CLOUDERA

Real-Time Streaming Pipelines With FLaNK (Apache Flink, Apache NiFi & Apache Kafka)

Timothy Spann - Principal DataFlow Field Engineer

15-April-2021

DATACON LA

FLaNK

flankstack.dev

Tim SPANN

https://github.com/tspannhw https://www.datainmotion.dev/

https://www.meetup.com/futureofdata-princeton/

FLaNK Stack for Cloud Data Engineers

Multiple users, frameworks, languages, clouds, data sources & clusters

CLOUD DATA ENGINEER

- Experience in ETL/ELT
- Coding skills in Python or Java
- Knowledge of database query languages such as SQL
- Experience with Streaming
- Knowledge of Cloud Tools

CAT

- Expert in ETL (Eating, Ties and Laziness)
- Edge Camera Interaction
- Typical User
- No Coding Skills
- Can use NiFi
- Questions your cloud spend

AI / Deep Learning / ML / DS

- Can run in Apache NiFi
- Can run in Kafka Streams
- Can run in Apache Flink
- Can run in MiNiFi Agents

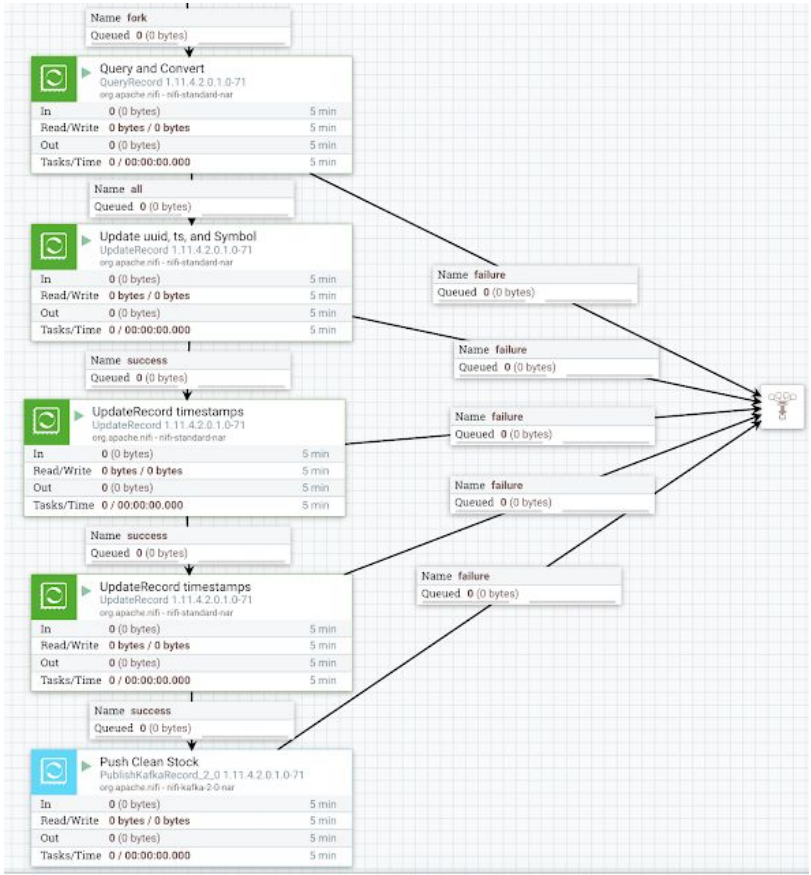
FLaNK

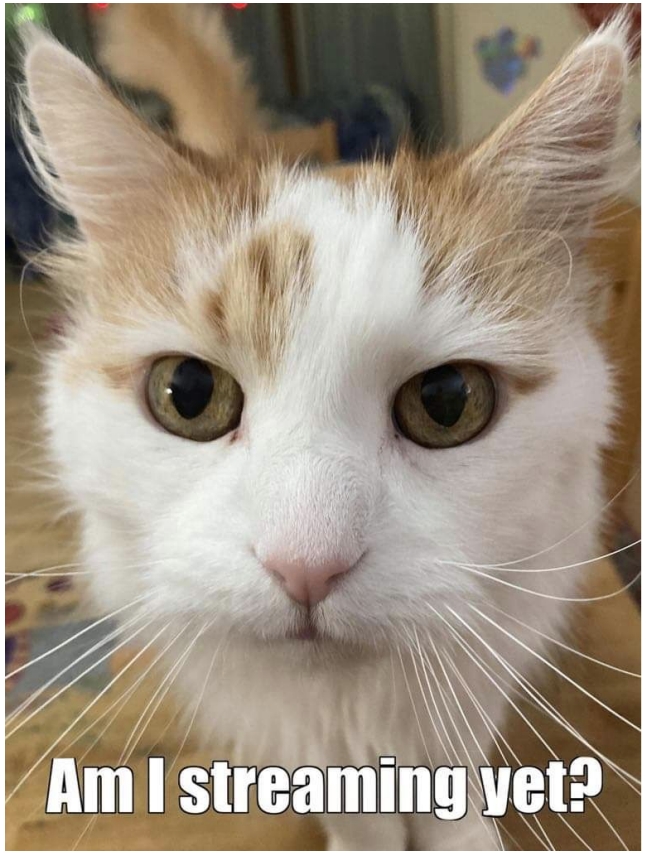
flankstack.dev

Today's Data. REST and Websocket JSON "stonks"

{"symbol":"CLDR",
"uuid":"10640832-f139-4b82-8780-e3ad37b3d0ce",
"ts":1618529574078,
"dt":1612098900000,
"datetime":"2021/01/31 08:15:00",
"open":"12.24500",
"close":"12.25500",
"high":"12.25500",
"volume":"12353",
"low":"12.24500"}

