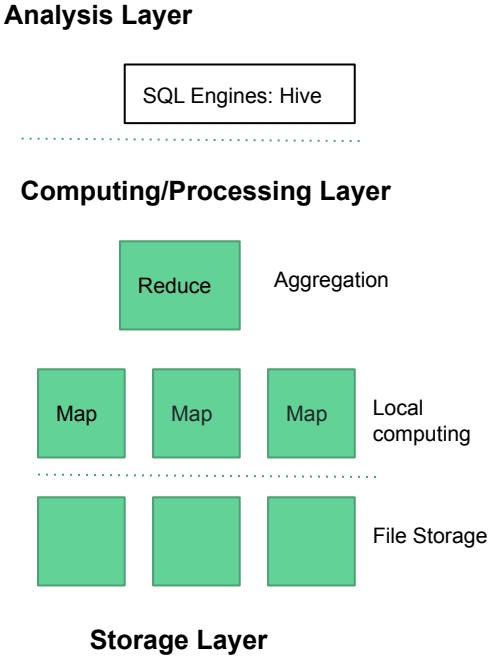
Evolution of the systems towards big data architecture



Vertical scaling, monolithic systems

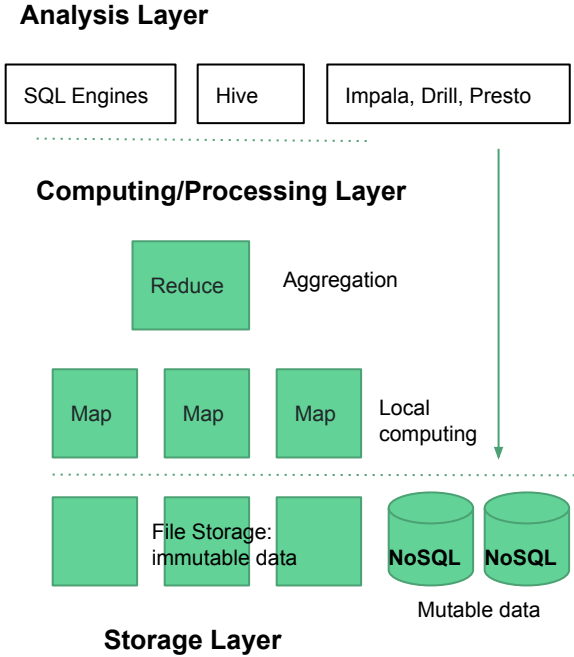


Hadoop: Distributed systems, horizontal scaling, separation of storage from computing, local and aggregation in computing



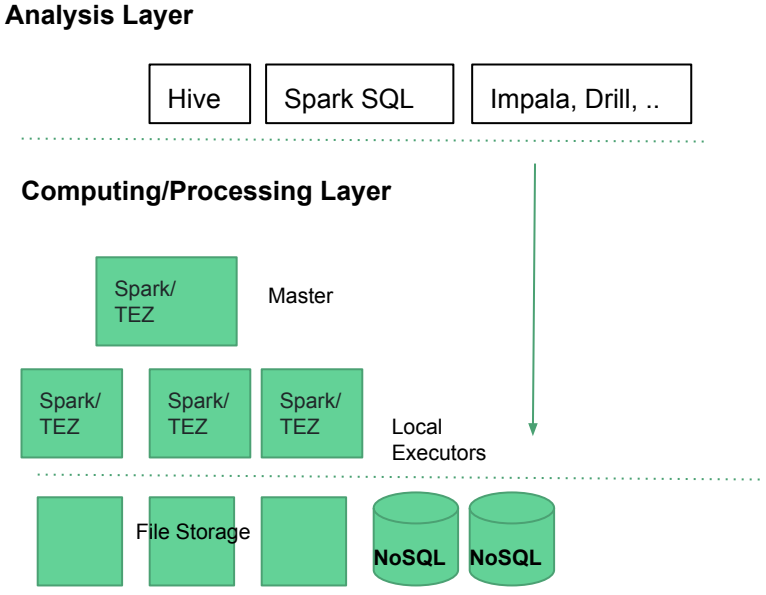
New analysis layer: to allow easy data analysis - SQL

