



Streaming Data Pipeline Development

Tim Spann
Principal Developer Advocate

25-April-2023



CLOUDERA



CLOUDERA



EDGE
2AI

CLOUDERA





Meet the NiFi Committers

Wednesday, May 3, 2023
10am-11amPT/12pm-1pmCT/1pm-2pmET



Hosts



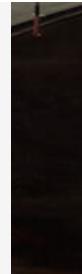
Joe Witt, VP of Engineering, Cloudera



Mark Payne, Principal Engineer, Cloudera



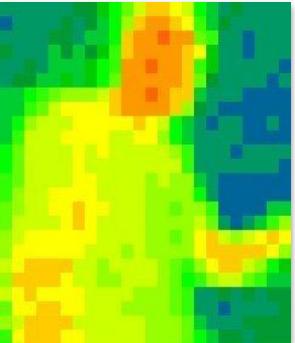
Matt Gilman, Principal Engineer, Cloudera



<https://attend.cloudera.com/nificommittees0503>

© 2023 Cloudera, Inc. All rights reserved.

FLaNK Stack



Tim Spann

@PaasDev // Blog: www.datainmotion.dev

Principal Developer Advocate.

Princeton Future of Data Meetup.

ex-Pivotal, ex-Hortonworks, ex-StreamNative, ex-PwC

<https://medium.com/@tspann>

<https://github.com/tspannhw>

Apache NiFi x Apache Kafka x Apache Flink x Java





Tim Spann
Principal Developer Advocate | Cloudera



Write the reasons someone should attend my meetup on Streaming Data Pipeline Development with Apache NiFi, Apache Kafka and Apache Flink



[View other drafts](#)



Here are some reasons why someone should attend your meetup on Streaming Data Pipeline Development with Apache NiFi, Apache Kafka and Apache Flink:

- Learn from experts in the field. The meetup will feature presentations from experts in Apache NiFi, Apache Kafka, and Apache Flink.
- Get hands-on experience. The meetup will also include hands-on workshops where you can learn how to use these technologies.
- Network with other developers. The meetup is a great opportunity to network with other developers who are interested in streaming data pipeline development.
- Stay up-to-date on the latest technologies. The meetup will cover the latest developments in Apache NiFi, Apache Kafka, and Apache Flink.
- Have fun! The meetup is a great opportunity to learn new things and network with other developers.



Google it



FLiP Stack Weekly



<https://bit.ly/32dAJft>



This week in Apache NiFi, Apache Flink, Apache Kafka, Apache Spark, Apache Iceberg, Python, Java and Open Source friends.

Future of Data - Princeton + Virtual



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

From Big Data to AI to Streaming to Containers to Cloud to Analytics to Cloud Storage to Fast Data to Machine Learning to Microservices to ...



@PaasDev

FREE LEARNING ENVIRONMENT

CSP Community Edition

- Kafka, KConnect, SMM, SR, Flink, and SSB in Docker
- Runs in Docker
- Try new features quickly
- Develop applications locally



- Docker compose file of CSP to run from command line w/o any dependencies, including Flink, SQL Stream Builder, Kafka, Kafka Connect, Streams Messaging Manager and Schema Registry
 - \$> docker compose up
- Licensed under the Cloudera Community License
- **Unsupported**
- Community Group Hub for CSP
- Find it on docs.cloudera.com under Applications

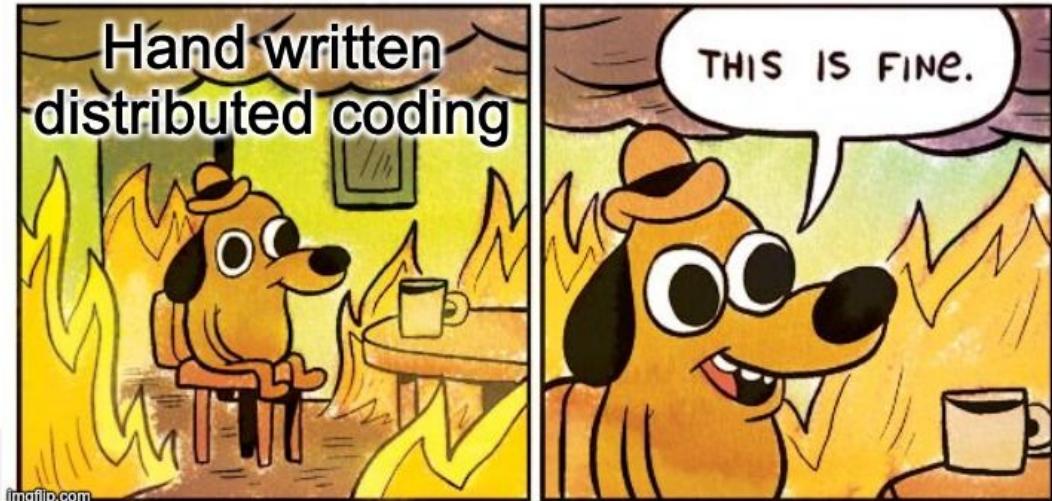


CSP Community Edition

A readily available, dockerized deployment of Apache Kafka and Apache Flink that allows you to test the features and capabilities of Cloudera Stream Processing.

[Learn More](#)

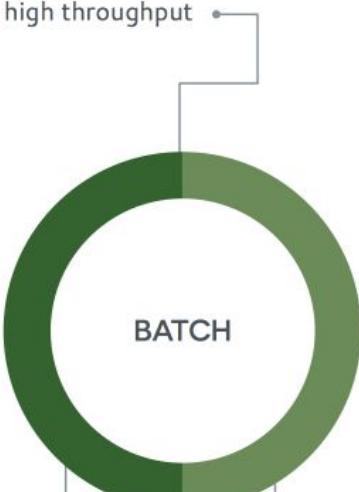
STREAMING



WHAT IS REAL-TIME?