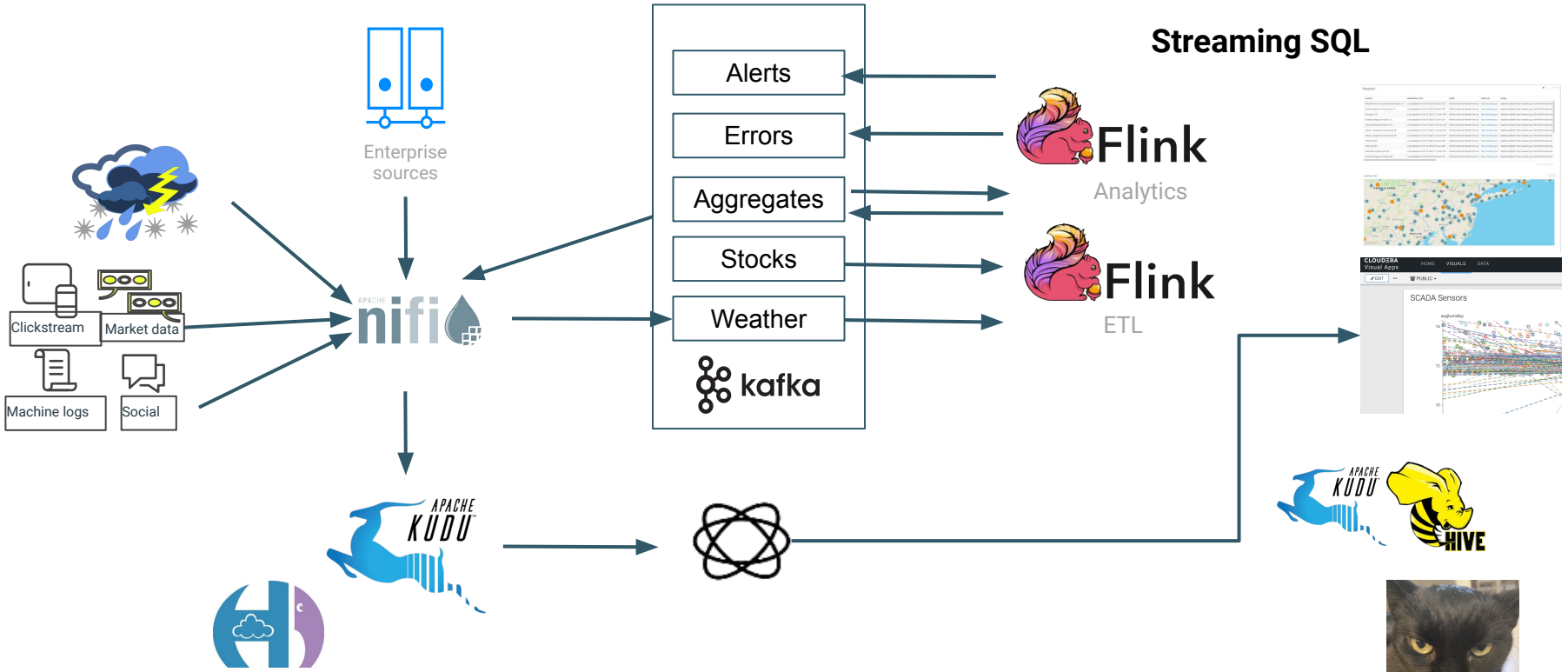
https://github.com/tspannhw/SmartStocks





FLaNK
flankstack.dev

End to End Streaming Demo Pipeline

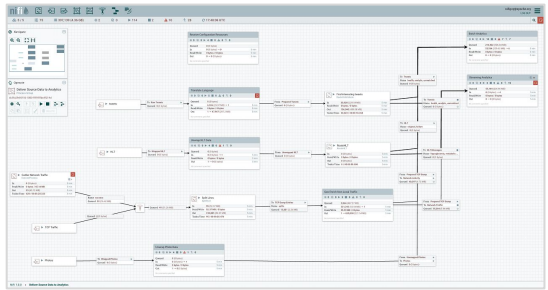


https://github.com/tspannhw/CloudDemo2021


FLANK
NIFI - KAFKA - FLINK

Apache NiFi in a Nutshell

Enable easy ingestion, routing, management and delivery of any data anywhere *(Edge, cloud, data center)* to any downstream system with built in end-to-end security and provenance



APACHE NIFI registry

Advanced tooling to industrialize flow development *(Flow Development Life Cycle)*