Resource Management: YARN



New tools in analysis layer: tools that can work directly with the storage layer.
Storage layer: NoSQL: storage system that allows random and rapid access to data

A new computing fw: Apache Spark

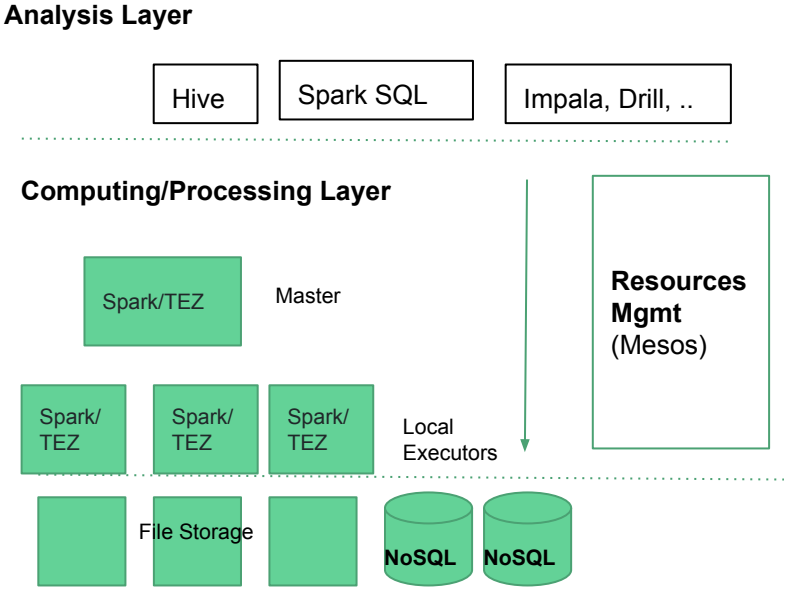


Spark Core came as a replacement for old Map Reduce and to improve computing times. **But, Spark evolved into a framework that consists of Spark Core, SQL, Streaming, Graph, Machine Learning..**