> 1 HOUR

high throughput



10 MS – 1 SEC

approximate



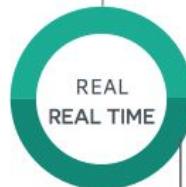
< 500 MS

latency sensitive



< 1 MS

low latency



adhoc queries

monthly active users relevance for ads

ad impressions count hash tag trends

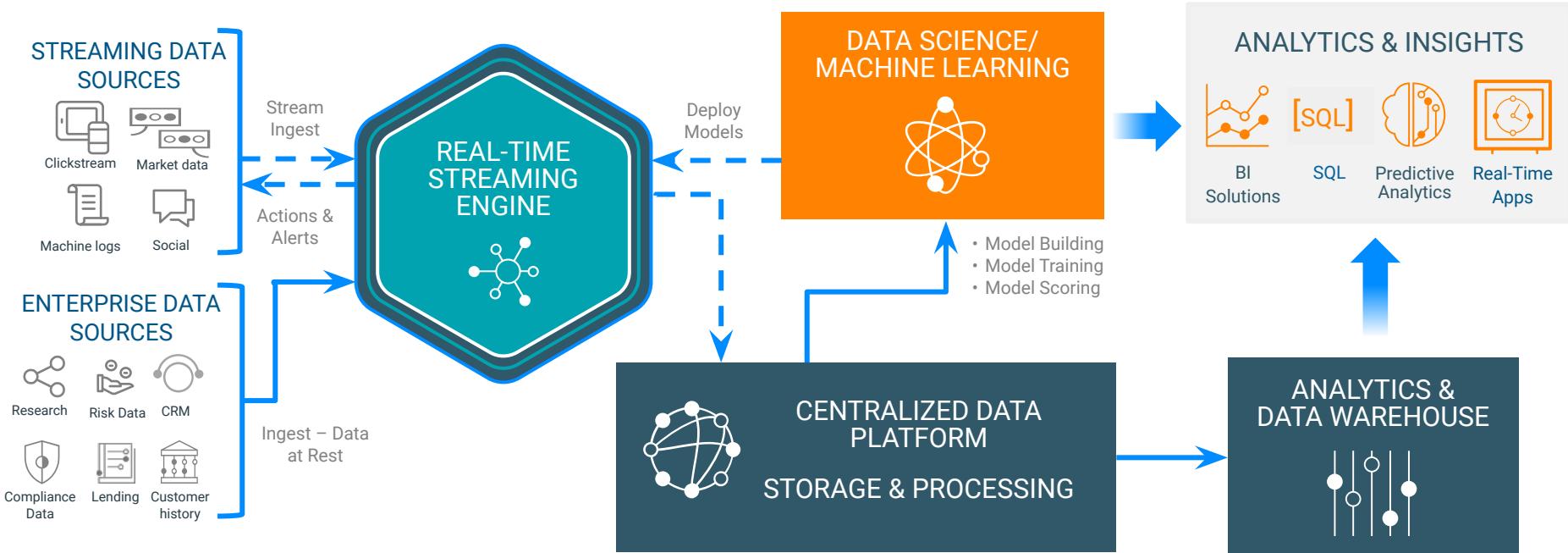
deterministic workflows

fanout Tweets search for Tweets

Financial Trading

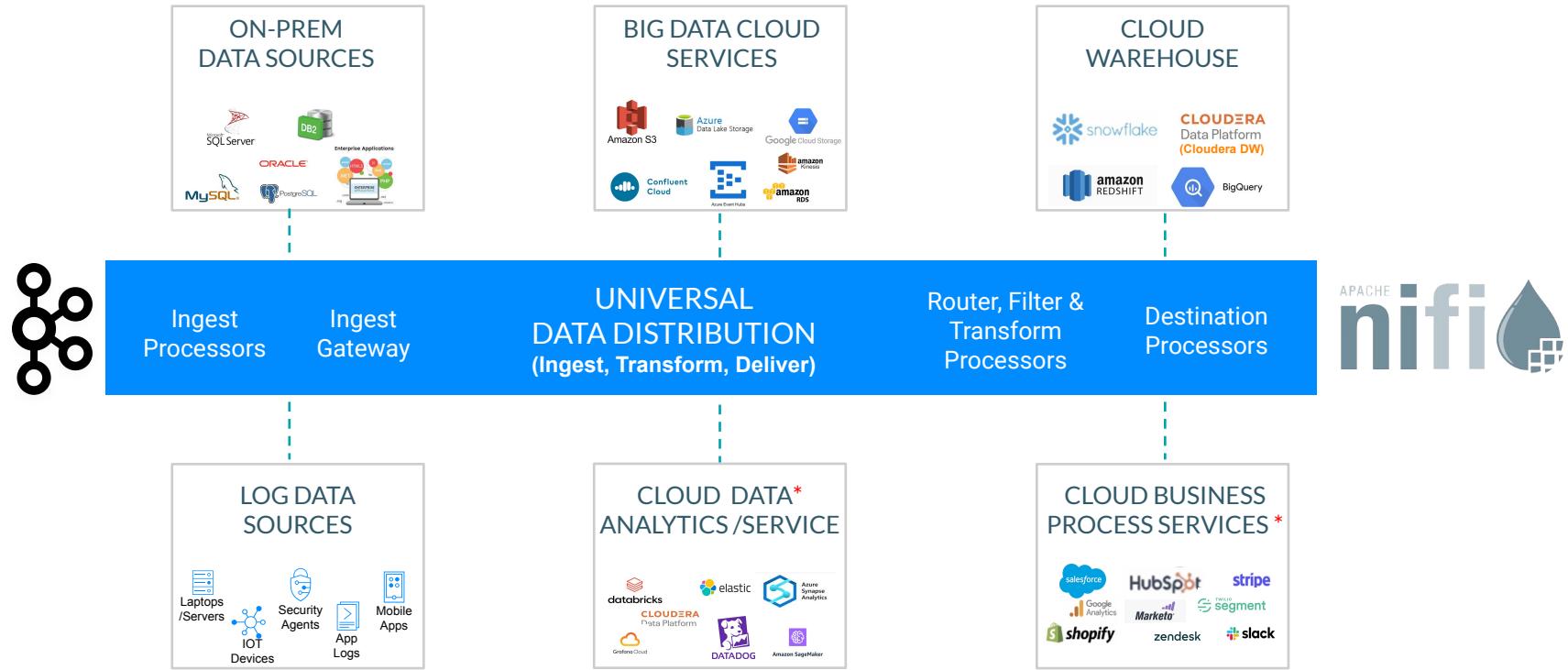
ENABLING ANALYTICS AND INSIGHTS ANYWHERE

Driving enterprise business value



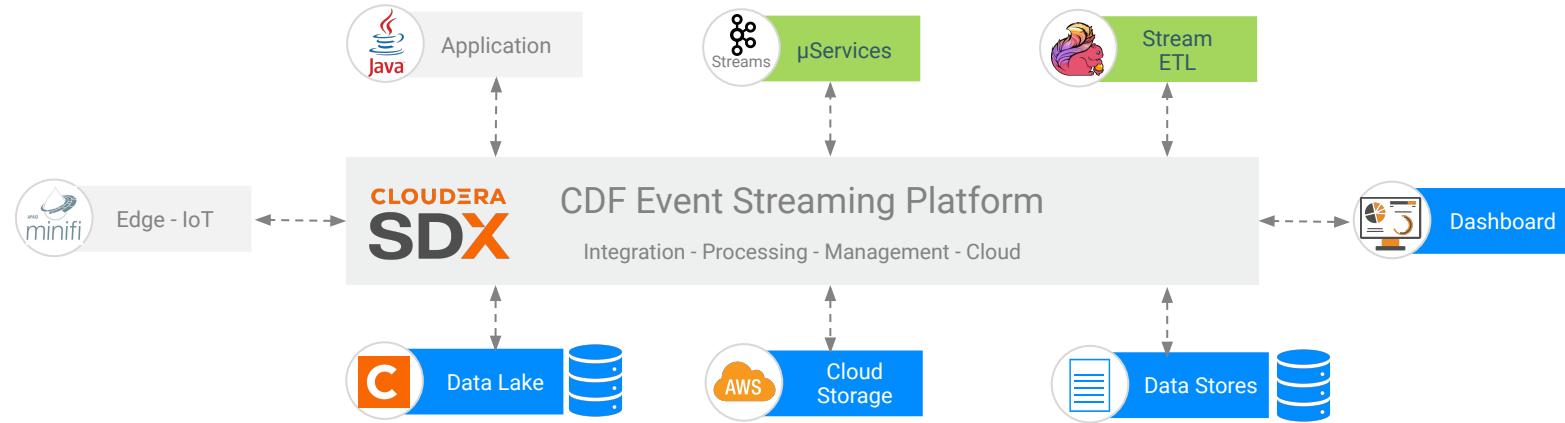
STREAMING FROM ... TO .. WHILE ..

Data distribution as a first class citizen



EVENT-DRIVEN ORGANIZATION

Modernize your data and applications

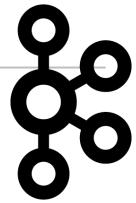


BUILDING REAL-TIME REQUIRES A TEAM



APACHE KAFKA





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](#)

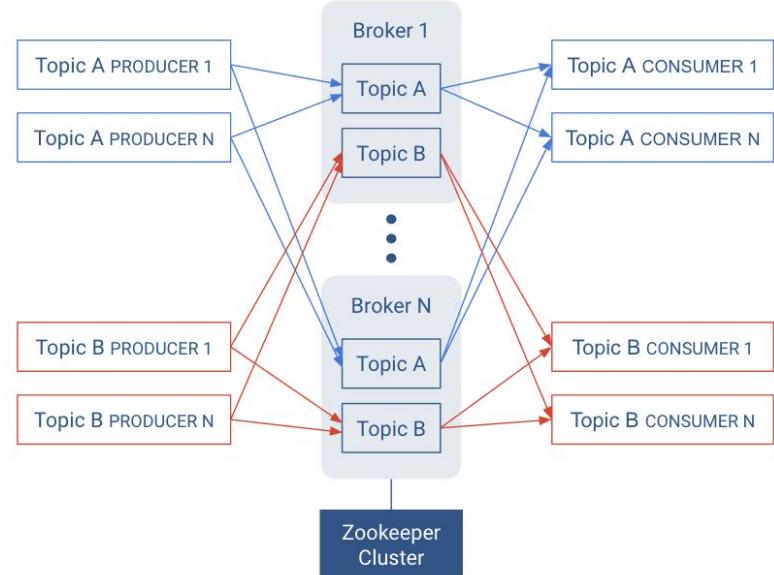


STREAMS MESSAGING WITH KAFKA



| WriteToKafka | | |
|---|-------------------|-------|
| PublishKafka2RecordCDP 1.0.0.2.2.2.0-127 com.cloudera - nifi-cdf-kafka-2-nar | | |
| In | 0 (0 bytes) | 5 min |
| Read/Write | 0 bytes / 0 bytes | 5 min |
| Out | 0 (0 bytes) | 5 min |
| Tasks/Time | 0 / 00:00:00.000 | 5 min |

- 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.



What is Apache Kafka?

- **Distributed:** horizontally scalable
- **Partitioned:** the data is split-up and distributed across the brokers
- **Replicated:** allows for automatic failover
- **Unique:** Kafka does not track the consumption of messages (the consumers do)
- **Fast:** designed from the ground up with a focus on performance and throughput
- Kafka was built at LinkedIn in 2011
- Open sourced as an Apache project

What is Can You Do With Apache Kafka?

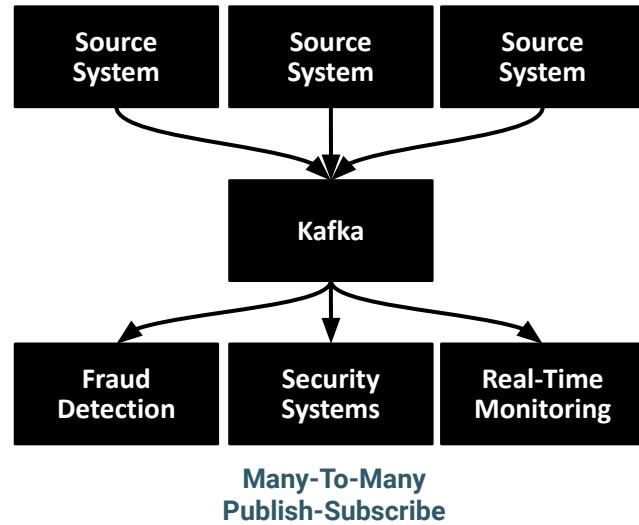
- Web site activity: track page views, searches, etc. in real time
- Events & log aggregation: particularly in distributed systems where messages come from multiple sources
- Monitoring and metrics: aggregate statistics from distributed applications and build a dashboard application
- Stream processing: process raw data, clean it up, and forward it on to another topic or messaging system
- Real-time data ingestion: fast processing of a very large volume of messages

KAFKA TERMINOLOGY

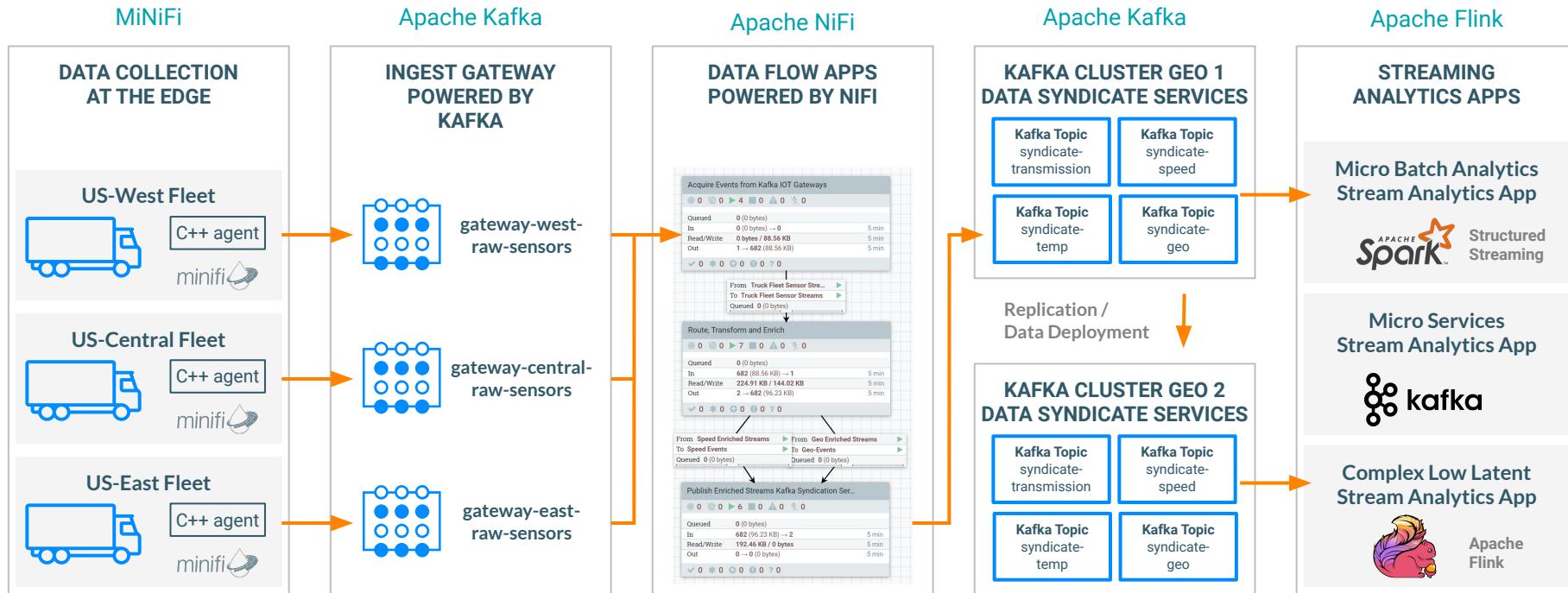
- Kafka is a publish/subscribe messaging system comprised of the following components:
 - **Topic**: a message feed
 - **Producer**: a process that publishes messages to a topic
 - **Consumer**: a process that subscribes to a topic and processes its messages
 - **Broker**: a server in a Kafka cluster

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



Apache Kafka



APACHE FLINK



Am I streaming yet?

Flink SQL



- Streaming Analytics
- Continuous SQL
- Continuous ETL
- Complex Event Processing
- Standard SQL Powered by Apache Calcite

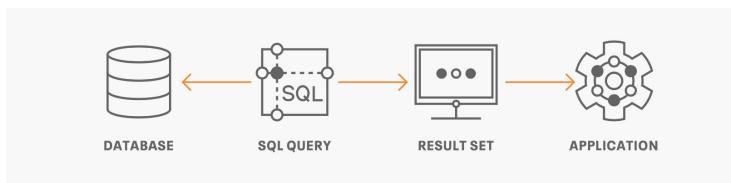
The screenshot shows the Apache Flink Dashboard interface. At the top, there's a navigation bar with links for Overview, Job, Task Managers, Job Manager, and Submit New Job. The main area is focused on a single job named "xenodochial_noxye" which is currently "RUNNING". The job ID is aa4436584ed28078f7c758f8903667b84. It was started at 2021-04-07 10:08:37 and has been running for 3h 56m 2.1s. Below the job details, there are tabs for Overview, Exceptions, Timeline, Checkpoints, and Configuration. The Overview tab is selected. It shows a complex data flow graph with nodes like "Source: kafkaSource: weather2" and "Sink: Webhook Process->Sink". The graph includes various operators such as Map, Filter, and Reduce. The task manager section shows a table with columns for Name, Status, Bytes Received, Records Received, Bytes Sent, Records Sent, Parallelism, and Start Time. One task, "Source: kafkaSource: weather2 -> Kafka TS assigner -> SourceConversionTablet", is listed with a status of "RUNNING". The overall version of the dashboard is 1.12-csa1.3.0.0.