ACQUIRE

- FTP
- SFTP
- HL7
- UDP
- XML
- HTTP
- EMAIL
- HTML
- IMAGE
- SYSLOG

PROCESS

- HASH
- MERGE
- EXTRACT
- DUPLICATE
- SPLIT
- ENCRYPT
- TALL
- ROUTE TEXT
- ROUTE CONTENT
- ROUTE CONTEXT
- ROUTE RATE
- DISTRIBUTE LOAD
- EVALUATE
- EXECUTE
- GEOENRICH
- SCAN
- REPLACE
- TRANSLATE
- CONVERT

DELIVER

- FTP
- SFTP
- HL7
- UDP
- XML
- HTTP
- EMAIL
- HTML
- IMAGE
- SYSLOG

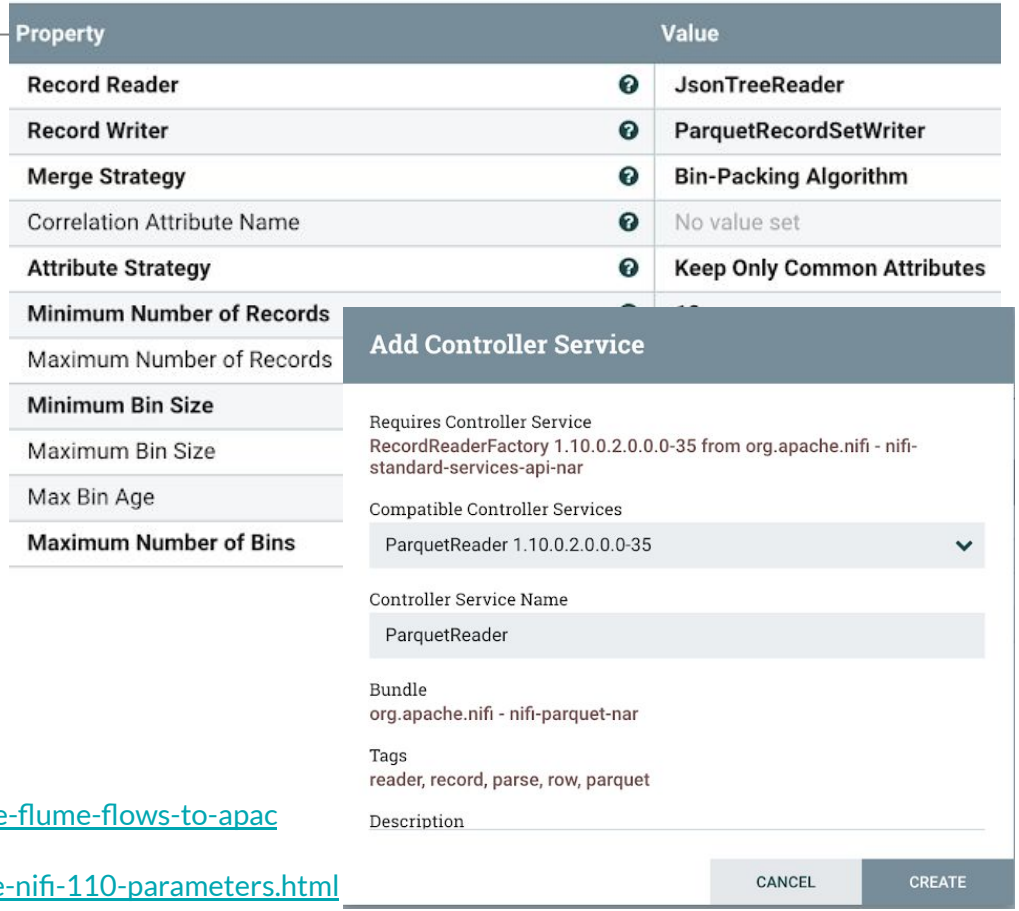
- Over 340 Prebuilt Processors
- Easy to build your own
- Parse, Enrich & Apply Schema
- Filter, Split, Merger & Route
- Throttle & Backpressure