Big Data Architecture

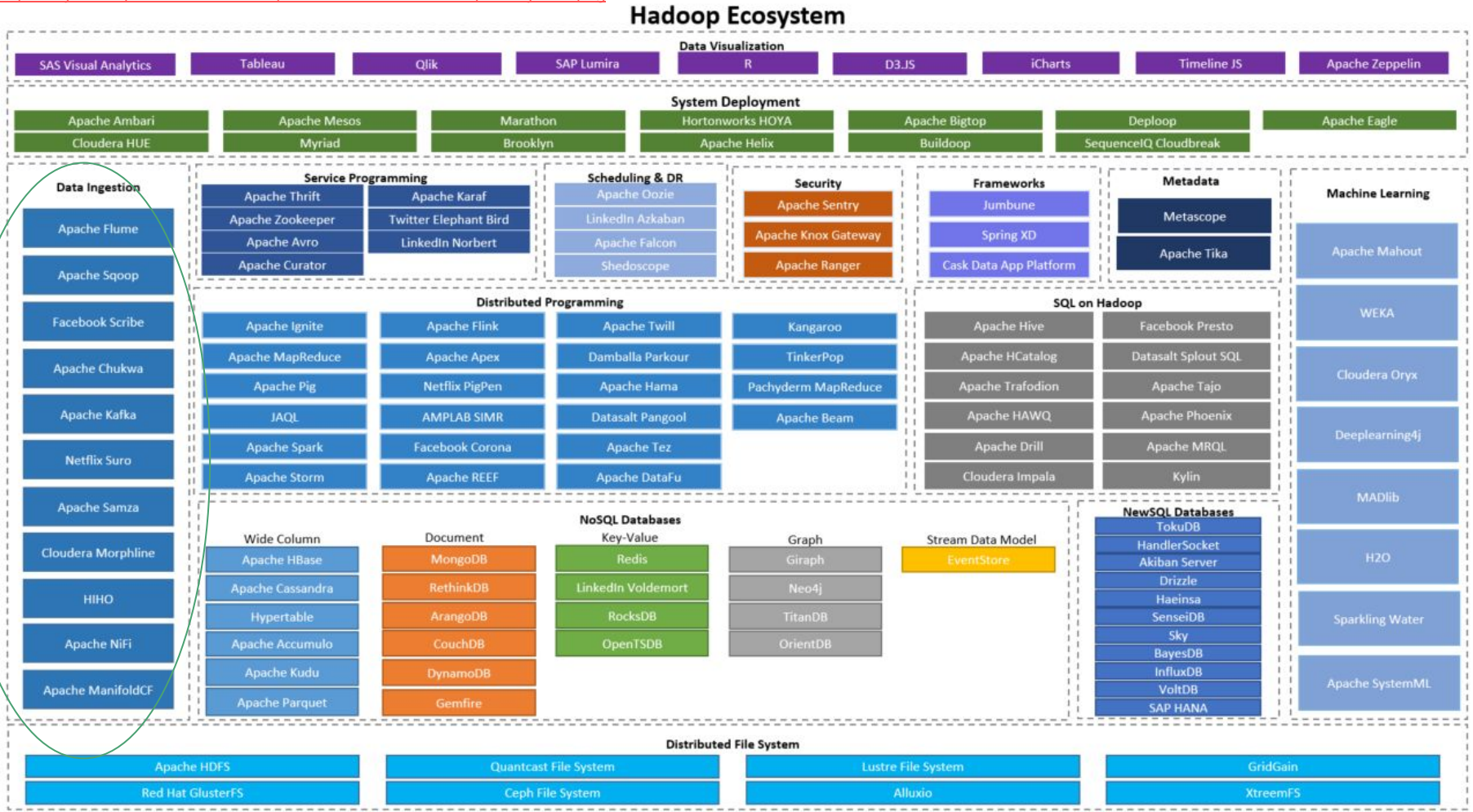
Visualization Layer (Zoomdata, Tableau, Zeppelin,..)

Ingestion Layer (Kafka, Nifi, …)

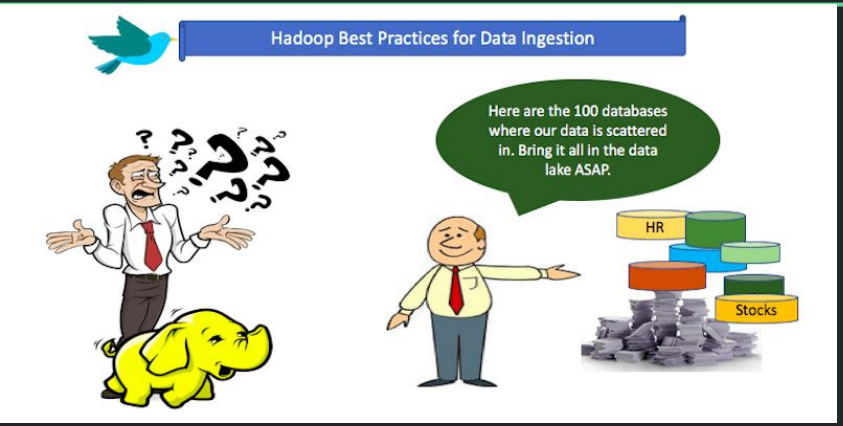


Other layers: Ingestion, Visualization, Security, ...

https://mydataexperiments.files.wordpress.com/2017/04/hadoop-ecosystem.png



Ingestion layer



Data ingestion

- **Collection of data from different sources - Ingestion** (“some data is stored in Hadoop, some in Oracle, and a little bit is in Excel”)
  - IOT like case (loads of small sources of data): tool that has a mini client version
  - Specialized tools: like Sqoop for moving data between RDBMS and Hadoop
- **Data Flow management**
  - Optimal movement of data between different systems
- **Messaging bus/broker**
  - Reliable storage of ingestion data
  - Reprocessing of data