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

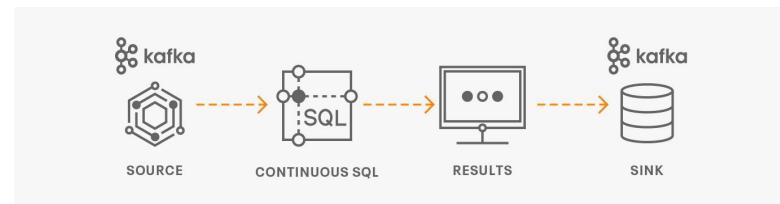
CONTINUOUS SQL

- SSB is a Continuous SQL engine
- It's SQL, but a slightly different mental model, but with big implications

Traditional Parse/Execute/Fetch model



Continuous SQL Model



Hint: The query is boundless and never finishes, and time matters

AKA: `SELECT * FROM foo WHERE 1=0 -- will run forever`

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;
```

CLOUDERA SQL STREAM BUILDER

Making Streaming Analytics accessible to everyone with SQL



Application Developer

- Develop & test SQL queries with a powerful UI
- Expose streaming data to applications through materialized views
- Single button “Push to production” turns SQL queries into Flink application



Business Analyst

- Explore Streaming Data using SQL without learning new skills
- Build new real-time business reporting applications

The screenshot shows the Cloudera Manager interface with the following details:

- Header:** Cloudera Manager, Clusters, Hosts, Diagnostics, Audits, Charts, Backup, Administration.
- Breadcrumbs:** CDFClusterAmsterdam / Flink.
- Actions:** Actions, 30 minutes preceding \$.
- Submenu:** Status, Instances, Configuration, Commands, Charts Library, Audits, SQLStreamBuilder, Quick Lin.
- SQL Tab:** Materialized View
- Code Preview:** A large code editor window displays the following SQL query:

```
1 | SELECT TUMBLE_END(geo_events.eventTimestamp, INTERVAL '3' MINUTE) as windowEnd,
2 | geo_events.driverId,geo_events.driverName,geo_events.route,
3 | avg(speed_events.speed) as driverAvgSpeed
4 | FROM
5 | geo_events,
6 | speed_events
7 | WHERE
8 | geo_events.driverId = speed_events.driverId AND
9 | geo_events.eventTimestamp BETWEEN
10 | speed_events.eventTimestamp - INTERVAL '1' SECOND AND
11 | speed_events.eventTimestamp + INTERVAL '1' SECOND
12 | GROUP BY
13 | TUMBLE(geo_events.eventTimestamp, INTERVAL '3' MINUTE),
14 | geo_events.driverId,
```

- Buttons:** default mode, solarized dark, Sample, Stop, Restart.
- Log Panel:** Shows log messages from the StreamBuilder job:

```
[9/4/2020, 3:35:12 PM][INFO] No persistent sink specified, using ephemeral sink.
[9/4/2020, 3:35:12 PM][INFO] StreamBuilder job Speeding Drivers Over 3 Minute Window is starting.
[9/4/2020, 3:35:25 PM][INFO] SSB version 8.0.4 selected for job.
[9/4/2020, 3:35:25 PM][INFO] Streaming job is now running in the background. You can safely navigate to other pages now, and re-visit the running job by clicking the SQL Jobs tab.
[9/4/2020, 3:35:25 PM][INFO] Stream sampler is running, and will display the next 100 messages matching your query.
[9/4/2020, 3:35:25 PM][INFO] Waiting for messages from stream.
[9/4/2020, 3:36:58 PM][INFO] Stopping job Speeding Drivers Over 3 Minute Window with job ID 4582
[9/4/2020, 3:37:00 PM][INFO] Job Speeding Drivers Over 3 Minute Window is stopped.
[9/4/2020, 3:37:17 PM][INFO] StreamBuilder job Speeding Drivers Over 3 Minute Window is starting.
```

SQL STREAM BUILDER (SSB)

Democratize access to real-time data with just SQL

SQL STREAM BUILDER allows developers, analysts, and data scientists to **write streaming applications** with industry standard **SQL**.

No Java or Scala code development required.

Simplifies access to data in Kafka & Flink. Connectors to batch data in HDFS, Kudu, Hive, S3, JDBC, CDC and more

Enrich streaming data with batch data in a single tool

```
CREATE TABLE `kafka_table_1670513700` (
  `col_str` STRING,
  `col_int` INT,
  `col_ts` TIMESTAMP(3),
  WATERMARK FOR `col_ts` AS col_ts - INTERVAL '5' SECOND
) WITH (
  'connector' = 'kafka', -- Specify what connector to use, for Kafka it must use 'kafka'.
  'format' = 'json', -- Topic name to read from.
  'topic' = 'yourTopicName', -- Comma separated list of Kafka brokers.
  'properties.bootstrap.servers' = '...', -- Optional flag to specify whether to encode all decimals as plain numbers instead of
  Note, only one of 'topic-pattern' and 'topic' can be specified for sources. When the table is used as sink, the topic name is the topic to write
  to. Note topic list is not supported for sinks.
  'json.decimals-as-plain-numbers' = 'false' -- Optional flag to parse integers as plain numbers by default.
  'parse-as-records' = 'true' -- Optional flag to parse records by default.
  'json.fail-on-missing-field' = 'false' -- Optional flag to fail if a field is missing or not, false by default.
  'json.ignore-errors' = 'false' -- Optional flag to skip fields and rows with parse errors instead of failing; fields are set to null in
  case of errors, false by default.
  'json.map-null-key.literal' = 'null' -- Optional flag to specify string literal for null keys when 'map-null-key.mode' is LITERAL, '\"null\"'
  by default.
  'map-null-key.mode' = 'FAIL' -- Optional flag to control the handling mode when serializing null key for map data, FAIL by default.
  Option DROP will drop null key entries for map data. Option LITERAL will use 'map-null-key.literal' as key literal.
)
```

Logs Results Events

SCHEMA

Key Takeaway: Integrated with schema registry, also auto-detection for JSON types.

- AVRO - Schema Registry
- JSON - Schema Auto-detect
- Virtual Table design pattern
- Kafka Data Source
auto-created in SSB

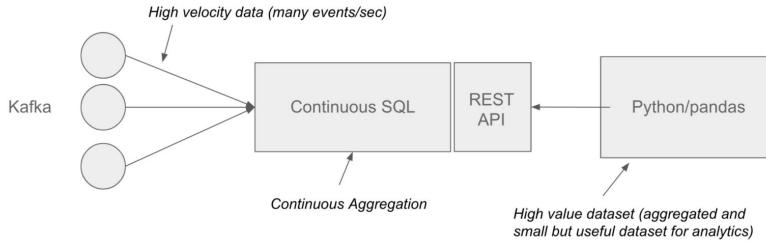
Detect Schema



```
{  
  "fields": [  
    {  
      "doc": "Type inferred from '215'",  
      "name": "userid",  
      "type": "long"  
    },  
    {  
      "doc": "Type inferred from '94204'",  
      "name": "amount",  
      "type": "long"  
    }  
  "name": "inferredSchema",  
  "type": "record"  
}
```

SSB MATERIALIZED VIEWS

Key Takeaway; MV's allow data scientist, analyst and developers consume data from the firehose

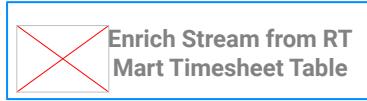
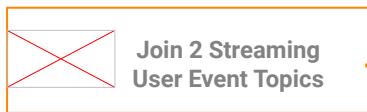


```
SELECT userid,
       max(amount) as max_amount,
       sum(amount) as sum_amount,
       count(*) as thecount,
       tumble_end(eventTimestamp, interval '5' second) as ts
  FROM authorizations
 GROUP BY userid, tumble(eventTimestamp, interval '5' second)
 HAVING count(*) > 1
```



```
[90]: import pandas as pd
[91]: mv = "https://xxxxxxxxxx"
[92]: df = pd.read_json(mv)
[93]: len(df.keys())
[93]: 5
[95]: df['ts'] = pd.to_datetime(df['ts'])
[97]: df.dtypes
[97]: max_amount          int64
       sum_amount          int64
       thecount            int64
       ts                  datetime64[ns]
       userid              int64
       dtype: object
[98]: df.set_index('userid').sort_values(by=['thecount'], ascending=False).head()
[98]:
      max_amount  sum_amount  thecount      ts
userid
    787      34911     57304     10 2020-06-16 19:52:15
    744      77407     95407      9 2020-06-16 19:52:15
    78      88761     330397      9 2020-06-16 19:52:15
    541      78762     282682      8 2020-06-16 19:52:15
    926      85636     129728      8 2020-06-16 19:52:15
```

Streaming ETL Data Pipeline Made Simple with SQL StreamBuilder

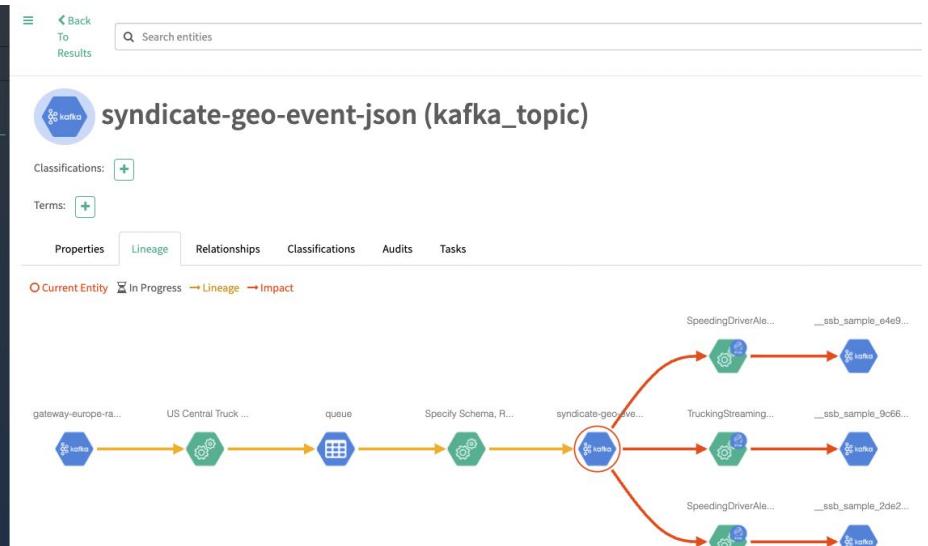
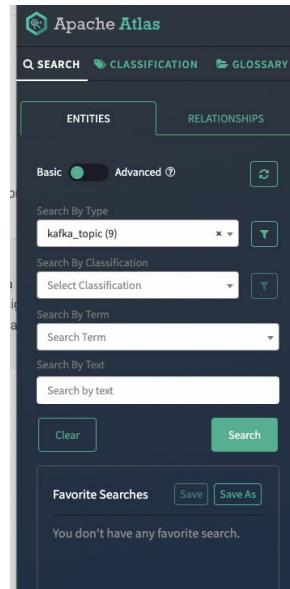