- Guaranteed Delivery
- Full data provenance from acquisition to delivery
- Diverse, Non-Traditional Sources
- Eco-system integration

ParquetReader / ParquetWriter Records

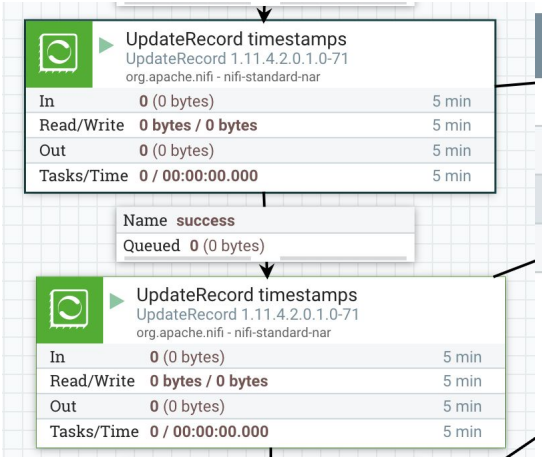
- Native Record Processors for Apache Parquet Files!
- CVS <-> Parquet
- XML <-> Parquet
- AVRO <-> Parquet
- JSON <-> Parquet
- More...

https://www.datainmotion.dev/2019/10/migrating-apache-flume-flows-to-apache_7.html
https://www.datainmotion.dev/2019/11/exploring-apache-nifi-110-parameters.html



UpdateRecord

- Use with LookupRecord
- ELT
- Works on CSV, XML, JSON, AVRO, …
- RecordPath or Literals
- Use Schemas and Schema Registry

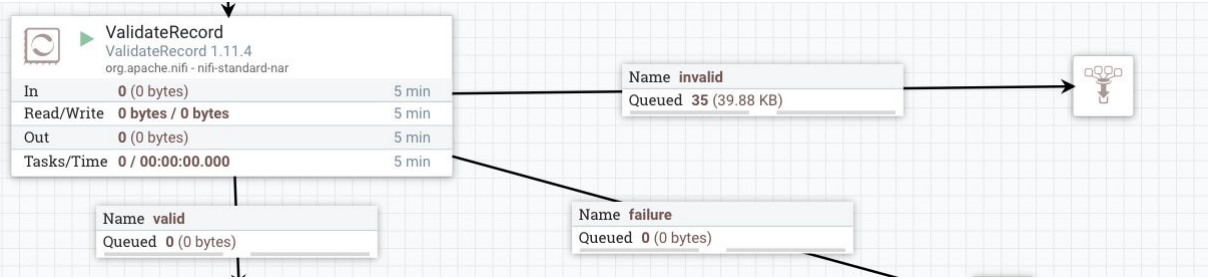


	Value
	SchemaAwareJsonTreeReader
	Standard Inherit JsonRecordSetWriter
	Record Path Value
	format(/datetime,"yyyy/MM/dd HH:mm:SS")

Property	Value
Record Reader	**SchemaAwareJsonTreeReader**
Record Writer	**Standard Inherit JsonRecordSetWriter**
Replacement Value Strategy	**Literal Value**
/datetime	${field.value:toDate("yyyy-mm-dd HH:mm:ss"):toNumber()}
/symbol	${symbol}
/ts	${now():toNumber()}
/uuid	${uuid}

ValidateRecord

- Works on CSV, XML, JSON, AVRO, …
- RecordPath or Literals
- Use Schemas and Schema Registry
- Checks fields, types, nullable



Record Reader	**SchemaAwareJsonTreeReader**
Record Writer	**Standard Inherit JsonRecordSetWriter**
Record Writer for Invalid Records	No value set
Schema Access Strategy	**Use Schema Name Property**
Schema Registry	Edge2ai Cloudera SchemaRegistry CSchema
Schema Name	${schema.name}
Schema Text	${avro.schema}
Allow Extra Fields	**true**
Strict Type Checking	**false**