Solutions in data ingestion space

- **Sqoop -- HDFS focused tool**
  - uses JDBC drivers to extract data from relational databases and then directly into the Hadoop stack, typically to Hive (or another metastore governed tool) or HDFS. Has options to schedule incremental data loads. Extracts are run in Mappers using Hadoop Map-Reduce and highly parallelized.
- **Flume -- HDFS focused tool**
  - The original log collector for Hadoop. It is made up of sources (where is the data coming from), channels (what are you doing with the data once you've grabbed it), and sink (where is it going). It is effectively replaced nowadays by a combination of Kafka/Storm. One cool thing Flume does do is that it can bind to a source (such as a web stream) and push data all the way to HDFS without having to write any code, all in one tool. Also supports contextual routing.
- **Streamsets & Streamsets Data Collector**
  - SDC was started by a California-based startup in 2014 as an open source ETL project available on GitHub. The first release was published in June 2015. Cloudera supported solution for dataflow management: builds, tests, runs and maintains dataflow pipelines connecting a variety of batch and streaming data sources and compute platforms.

Solutions in data ingestion space

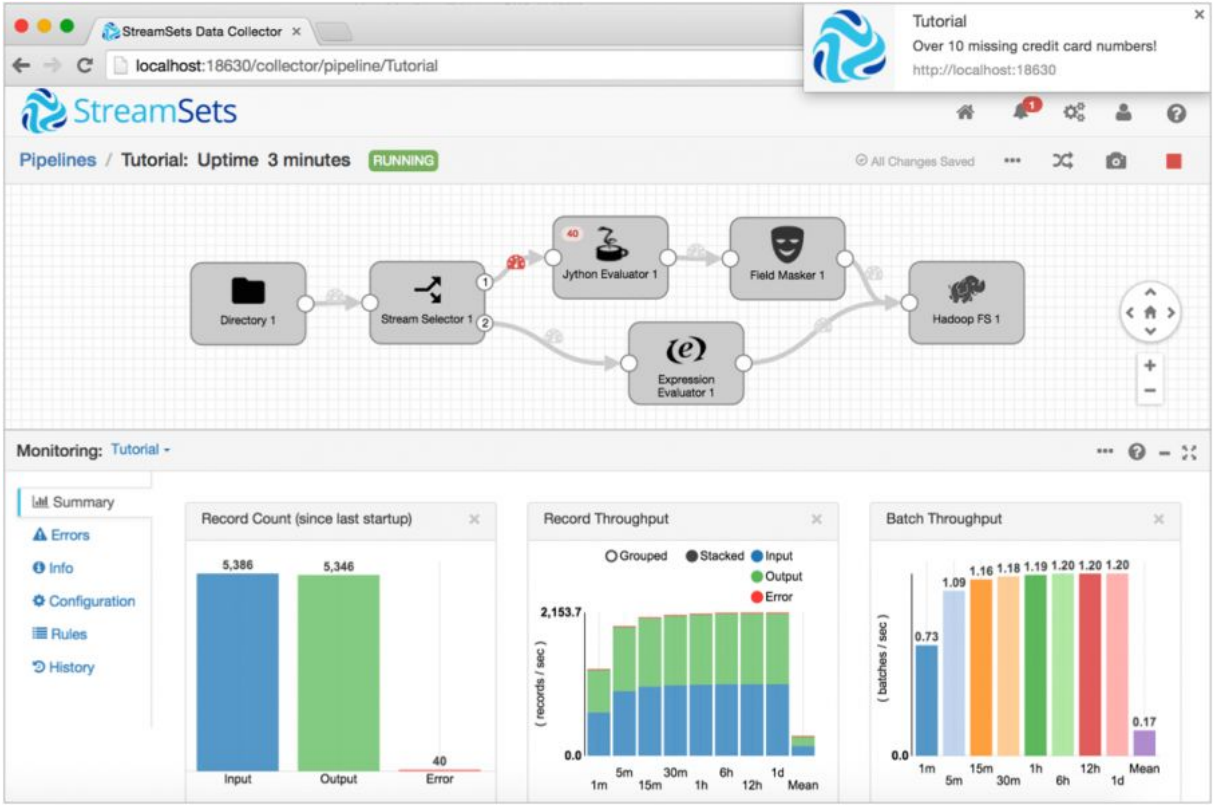
- **Nifi**
  - Hortonworks driven dataflow management solution. NiFi can be used in mission-critical data flows with rigorous security & compliance requirements, where we can visualize the entire process and make changes immediately, in real-time.
- **Apache Spark**: Non specific to ingestion layer, its computing framework can be used for ingestion data aggregation
- **Kafka**
  - Distributed messaging bus
- **Others: Kinesys**