```
1 SELECT
2     geo_event.eventTimestamp, geo_event.driverId, geo_event.eventTime, geo_event.eventSource,
3     geo_event.truckId, geo_event.driverName, geo_event.routeId, geo_event.route, geo_event.eventType,
4     geo_event.latitude, geo_event.longitude, geo_event.correlationId, geo_event.geoAddress,
5     speed_event.speed,
6     driver.certified, driver.wage_plan,
7     timesheet.hours_logged, timesheet.miles_logged
8
9 FROM
10    geo_events_json AS geo_event
11    JOIN speed_events_json AS speed_event
12    ON (geo_event.driverId = speed_event.driverId)
13    LEFT JOIN CDP_Hive_Catalog.employees_hr_hive_db.driver
14    FOR SYSTEM_TIME AS OF PROCTIME() driver
15    ON driver.driverid = geo_event.driverId
16    LEFT JOIN `CDP_Kudu_Catalog`.`default_database`.`impala::employees_hr_kudu_impala_db.timesheet`
17    FOR SYSTEM_TIME AS OF PROCTIME() timesheet
18    ON (timesheet.driverid = geo_event.driverId AND timesheet.week = 1)
19    WHERE
20        geo_event.eventTimestamp BETWEEN
21            speed_event.eventTimestamp - INTERVAL '1' SECOND AND
22            speed_event.eventTimestamp + INTERVAL '1' SECOND
23        AND geo_event.eventType <> 'Normal'
24        AND driver.wage_plan = 'hours'
25        AND timesheet.hours_logged > .45
```

DATA GOVERNANCE FOR THE ENTIRE STREAMING PIPELINE

Streaming Data Lineage with SDX

- Track Consumer, Producer, Topics and Consumer Group Lineage
- No changes required to Consumers or Producers
- End-To-End lineage from consumer to producer



SSB Projects - Container Structure for All Assets of SQL Streaming Job

SDLC for Streaming SQL Applications With First Class Git Integration

Project in SSB

Projects / Trucking Streaming IOT App

Search in SSB

Explorer

Trucking Streaming IOT App

Resource Total Details

- Jobs 3 3 STOPPED
- Virtual Tables 3 kafka
- Functions 0
- Kafka Data Providers 1
- Catalogs 1 Schema Registry
- Materialized Views 1 ConnectionDriverAlertsToKafka
- API Keys 1
- Job Notifications 1 vett_speeding_drivers

Clone URL https://github.com/georgeveticaden/ssb-project-1.git

Branch main

Allow deletions on import

Authentication

Method BASIC

Username georgeveticaden

Password <UNCHANGED>

Modify

SSB Project provides the container structure for all the assets for your streaming app. Project is configured with a git repository

Projects / Trucking Streaming IOT App

Commit Message

commit message: new jobs for calculating speeding drivers, new virtual tables, materialized views and job notifications

Push

SSB allows you to push/import projects to/from Git

Project Represented In Git

georgeveticaden / ssb-project-1 Public

Code Issues Pull requests Actions Projects Wiki Security

main ssb-project-1 / Trucking Streaming IOT App / Go to file Add file ...

commit message: new jobs for calculating speeding drivers, new virtual tables, materialized views and job notifications 1 hour ago

..

api_keys new jobs for calculating speeding drivers, new virtual tables, materialized views and job notifications 1 hour ago

data_sources new jobs for calculating speeding drivers, new virtual tables, materialized views and job notifications 1 hour ago

jobs new jobs for calculating speeding drivers, new virtual tables, materialized views and job notifications 1 hour ago

tables new jobs for calculating speeding drivers, new virtual tables, materialized views and job notifications 1 hour ago

main ssb-project-1 / Trucking Streaming IOT App / jobs / Go to file Add file ...

csso_gveticaden and SQL Stream Builder new jobs for calculating speeding drivers, new virtual tables, materialized views and job notifications 1 hour ago

..

SpeedingDriverAlertsToKafka.json new jobs for calculating speeding drivers, new virtual tables, materialized views and job notifications 1 hour ago

SpeedingDriverAlertsToMv.json new jobs for calculating speeding drivers, new virtual tables, materialized views and job notifications 1 hour ago

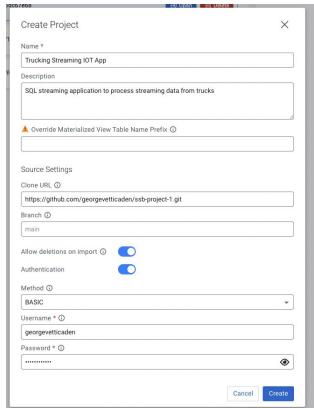
TestTruckingStreamingAnalytics.json new jobs for calculating speeding drivers, new virtual tables, materialized views and job notifications 1 hour ago

The streaming application assets in git within the project structure

SDLC Life Cycle with SSB Projects

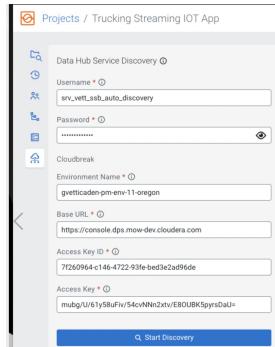
Step 1

Create SSB Project & Configure Git Repo



Step 2

Run Service Discovery to register Kafka, Hive, etc

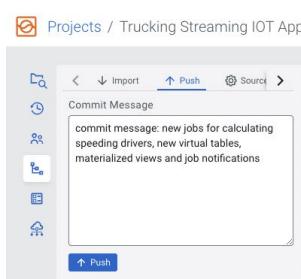


Step 3

Create/Develop Streaming Assets & Test

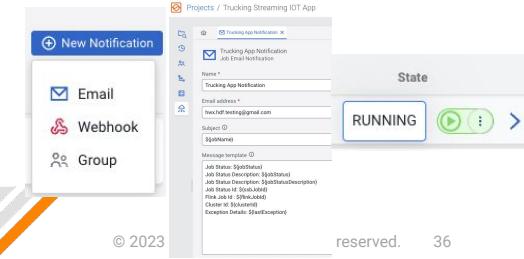
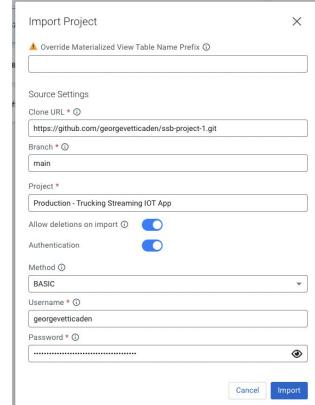
Step 4

Check-in Project Into Git

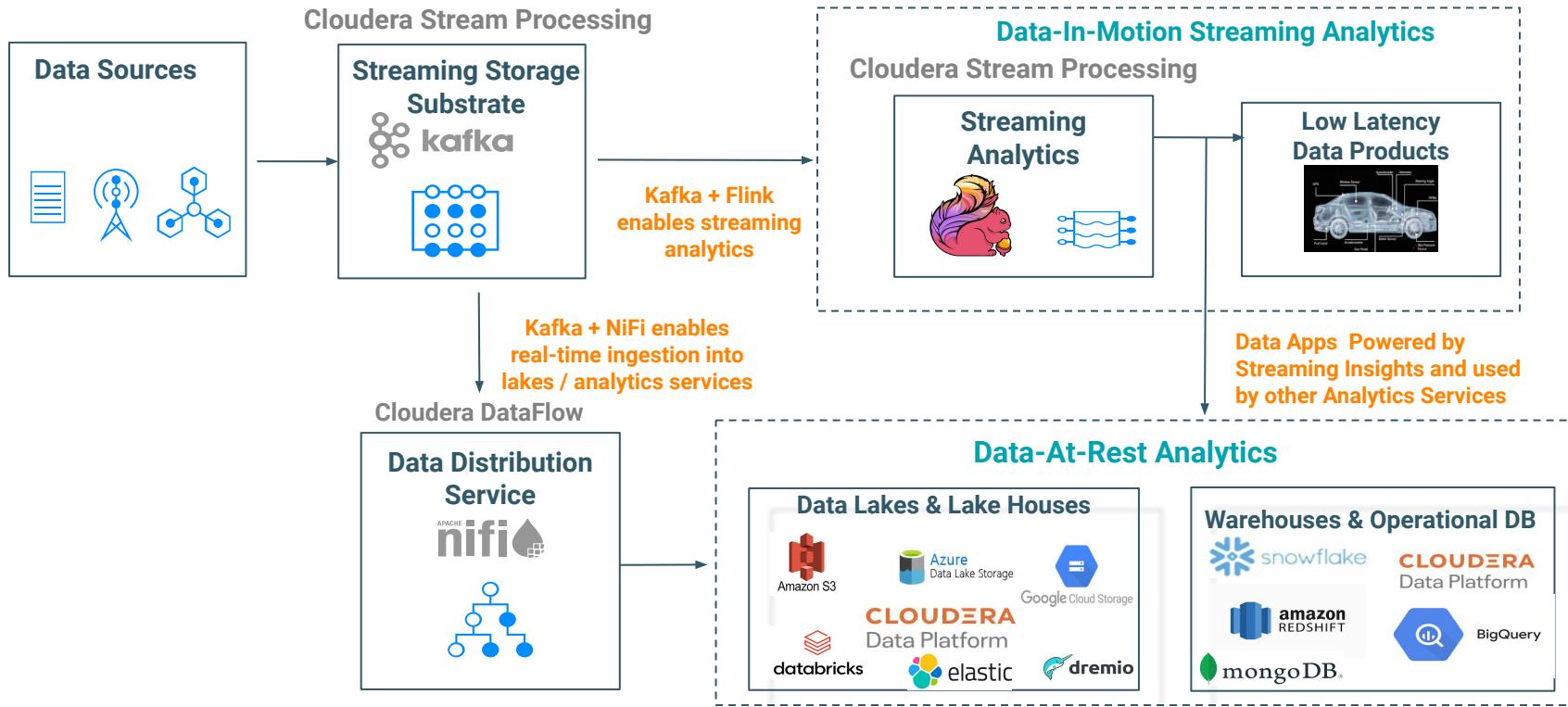


Step 5

Import Project from Git into SSB Prod, Setup Monitoring & Deploy