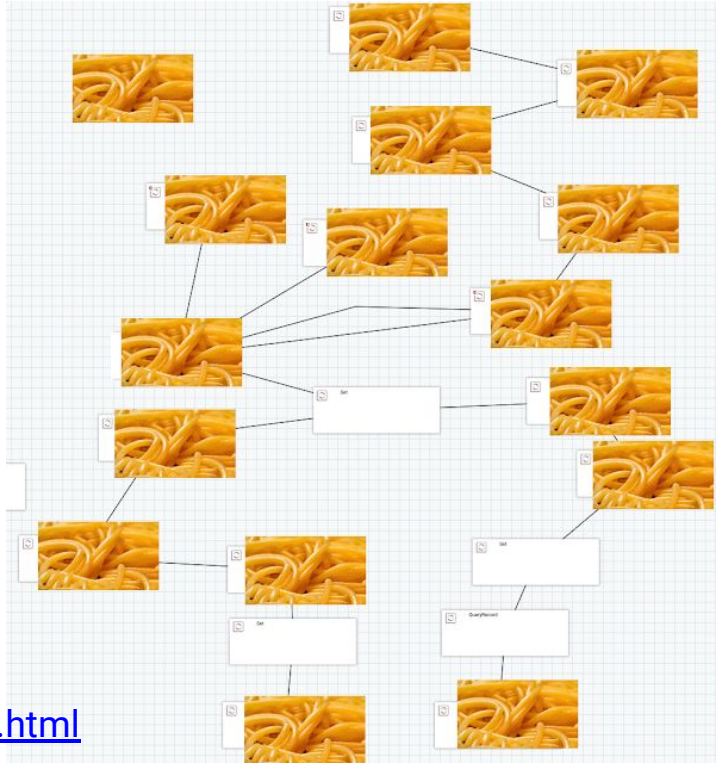
RestLookupService

- Works on CSV, XML, JSON, AVRO, …
- Use Schemas and Schema Registry
- Can call Cloduera ML Models
- SSL and Proxy enabled

Property	Value
URL	**#{model_url}**
Record Reader	**JsonTreeReader**
Record Path	#{sentiment.record.path}
SSL Context Service	No value set
Proxy Configuration Service	No value set
Basic Authentication Username	No value set
Basic Authentication Password	No value set
Use Digest Authentication	false
Connection Timeout	**15 secs**
Read Timeout	**45 secs**
Authorization	Bearer #{model.api.key}

No More Spaghetti Flows

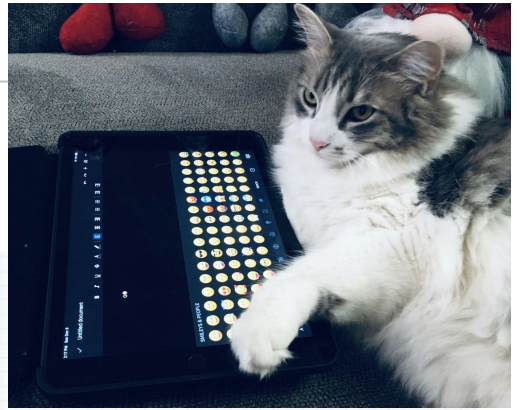
- Reduce, Reuse, Recycle. Use Parameters to reuse common modules.
- Put flows, reusable chunks into separate Process Groups.
- Write custom processors if you need new or specialized features
- Use Cloudera supported NiFi Processors
- Use Record Processors everywhere



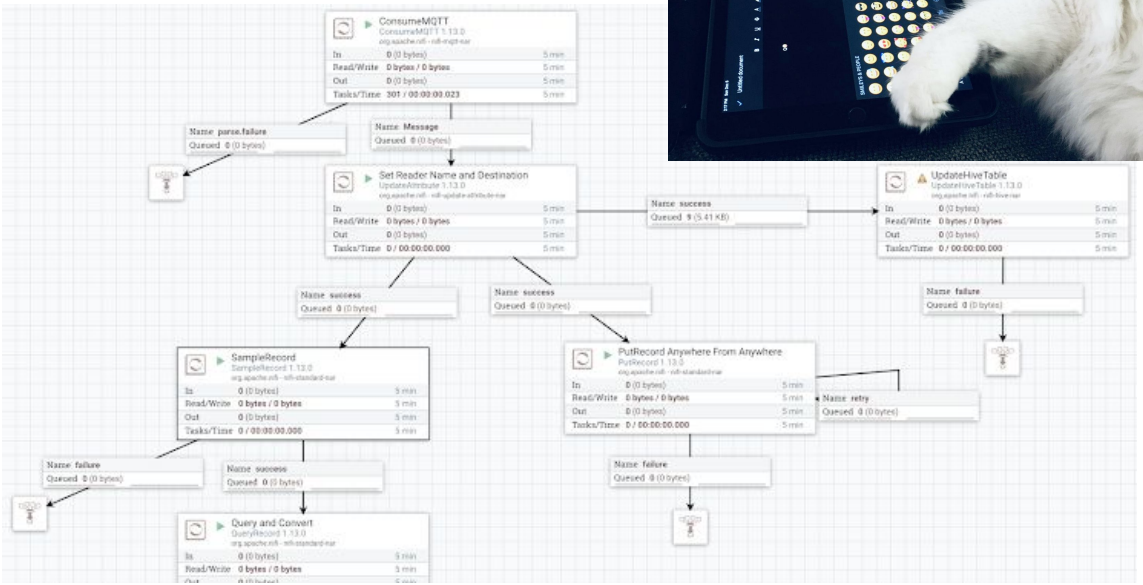
https://www.datainmotion.dev/2020/06/no-more-spaghetti-flows.html