Flume, NiFi, Kafka

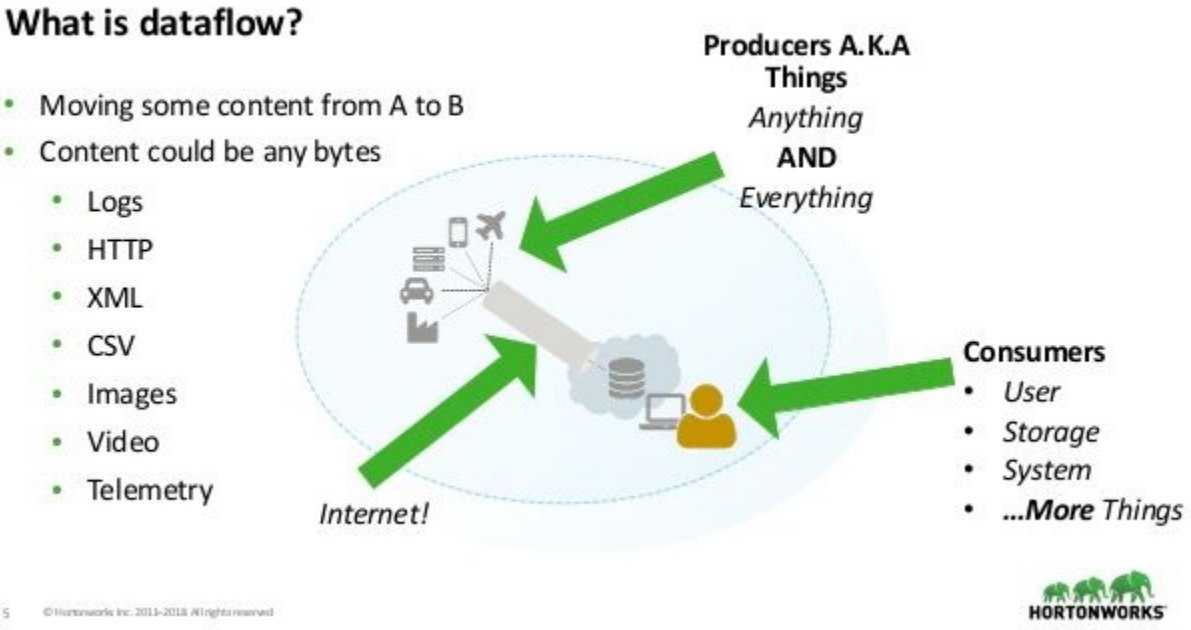
Ingestion Tool	Out-of-the-box	Limits	Use Cases
Flume	- Configuration-based - Sources, channels & sinks - Interceptors	- Data loss scenarios when not using Kafka Channel - Data size (KB) - No data replication	- collecting, aggregating, and moving high-volume streaming events into Hadoop.
Kafka	- Back-pressure - Reliable stream data storage - Kafka-Streams - Sources/sinks with Kafka-Connect	- Custom coding often needed - Data size (KB) - Fixed protocol/format/schema	- Streaming data - Messaging - Systems integration - Commit log
NiFi	- Configuration-based UI - Many drag & drop processors - Back-pressure - Prioritized queuing - Data provenance - Flow templates	- Not for CEP or windowed computations - No data replication	- Dataflow management with visual control - Data routing between disparate systems - Arbitrary data size