Moving Beyond Draining of Streams Into Lakes: Analytics-in-Stream

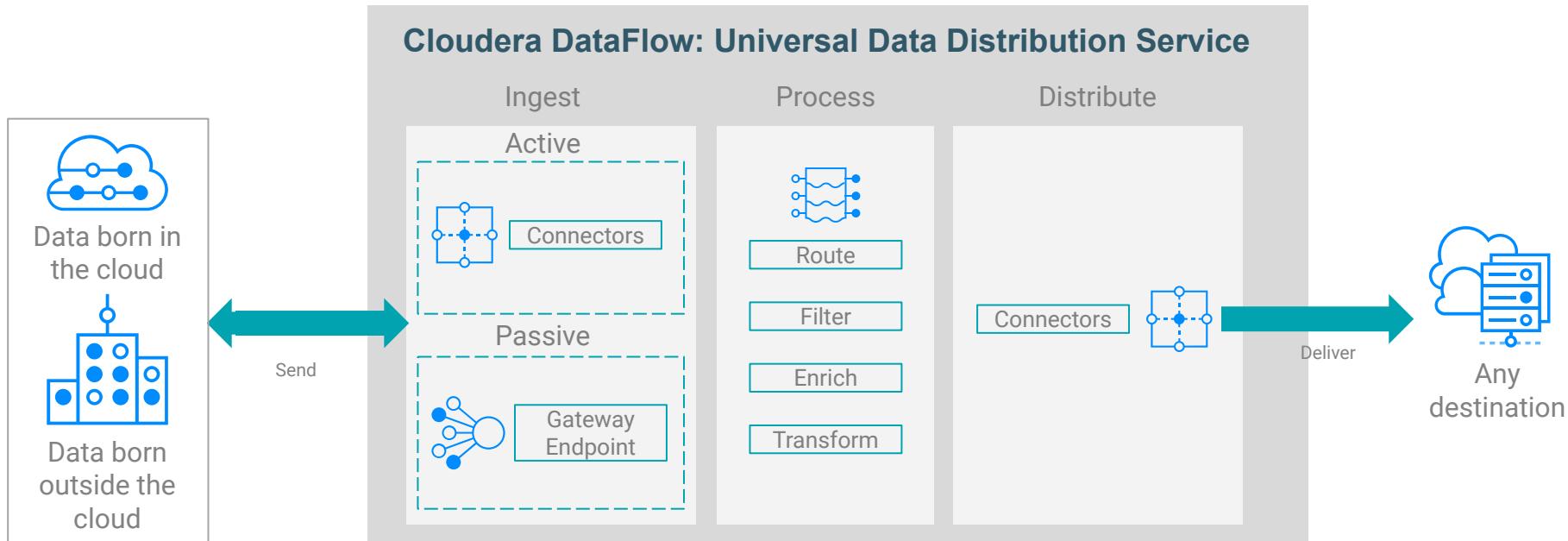


DATAFLOW APACHE NIFI



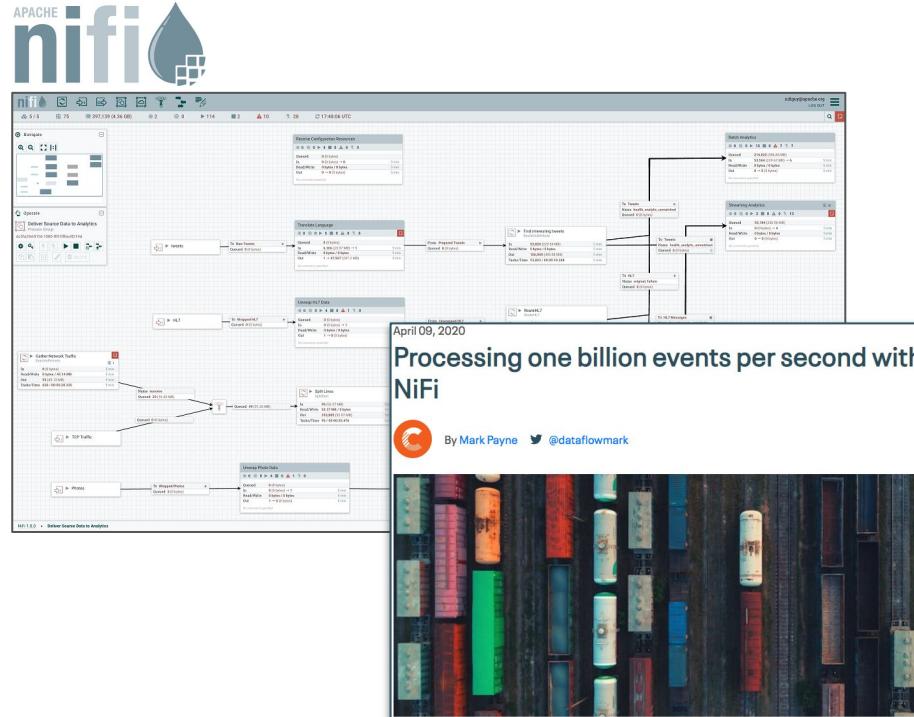
UNIVERSAL DATA DISTRIBUTION WITH CLOUDERA DATAFLOW (CDF)

Connect to Any Data Source Anywhere then Process and Deliver to Any Destination



CLOUDERA DATAFLOW - POWERED BY APACHE NiFi

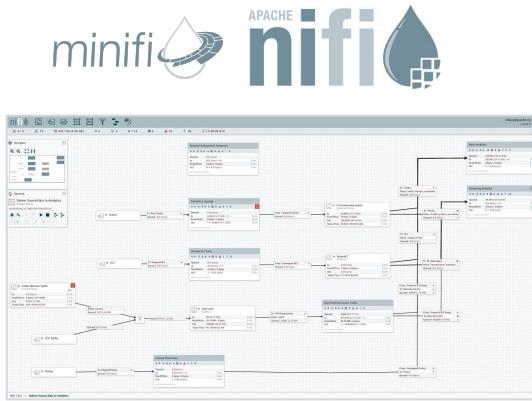
Ingest and manage data from edge-to-cloud using a no-code interface



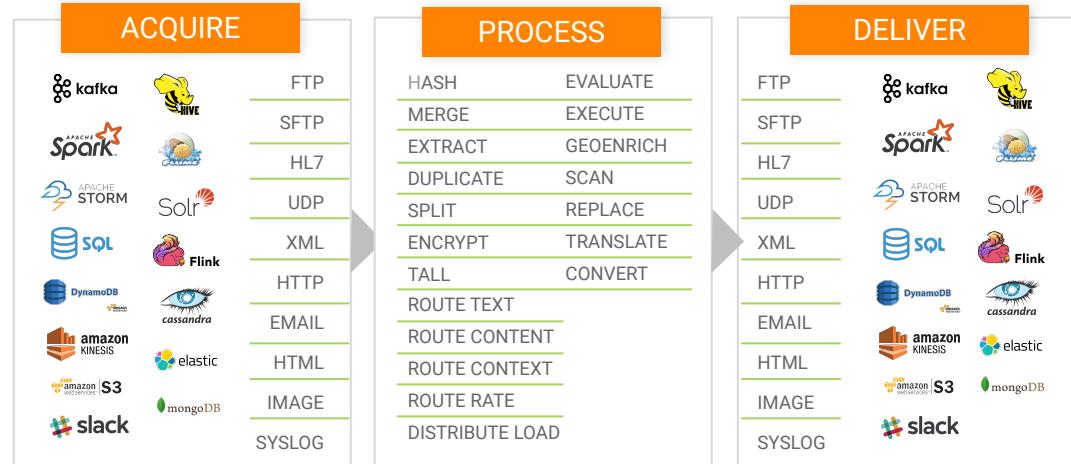
- #1 data ingestion/movement engine
- Strong community
- Product maturity over 11 years
- Deploy on-premises or in the cloud
- Over 400+ pre-built processors
- Built-in data provenance
- Guaranteed delivery
- Throttling and Backpressure

CLOUDERA FLOW AND EDGE MANAGEMENT

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



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

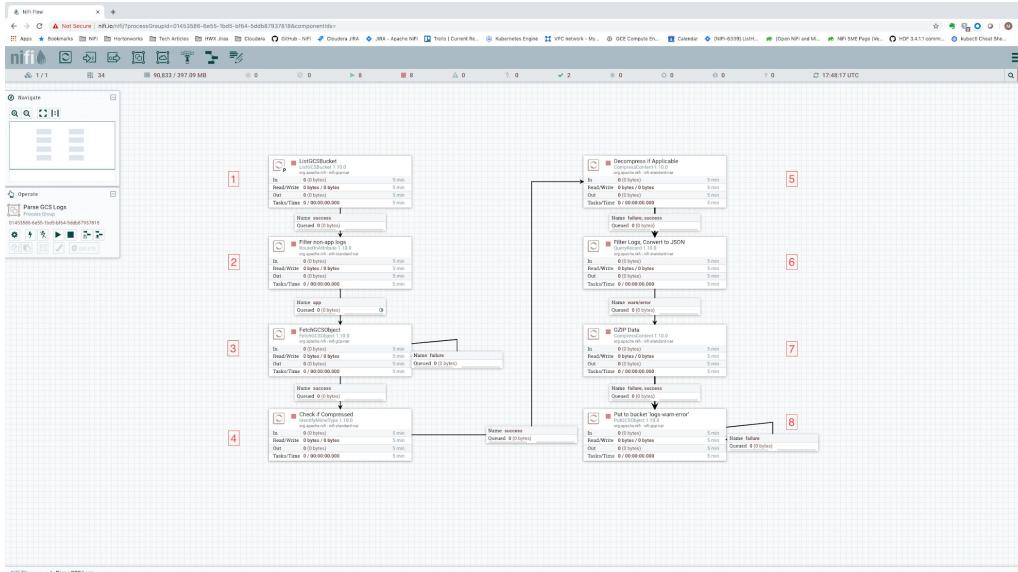


- Over 300 Prebuilt Processors
- Easy to build your own
- Parse, Enrich & Apply Schema
- Filter, Split, Merger & Route
- Throttle & Backpressure

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

Processing one millions events per second with Apache NiFi

<https://blog.cloudera.com/benchmarking-nifi-performance-and-scalability/>



| Nodes | Data rate/sec | Events/sec | Data rate/day | Events/day |
|-------|---------------|---------------|---------------|---------------|
| 1 | 192.5 MB | 946,000 | 16.6 TB | 81.7 Billion |
| 5 | 881 MB | 4.97 Million | 76 TB | 429.4 Billion |
| 25 | 5.8 GB | 26 Million | 501 TB | 2.25 Trillion |
| 100 | 22 GB | 90 Million | 1.9 PB | 7.8 Trillion |
| 150 | 32.6 GB | 141.3 Million | 2.75 PB | 12.2 Trillion |

PROVENANCE

Displaying 13 of 104
Oldest event available: 11/15/2016 13:34:50 EST

Showing the most recent events.

ConsumeKafka by component name

| Date/Time | Type | FlowFile Uuid | Size | Component Name | Component Type |
|--------------------------|---------|-------------------------------|----------|----------------|----------------|
| 11/15/2016 13:35:03.8... | RECEIVE | 379fc4f6-60e0-4151-9743-28... | 44 bytes | ConsumeKafka | ConsumeKafka |
| 11/15/2016 13:35:02.7... | RECEIVE | 78f8c38b-89fc-4d00-a8d8-51... | 44 bytes | ConsumeKafka | ConsumeKafka |
| 11/15/2016 13:35:01.6... | RECEIVE | 2bcd5124-bb78-489f-ad8a-7... | 44 bytes | ConsumeKafka | ConsumeKafka |

• Tracks data at each point as it flows through the system

• Records, indexes, and makes events available for display

• Handles fan-in/fan-out, i.e. merging and splitting data

• View attributes and content at given points in time

The diagram illustrates a data flow process. It starts with a red circle labeled "RECEIVE", which has an arrow pointing down to a grey circle labeled "JOIN". From the "JOIN" circle, an arrow points down to a grey circle labeled "DROP". A green arrow originates from the "RECEIVE" circle and points to a "Provenance Event" panel on the right. The "Provenance Event" panel contains three tabs: DETAILS, ATTRIBUTES, and CONTENT. The ATTRIBUTES tab is selected, showing the following attribute values:

| Attribute | Value | Previously Set |
|-----------------|---|-------------------------|
| filename | 328717796819631 | No value previously set |
| kafka.offset | 44815 | No value previously set |
| kafka.partition | 6 | No value previously set |
| kafka.topic | nifi-testing | No value previously set |
| path | / | No value previously set |
| uuid | 328717796819631-0000-0000-0000-000000000000 | No value previously set |

EXTENSIBILITY

- Built from the ground up with extensions in mind
- Service-loader pattern for...
 - Processors
 - Controller Services
 - Reporting Tasks
 - Prioritizers
- Extensions packaged as NiFi Archives (NARs)
 - Deploy NiFi lib directory and restart
 - Same model as standard components

The screenshot shows the IntelliJ IDEA interface with the project 'nifi-mxnetinference-processor' open. The code editor displays the file 'InferenceProcessorTest.java'. The test class contains methods for testing the 'InferenceProcessor' and 'SSDClassifierService'. Below the code editor, the 'Run' tool window shows a test run for 'InferenceProcessorTest.testProcessor'. The run details show three sequential steps: 1. A 'LinkProcessor' step with metrics: In 0 bytes, Read/Write 0 bytes / 31.45 KB, Out 2 (31.45 KB), Tasks/Time 2 / 0:00:04.808. 2. An 'UpdateAttribute' step with metrics: In 2 (31.45 KB), Read/Write 0 bytes / 0 bytes, Out 2 (31.45 KB), Tasks/Time 2 / 0:00:00.005. 3. A 'PutHDFS' step with metrics: In 2 (31.45 KB), Read/Write 31.45 KB / 0 bytes, Out 0 (0 bytes), Tasks/Time 2 / 0:00:00.603. Arrows indicate the flow from one step to the next.