New Features

… based on Apache NiFi 1.13.2



- ListenFTP
- Data Drift
- SampleRecord
- Generic Record Sink
- Generic Record Reader
- PutRecord
- WindowsEventLogReader

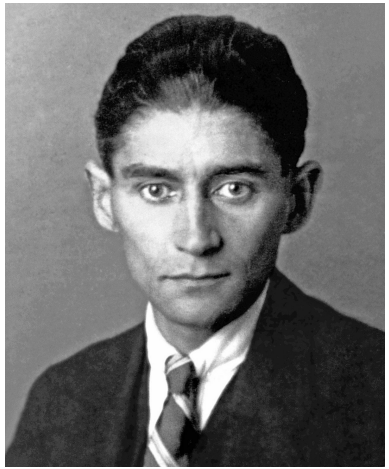


https://www.datainmotion.dev/2021/02/new-features-of-apache-nifi-1130.html

Yes, Franz, It's Kafka

Let's do a metamorphosis on your data. Don't fear changing data.

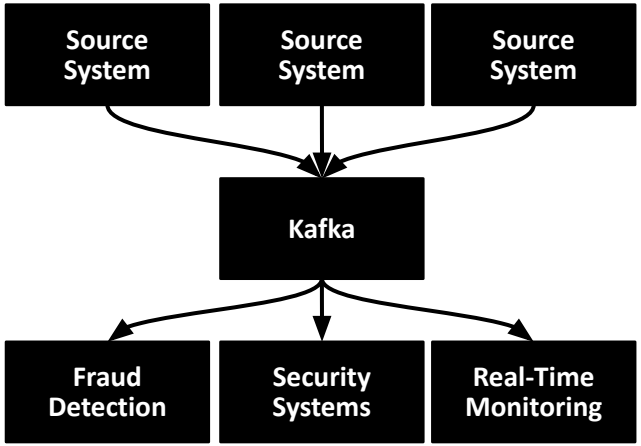
You don't need to be a brilliant writer to stream data.



Franz Kafka was a German-speaking Bohemian novelist and short-story writer, widely regarded as one of the major figures of 20th-century literature. His work fuses elements of realism and the **fantastic**.
Wikipedia

Apache Kafka

- Highly reliable distributed messaging system
- Decouple applications, enables many-to-many patterns
- Publish-Subscribe semantics
- Horizontal scalability
- Efficient implementation to operate at speed with big data volumes
- Organized by topic to support several use cases