can handle messages with arbitrary sizes

Streamsets


StreamSets Data Collector
localhost:18630/collector/pipeline/Tutorial
Tutorial
Over 10 missing credit card numbers!
http://localhost:18630
StreamSets
Pipelines / Tutorial: Uptime 3 minutes
RUNNING
All Changes Saved
Directory 1
Stream Selector 1
Jython Evaluator 1
Field Masker 1
Hadoop FS 1
Expression Evaluator 1
Monitoring: Tutorial
Summary
Errors
Info
Configuration
Rules
History
Record Count (since last startup)
5,386
5,346
40
Input
Output
Error
Record Throughput
Grouped
Stacked
Input
Output
Error
2,153.7
0.0
(records / sec)
1m
5m
15m
30m
1h
6h
12h
1d
Mean
Batch Throughput
0.73
1.09
1.16
1.18
1.19
1.20
1.20
1.20
0.17
0.0
(batches / sec)

Data Flow Management



Source: https://www.slideshare.net/Hadoop_Summit/apache-nifi-crash-course-98121902

Apache NiFi

NiFi was donated by the NSA to the Apache Foundation in 2014 and current development and support is provided mostly by Hortonworks.

Used mainly for

- Reliable and secure transfer of data between systems
- Delivery of data from sources to analytics platforms
- Enrichment and preparation of data:
  - Conversion between formats
  - extraction/parsing
  - Routing decision

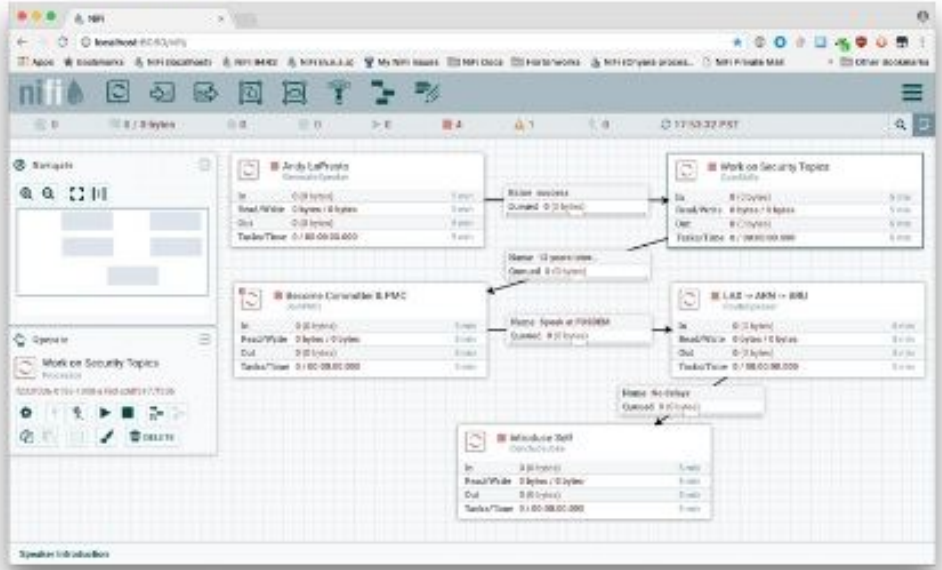
Note: Nifi dosent have a built-in fault-tolerant mechanism. Meaning if one node goes down then there has to be a manual intervention to prevent data loss/redirect flow etc

What is not used for

- Distributed computing
- Complex event processing, Joins/Complex Rolling Windows operations
- Data reprocessing - e.g. Kafka like

User Interface

Less of this... **... more of this**



HORTONWORKS