```
/*
 */
public class InferenceProcessorTest {

    private TestRunner testRunner;

    @Before
    public void init() {
        testRunner = TestRunners.newTestRunner(InferenceProcessor.class);
    }

    private String pathForResource
        URL r = this.getClass()
        URI uri = r.toURI();
        return Paths.get(uri)
    }

    private void runAndAssert(
        testRunner.setValidation();
        testRunner.assertValue();
        testRunner.assertAll();
        List<MockFlowFile> success;
        for (MockFlowFile mockFlowFile : success) {
            assertEquals(expect, mockFlowFile.getAttribute("key_1"));
        }
    }

    @Test
    public void testProcessor() {
        runAndAssert();
    }
}
```

NiFi Load Balancing

- Improve NiFi cluster throughput
- Defined at connection level
- Configurable balancing strategies
- Critical for scale up paradigm in Kubernetes
- Alleviates S2S balancing “hack” customers use

The screenshot shows the NiFi interface with a flow editor. At the top, there's a 'DETAILS' tab and a 'SETTINGS' tab. Under 'DETAILS', you can see the flowfile's Name, Id (88cbd631-0166-1000-0000-00002af80f96), FlowFile Expiration (0 sec), and Back Pressure settings (Object Threshold: 10000, Size Threshold: 1 GB). The 'Available Prioritizers' list includes FirstInFirstOutPrioritizer, NewestFlowFileFirstPrioritizer, OldestFlowFileFirstPrioritizer, and PriorityAttributePrioritizer. The 'Selected Prioritizers' list is currently empty. Below these, the 'Load Balance Strategy' dropdown is set to 'Do not load balance'. In the center, there's a 'GenerateFlowFile' processor with the following metrics:

| | | |
|------------|-------------------|-------|
| In | 0 (0 bytes) | 5 min |
| Read/Write | 0 bytes / 42 KB | 5 min |
| Out | 42 (42 KB) | 5 min |
| Tasks/Time | 42 / 00:00:00.117 | 5 min |

On the right, there's a 'LogAttribute' processor with similar metrics:

| | | |
|------------|-------------------|-------|
| In | 41 (41 KB) | 5 min |
| Read/Write | 0 bytes / 0 bytes | 5 min |
| Out | 0 (0 bytes) | 5 min |
| Tasks/Time | 41 / 00:00:00.141 | 5 min |

Below these processors, a table lists flowfiles with their sizes and destinations:

| FlowFile | Size | Destination | Port |
|-------------------------------------|-------------|--------------|--------------|
| 0733ad94-3c80-44d7-9fc2-480caa... | 1,024 bytes | LogAttribute | LogAttribute |
| 2bc7b5c1-c164-40fb-9e7e-da57884... | 1,024 bytes | LogAttribute | LogAttribute |
| 80dece7a-15c8-4eb7-80ad-176bfef9... | 1,024 bytes | LogAttribute | LogAttribute |
| 98d9f9c4-bb47-4fe7-9786-964d027... | 1,024 bytes | LogAttribute | LogAttribute |
| 26c165ca-2fd6-4714-8c0c-e1de6e2... | 1,024 bytes | LogAttribute | LogAttribute |
| 8bfff920b-97a3-4b64-998d-046324a... | 1,024 bytes | LogAttribute | LogAttribute |
| 6345a326-4843-442e-b77d-480d20... | 1,024 bytes | LogAttribute | LogAttribute |
| 5fc30a5a-641e-4aa0-9c67-3b1d438... | 1,024 bytes | LogAttribute | LogAttribute |
| 1e90e7ee-92fe-47b3-9aa4-4094fa8... | 1,024 bytes | LogAttribute | LogAttribute |

A red box highlights the destination port 'hdf-pm-fe1' for the first few flowfiles, indicating a specific node in the cluster.

QUEUE CONFIGURATION

- **FlowFile Expiration** - Data that cannot be processed in a timely fashion can be automatically removed from the flow.
- **Back Pressure Thresholds** - Thresholds indicate how much data should be allowed to exist in the queue before the component that is the source of the Connection is no longer scheduled to run. This allows the system to avoid being overrun with data.
- **Load Balance Strategy** – Strategy to distribute the data in a flow across the nodes in the cluster. When enabled, compression can be configured on FlowFile contents and attributes.
- **Prioritization** – Determines the order in which flow files are processed.

| Generate Syslog RFC5424 | |
|--|-------------------|
| ExecuteScript 1.13.2.2.2.0-127 org.apache.nifi - nifi-scripting-nar | |
| In | 0 (0 bytes) |
| Read/Write | 0 bytes / 0 bytes |
| Out | 0 (0 bytes) |
| Tasks/Time | 0 / 00:00:00.000 |
| | 5 min |

Configure Connection

DETAILS SETTINGS

| | | | |
|--------------------------------|--------------------------------------|------------------------|--|
| Name | success_Generate-FilterEvents | Available Prioritizers | FirstInFirstOutPrioritizer NewestFlowFileFirstPrioritizer OldestFlowFileFirstPrioritizer PriorityAttributePrioritizer |
| Id | 64146cca-d197-3c27-9c47-015dd7b7a6c6 | Selected Prioritizers | |
| FlowFile Expiration | 0 sec | Load Balance Strategy | Round robin |
| Back Pressure Object Threshold | 10000 | Size Threshold | 1 GB |
| Load Balance Compression | Do not compress | | |

RECORD-ORIENTED DATA WITH NIFI

- **Record Readers** - Avro, CSV, Grok, IPFIX, JSON1, JSON, Parquet, Scripted, Syslog5424, Syslog, WindowsEvent, XML
- **Record Writers** - Avro, CSV, FreeFromText, Json, Parquet, Scripted, XML
- Record Reader and Writer support referencing a schema registry for retrieving schemas when necessary.
- Enable processors that accept any data format without having to worry about the parsing and serialization logic.
- Allows us to keep FlowFiles larger, each consisting of multiple records, which results in far better performance.

| Filter Events | |
|---|-------------------------|
| QueryRecord 1.13.2.2.2.2.0-127 org.apache.nifi - nifi-standard-nar | |
| In | 0 (0 bytes) 5 min |
| Read/Write | 0 bytes / 0 bytes 5 min |
| Out | 0 (0 bytes) 5 min |
| Tasks/Time | 0 / 00:00:00.000 5 min |

Configure Processor

SETTINGS SCHEDULING PROPERTIES COMMENTS

Required field

| Property | Value |
|---------------|---------------------|
| Record Reader | CSVReader |
| Record Writer | JsonRecordSetWriter |

+

RUNNING SQL ON FLOWFILES

- Evaluates one or more SQL queries against the contents of a FlowFile.
- This can be used, for example, for field-specific filtering, transformation, and row-level filtering.
- Columns can be renamed, simple calculations and aggregations performed.
- The SQL statement must be valid ANSI SQL and is powered by Apache Calcite.

| Filter Events | | |
|---|-------------------|-------|
| QueryRecord 1.13.2.2.2.2.0-127 org.apache.nifi - nifi-standard-nar | | |
| In | 0 (0 bytes) | 5 min |
| Read/Write | 0 bytes / 0 bytes | 5 min |
| Out | 0 (0 bytes) | 5 min |
| Tasks/Time | 0 / 00:00:00.000 | 5 min |

Configure Processor | QueryRecord 1.13.2.2.2.2.0-127

Stopped

SETTINGS SCHEDULING PROPERTIES COMMENTS

Required field

| Property | Value |
|-------------------------------|-------------------------|
| Record Reader | Syslog_5424_Reader |
| Record Writer | JSON_Syslog_5424_Writer |
| Include Zero Record FlowFiles | false |
| Cache Schema | false |
| Default Decimal Precision | 10 |
| Default Decimal Scale | 0 |
| filtered_events | #(Filter Rule) |

Apache NiFi with Python Custom Processors

Python as a 1st class citizen

```
import cv2
import numpy as np
import json
from nifiapi.properties import PropertyDescriptor
from nifiapi.properties import ResourceDefinition
from nifiapi.flowfiletransform import FlowfiletransformResult

SCALE_FACTOR = 0.00392
NMS_THRESHOLD = 0.4 # non-maximum suppression threshold
CONFIDENCE_THRESHOLD = 0.5

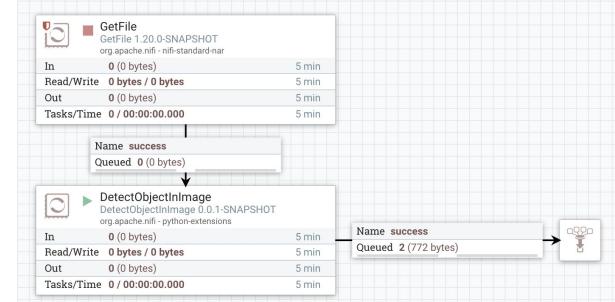
class DetectObjectInImage:
    class Java:
        implements = ['org.apache.nifi.python.processor.FlowfileTransform']
        class ProcessorDetails:
            version = '0.0.1-SNAPSHOT'
            dependencies = ['numpy >= 1.23.5', 'opencv-python >= 4.6']

    def __init__(self, jvm=None, **kwargs):
        self.jvm = jvm

        # Build Property Descriptors
        self.model_file = PropertyDescriptor(
            name = 'Model File',
            description = 'The binary file containing the trained Deep Neural Network weights. Supports Caffe (*.caffemodel), TensorFlow (*.pb), Torch (*.t7, *.net), Darknet (*.weights), ' +
                        'OLDY (*.bin), and ONNX (*.onnx)',
            required = True,
            resource_definition = ResourceDefinition(allow_file = True)
        )
        self.config_file = PropertyDescriptor(
            name = 'Network Config File',
            description = 'The text file containing the Network configuration. Supports Caffe (*.prototxt), TensorFlow (*.pbtxt), Darknet (*.cfg), and DLDT (*.xml)',
            required = False,
            resource_definition = ResourceDefinition(allow_file = True)
        )
        self.class_name_file = PropertyDescriptor(
            name = 'Class Names File',
            description = 'A text file containing the names of the classes that may be detected by the model. Expected format is one class name per line, new-line terminated.',
            required = True,
            resource_definition = ResourceDefinition(allow_file = True)
        )
        self.descriptors = [self.model_file, self.config_file, self.class_name_file]