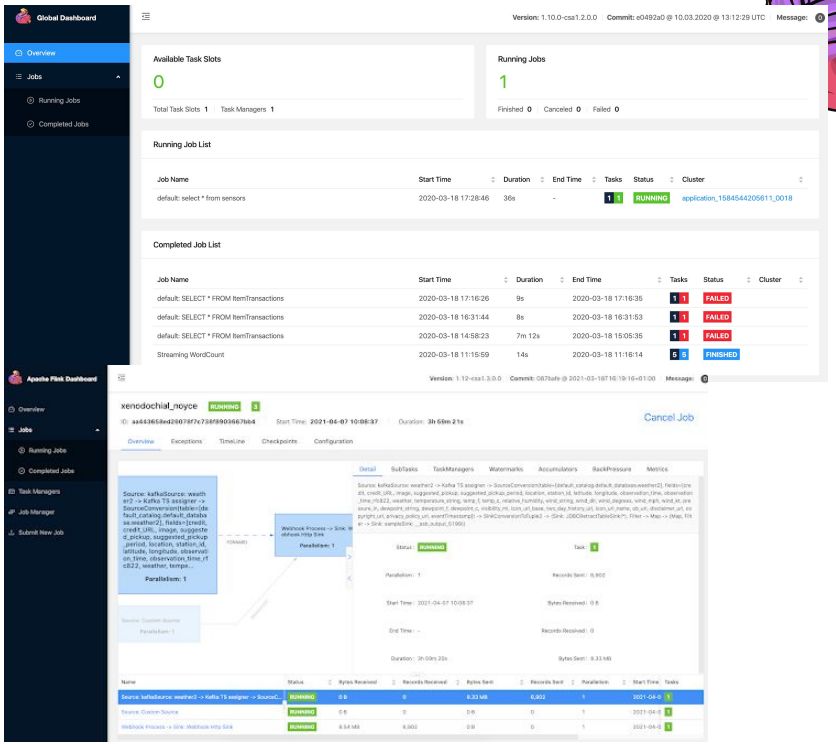


Many-To-Many Publish-Subscribe

Flink SQL

… based on Apache Flink 1.12

- Streaming Analytics
- Continuous SQL
- Continuous ETL
- Complex Event Processing
- Standard SQL Powered by Apache Calcite
- Deployed Apache Flink Apps on YARN
- Scalable Stream Processing



https://www.datainmotion.dev/2021/04/cloudera-sql-stream-builder-ssb-updated.html

Flink SQL

Key Takeaway: Rich SQL grammar with advanced time and aggregation tools

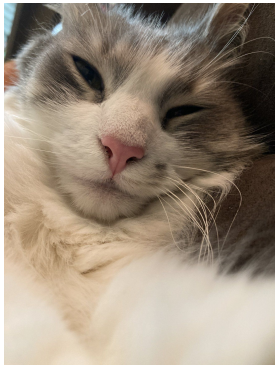
```
-- specify Kafka partition key on output
SELECT foo AS _eventKey FROM sensors

-- use event time timestamp from kafka
-- exactly once compatible
SELECT eventTimestamp FROM sensors

-- nested structures access
SELECT foo.’bar’ FROM table; -- must quote nested
column

-- timestamps
SELECT * FROM payments
WHERE eventTimestamp > CURRENT_TIMESTAMP-interval
'10' second;

-- unnest
SELECT b.*, u.*
FROM bgp_avro b,
UNNEST(b.path) AS u(pathitem)
```