    def getPropertyDescriptors(self):
        return self.descriptors

    def onScheduled(self, context):
        # read class names from text file
        class_name_file = context.getProperty(self.class_name_file.name).getValue()
        if class_name_file is None:
```



READYFLOW GALLERY

- Cloudera provided flow definitions
- Cover most common data flow use cases
- Optimized to work with CDP sources/destinations
- Can be deployed and adjusted as needed

ReadyFlow Gallery

Search by name

Added

Kafka filter to Kafka Version 1

Consumes JSON, CSV or Avro events from Kafka, filters them before writing them back to Kafka as JSON, CSV or Avro.

[View Added Flow Definition](#)

Added

Kafka to Cloudera Operational Database Version 1

Consumes JSON, CSV or Avro events from Kafka and ingests them into Cloudera Operational Database (COD).

[View Added Flow Definition](#)

Kafka to Kafka Version 1

Consumes events from Kafka and writes them to another Kafka topic.

[Add To Catalog](#)

Kafka to Kudu Version 1

Consumes JSON, CSV or Avro events from Kafka and ingests them into Kudu.

[Add To Catalog](#)

Kafka to S3 Avro Version 1

Consumes JSON, CSV or Avro events from Kafka and writes Avro files to S3.

[View Added Flow Definition](#)

S3 to S3 Avro Version 1

Consumes JSON, CSV or Avro files from source S3 location and writes Avro files to a destination S3 location.

[Add To Catalog](#)

FLOW CATALOG

- Central repository for flow definitions
- Import existing NiFi flows
- Manage flow definitions
- Initiate flow deployments

The screenshot shows the Cloudera DataFlow Flow Catalog interface. On the left is a dark sidebar with navigation links: Dashboard, Catalog (which is selected), ReadyFlow Gallery, Environments, Help, and a user icon. Below the sidebar, the version is listed as 1.0.1-b570. The main area is titled "Flow Catalog" and contains a search bar and a refresh button indicating it was last refreshed 25 seconds ago. A blue "Import Flow Definition" button is located in the top right. The catalog table lists ten flow definitions, each with a name, type, versions, and last updated date, followed by a "View" link. The columns are Name (sorted ascending), Type, Versions, Last Updated, and a View link.

| Name ↑ | Type | Versions | Last Updated | |
|--|------------------------|----------|--------------|---|
| cc_fraud_template_int101run | Custom Flow Definition | 2 | a day ago | > |
| cc_fraud_template_int101run2 | Custom Flow Definition | 1 | 9 days ago | > |
| JSON_Kafka_To_Avro_S3 | Custom Flow Definition | 2 | a day ago | > |
| Kafka filter to Kafka | ReadyFlow | 1 | 2 days ago | > |
| Kafka to Cloudera Operational Database | ReadyFlow | 1 | 2 days ago | > |
| Kafka to S3 Avro | ReadyFlow | 1 | 14 hours ago | > |
| nifi_flows | Custom Flow Definition | 1 | 2 months ago | > |
| Weather Data Flow | Custom Flow Definition | 1 | a day ago | > |
| Weather_Data | Custom Flow Definition | 1 | 15 days ago | > |
| Weather_JSON_Kafka_To_Avro_S3 | Custom Flow Definition | 1 | 21 days ago | > |

Items per page: 10 | < < > > | 1 – 10 of 10

DEPLOYMENT WIZARD

- Turns flow definitions into flow deployments
- Guides users through providing required configuration
- Choose NiFi runtime version
- Pick from pre-defined NiFi node sizes
- Define KPIs for the deployment

Start Deployment Wizard

dataflow-demo-new / New Flow Deployment

Overview

Selected Flow Definition

| | | | |
|------|---------------------------|---------|---|
| NAME | Machine Data To Warehouse | VERSION | 3 |
|------|---------------------------|---------|---|

Target Environment

| | |
|------|-------------------|
| NAME | dataflow-demo-new |
|------|-------------------|

NiFi Runtime Version

| | | |
|-----------------|-----------------------------------|----------------|
| CURRENT VERSION | Latest Version (1.14.0.2.3.1.0-3) | Change Version |
|-----------------|-----------------------------------|----------------|

Deployment Name

Provide Parameters

Flow Parameters

Data entered here never leaves the environment in your cloud account. Provide parameter values directly in the text input or upload a file for parameters that expect a file.

MachineData

AWS Credential File

Select File

Drop file or browse

CDP Truststore

Select File

Drop file or browse

CDPSchemaRegistry

<https://dataflow-streams-master0.dataflow.xcu2-8y8x.dev.cldr.work:7790/api/v1>

Configure Sizing & Scaling

Overview

Flow Parameters

Sizing & Scaling

Key Performance Indicators

Review

Sizing & Scaling

Select the NiFi node size and the number of nodes provisioned for your flow.

NiFi Node Sizing

| | | | |
|--|------------------------------------|-------------------------------------|--------------------------------------|
| <input checked="" type="radio"/> Extra Small | <input type="radio"/> Small | <input type="radio"/> Medium | <input type="radio"/> Large |
| 2 vCores Per Node 4 GB Per Node | 4 vCores Per Node 8 GB Per Node | 8 vCores Per Node 16 GB Per Node | 16 vCores Per Node 32 GB Per Node |

Number of NiFi Nodes

Auto Scaling

Enabled

Min. Nodes: 1

Max. Nodes: 3

Define KPIs

Overview

Flow Parameters

Sizing & Scaling

Key Performance Indicators

Review

Key Performance Indicators

Set up KPIs to track specific performance metrics of a deployed flow. Click and drag to reorder how they are displayed.

| | | | |
|-------------|-----------------|--|------------------------------------|
| Entire Flow | METRIC TO TRACK | <input type="checkbox"/> Data In | <input type="checkbox"/> ALERT SET |
| | | Notify if less than 150 KB/sec, for at least 30 seconds. | |

| | | | |
|--|-----------------|-------------------------------------|------------------------------------|
| Processor: Write to S3 using HDFS proc | METRIC TO TRACK | <input type="checkbox"/> Bytes Sent | <input type="checkbox"/> ALERT SET |
| | | No alert set | |

Add New KPI

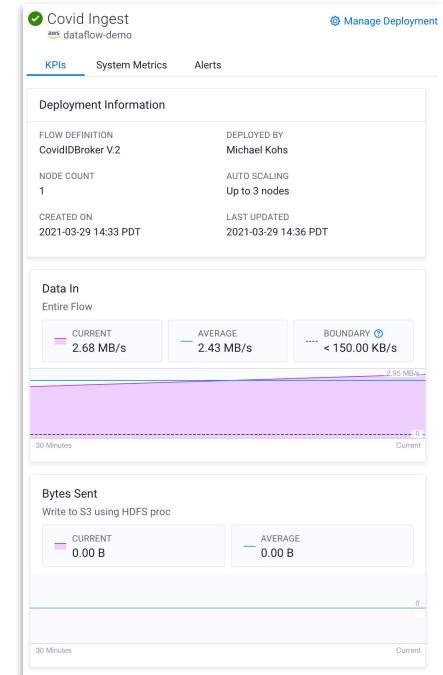
KEY PERFORMANCE INDICATORS

- Visibility into flow deployments
- Track high level flow performance
- Track in-depth NiFi component metrics
- Defined in Deployment Wizard
- Monitoring & Alerts in Deployment Details

KPI Definition in Deployment Wizard

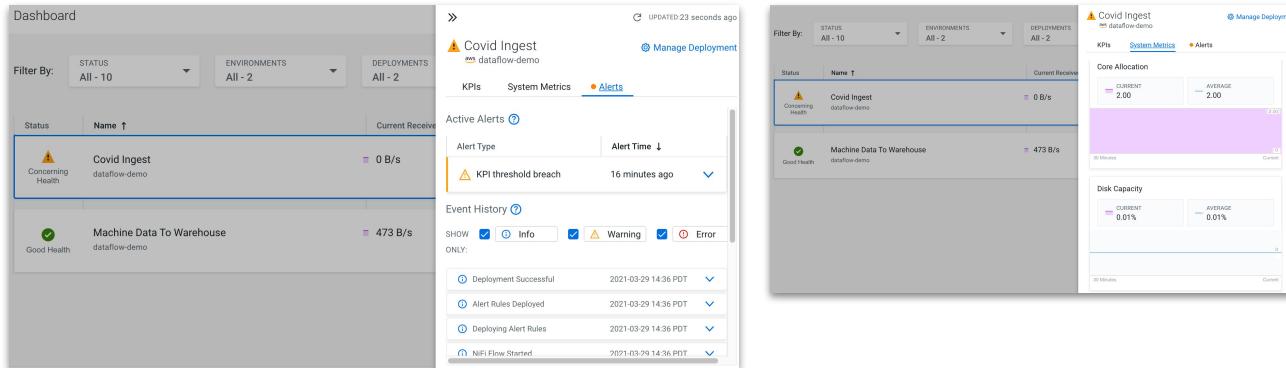
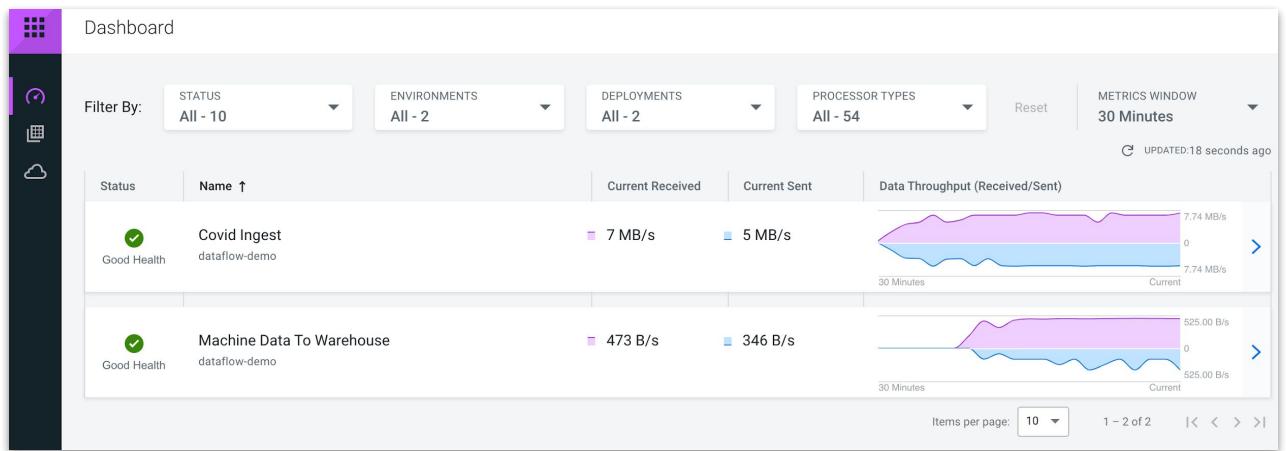
The screenshot shows the 'New Deployment' step of the deployment wizard. On the left, a sidebar lists steps: Overview, NiFi Configuration, Parameters, Sizing & Scaling, Key Performance Indicators (selected), and Review. The main area is titled 'Key Performance Indicators' with the sub-section 'Entire Flow'. It shows 'METRIC TO TRACK' set to 'Flow Files Queued', 'ALERT SET' as 'No alert set', and a note: 'Notify if outside the range of 999 MB/sec - 1 MB/sec, for at least 5 minutes.' A button at the bottom right says '(+) Add New KPI'.

KPI Monitoring



DASHBOARD

- Central Monitoring View
- Monitors flow deployments across CDP environments
- Monitors flow deployment health & performance
- Drill into flow deployment to monitor system metrics and deployment events



DEPLOYMENT MANAGER

- Manage flow deployment lifecycle
(Suspend/Start/Terminate)
- Add/Edit KPIs
- Change sizing configuration
- Update parameters
- Change NiFi version of the deployment
- Gateway to NiFi canvas

Dashboard / dataflow-demo-new / Kafka to COD

REFRESHED 12 seconds ago

Actions ▾

Back to Deployment Details

Deployment Manager

Status: Good Health

Deployment Name: Kafka to COD

Flow Definition: Kafka to Cloudera Operational Database V1

Deployed By: Michael Kohs

Node Count: 1

Auto Scaling: Disabled

Created On: 2021-07-26 17:05 PDT

Last Updated: 2021-07-26 17:07 PDT

Environment: dataflow-demo-new

Region: US West (Oregon)

NIFI Runtime Version: 1.14.0-2.3.0-0-89

Deployment Settings

KPIs and Alerts Sizing and Scaling Parameters

Parameters

Running Processors that are affected by the Parameter changes will automatically be restarted.

Data entered here never leaves the environment in your cloud account. Provide parameter values directly in the text input or upload a file for parameters that expect a file.

The selected flow definition references an external Default NiFi SSL Context Service. Hence, DataFlow will automatically create a matching SSL Context Service with a keystore and truststore generated from the target environment's FreeIPA certificate.

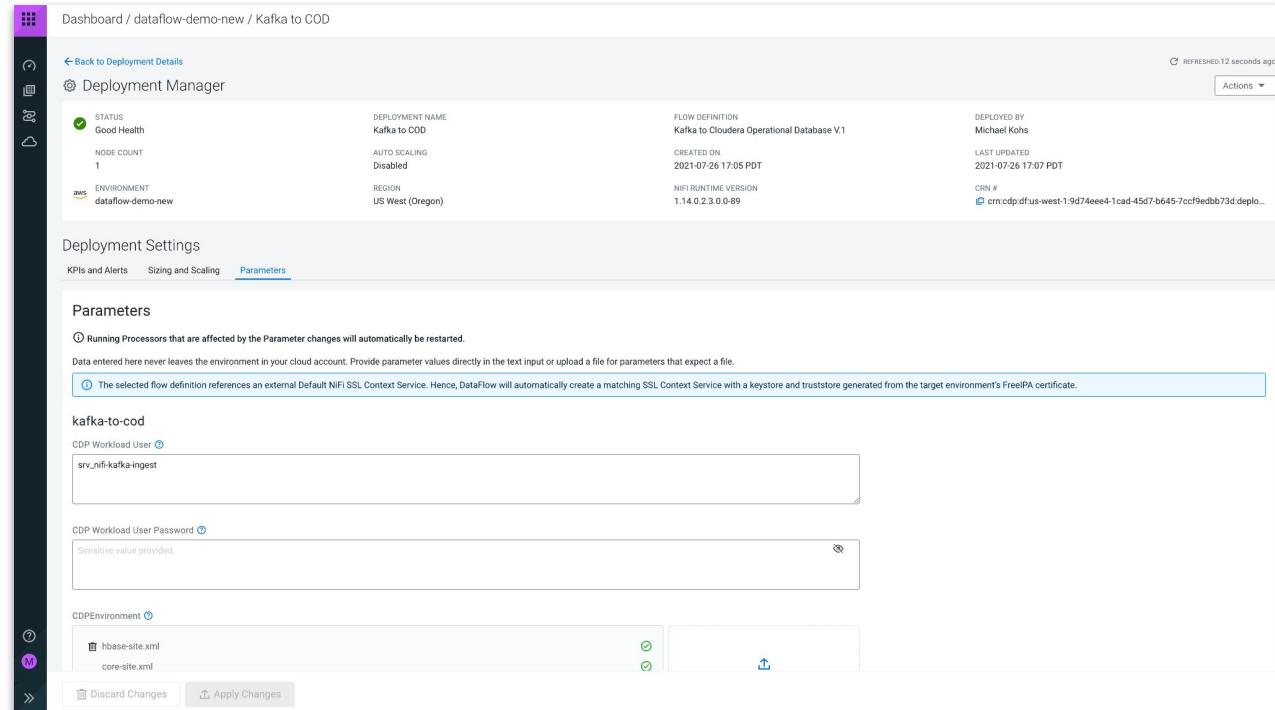
kafka-to-cod

CDP Workload User:

CDP Workload User Password:

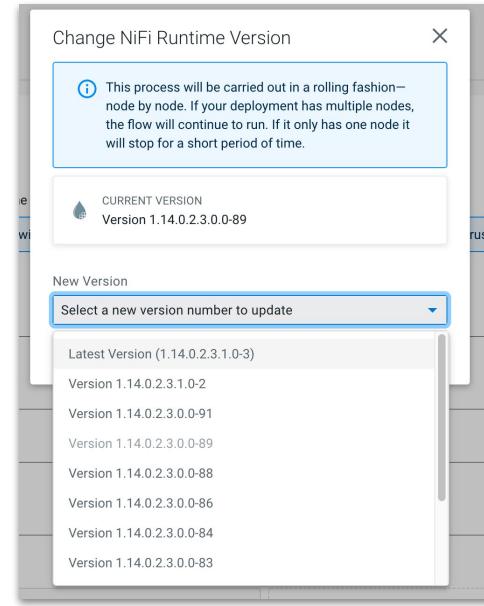
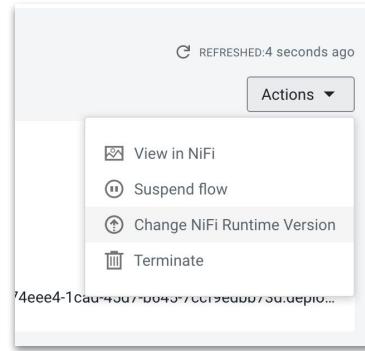
CDP Environment: hbase-site.xml core-site.xml

Discard Changes Apply Changes



NIFI VERSION UPGRADES

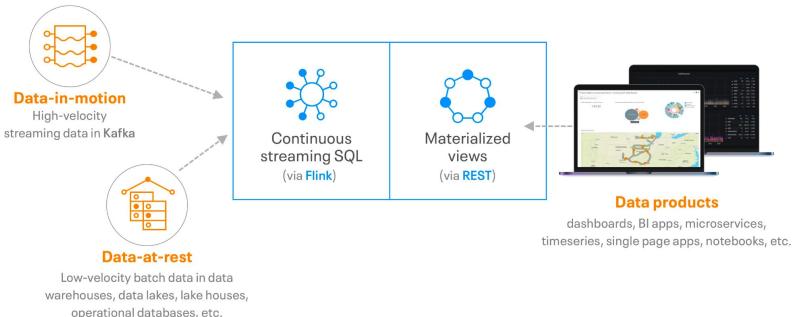
- Pick up NiFi hotfixes easily
- Upgrade (or downgrade) the hotfix version of existing deployments
- Rolling upgrade (if the deployment has >1 NiFi nodes)



BEST PRACTICES

STREAMING TECH DEBT TIPS

- Version Control All Assets
- Managed Public Cloud like Cloudera
- Use DevOps and APIs
- Latest Java and Python
- Stream Sizing (NiFi, Kafka, Flink)



Streaming Solutions

When to use what?

Routing vs Analytics

Listeners

Joins

In-Memory

Operational Load

Current Skills

Use NiFi

Doing more than just Syndication

Not just small Kafka sized events

Edge Management is needed

Listener Type use cases that bind to ports

Lightweight ETL, Lineage, Provenance, Message Replay

Use Flink

Joining Streams

Windowing

Late Data Handling

Streaming Analytics

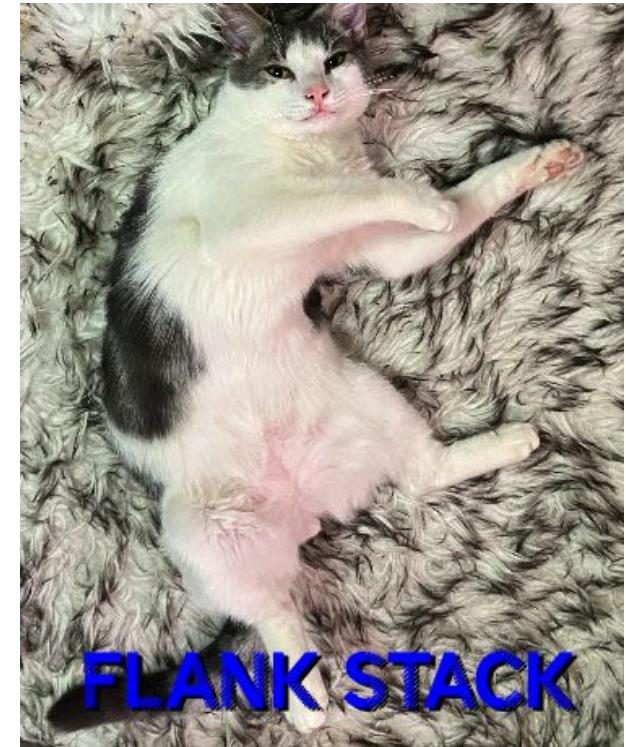
Use KConnect

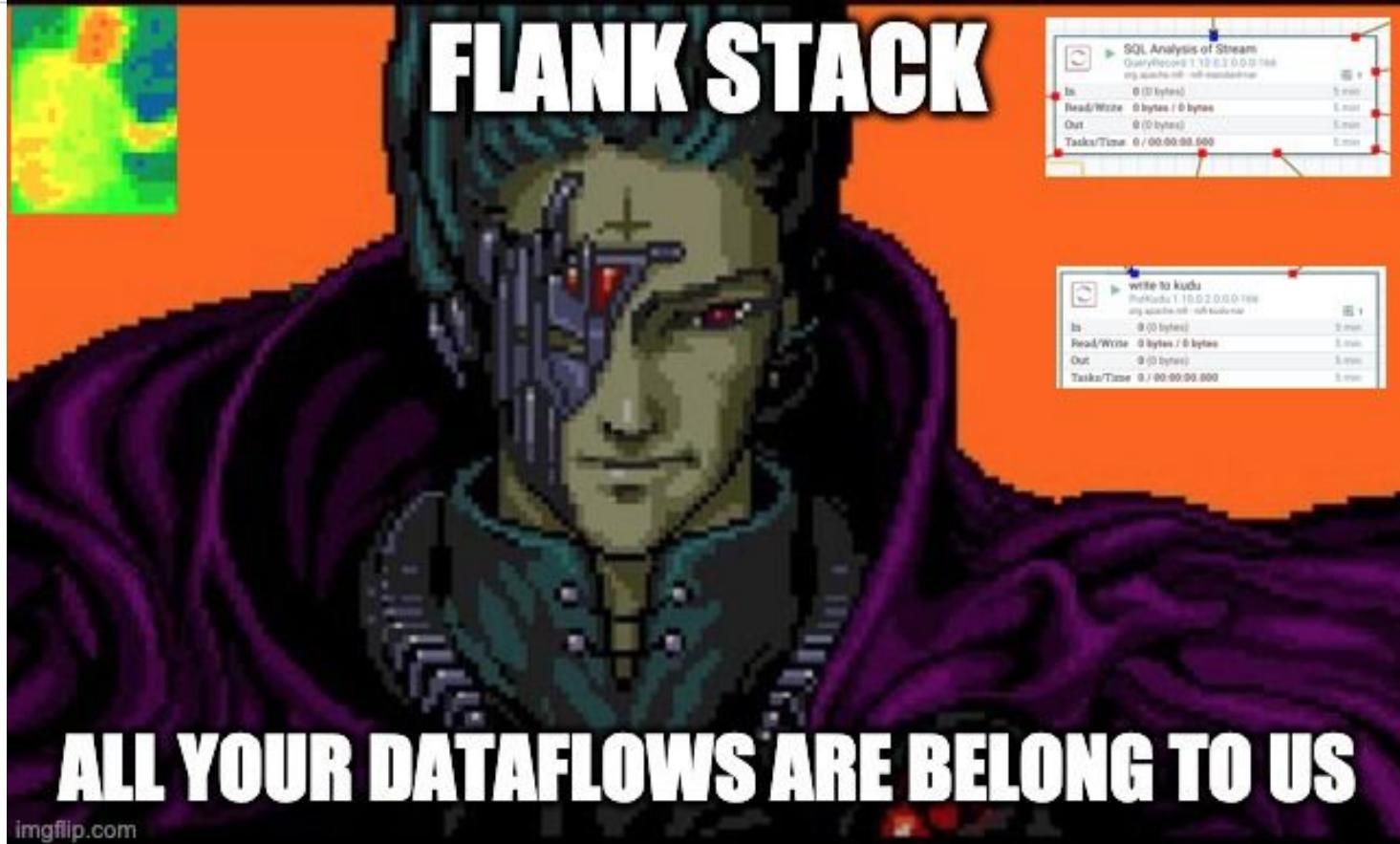
Kafka Centric

In-Memory Stateless

RESOURCES AND WRAP-UP

Resources





Upcoming Events

April 26

BREAKOUT SESSION
Building a Real-Time IoT Application with Apache Pulsar and Apache Pinot

Timothy Spann
Cloudera

The slide features a dark background with a grid of yellow dots in the upper right. In the lower right, there's a graphic for the "REAL-TIME ANALYTICS SUMMIT 2023".

May 9



Next Meeting O
Modern Data Streaming Pipelines
Abstract

In his modern approach, Tim will share his perspective that he has discovered what makes a great application. We often think of streaming as just another way of processing data flowing into Apache Kafka. From there we build interesting dashboards, ML models, and other cool things. But what if you want to do something more? What if you want to make your application better?

Tweets from @OrderlessJUG

CLOUDERA

May 10

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

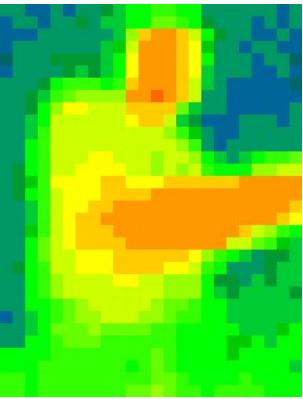
BRINGING OPEN TOGETHER,
MAY 10 – 12, 2023
VANCOUVER, CANADA
#OSSUMMIT

CLOUDERA

Open Source Summit
North America 2023

**Building Modern
Data Streaming Apps**
May 10, 2023

Tim Spann
Principal DTM Specialist and
Developer Advocate
Cloudera



TH^ON^G Y^OU[★]