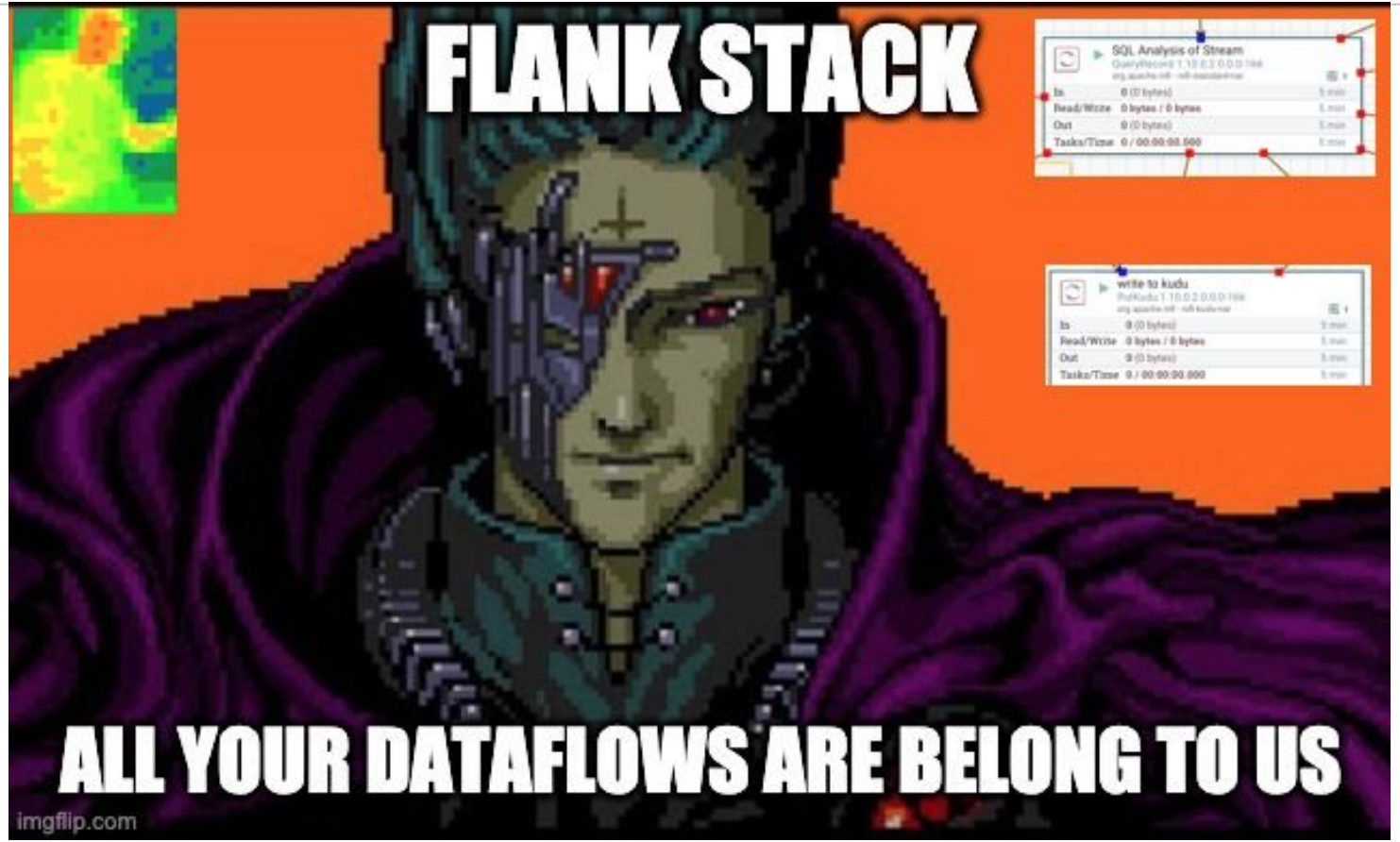
```
-- aggregations and windows
SELECT card,
MAX(amount) as theamount,
TUMBLE_END(eventTimestamp, interval '5' minute) as
ts
FROM payments
WHERE lat IS NOT NULL
AND lon IS NOT NULL
GROUP BY card,
TUMBLE(eventTimestamp, interval '5' minute)
HAVING COUNT(*) > 4  -- >4==fraud

-- try to do this ksql!
SELECT us_west.user_score+ap_south.user_score
FROM kafka_in_zone_us_west us_west
FULL OUTER JOIN kafka_in_zone_ap_south ap_south
ON us_west.user_id = ap_south.user_id;
```

Flink SQL

SELECT location, station_id, latitude, longitude, observation_time, weather, temperature_string, relative_humidity, wind_string, wind_dir, wind_degrees, wind_mph, pressure_in, dewpoint_string, dewpoint_f, dewpoint_c FROM weather2 **WHERE** location is not null and location <> 'null' and trim(location) <> '' and location like '%NJ'

SELECT HOP_END(eventTimestamp, INTERVAL '1' SECOND, **INTERVAL** '30' SECOND) as windowEnd, **count**("close") as closeCount, **sum**(cast("close" as float)) as closeSum, **avg**(cast("close" as float)) as closeAverage, **min**("close") as closeMin, **max**("close") as closeMax, **sum**(case when "close" > 14 then 1 else 0 end) as stockGreaterThan14 **FROM** stocksraw **GROUP BY HOP**(eventTimestamp, INTERVAL '1' SECOND, INTERVAL '30' SECOND)



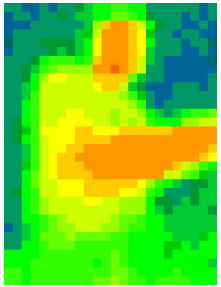


Upcoming Events

April 27

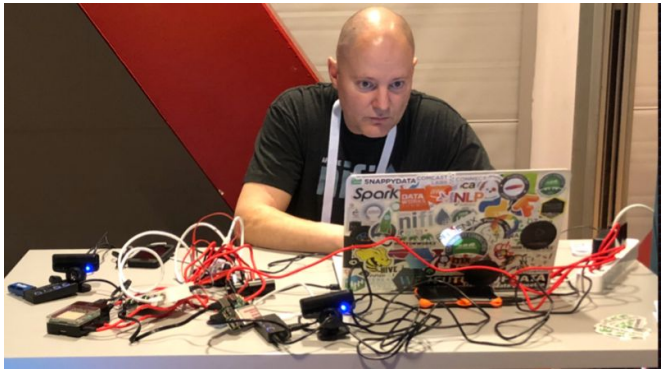
May 19

FLaNK
flankstack.dev




FLaNK
flankstack.dev
POWER OF DATA
HORTONWORKS
CLOUDERA
HIVE
Apache Flink

THANK YOU


SNAPPYDATA
Spark
NLP