CLOUDERA

# Meetup: Mastering Data Streaming Pipelines

**Tim Spann**
**Principal Developer Advocate**

**28-June-2023**

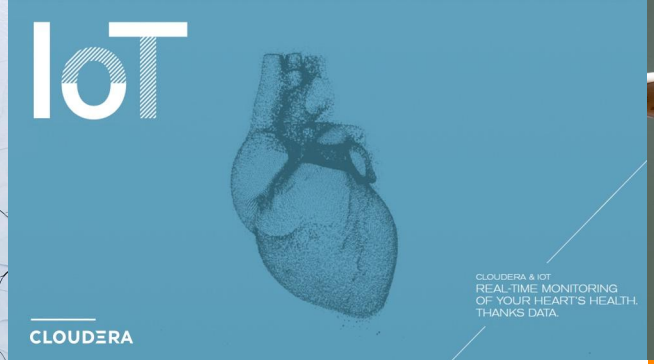
CLOUDERA

# Streaming Data Pipeline Development

**Ian Brooks, Ph.D.**
Principal Solutions Engineer

Join us on Wednesday, June 28, 2023 | 6:00 PM ET

**Timothy Spann**
Principal Developer Advocate




CLOUDERA

EDGE
2AI
CLOUDERA

ENTERPRISE
DATA
CLOUD
CLOUDERA

# Future of Data - Princeton + Virtual

FUTURE OF DATA

AN OPEN SOURCE COMMUNITY

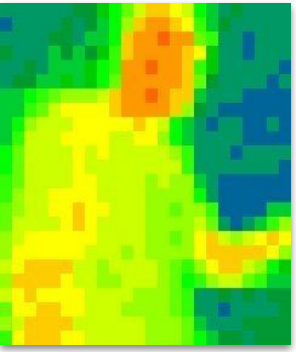
https://www.meetup.com/futureofdata-princeton/

**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 ...**

ICEBERG

@PaasDev

# FLaNK Stack



**Tim Spann**

@PaasDev // Blog: **www.datainmotion.dev**

**Principal Developer Advocate.**

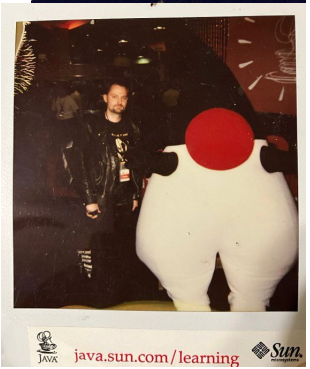
Princeton Future of Data Meetup.

**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**





# FLaNK 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.

CLOUDERA

# NOW 23

https://www.cloudera.com/about/events/cloudera-now-cdp.html

# 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

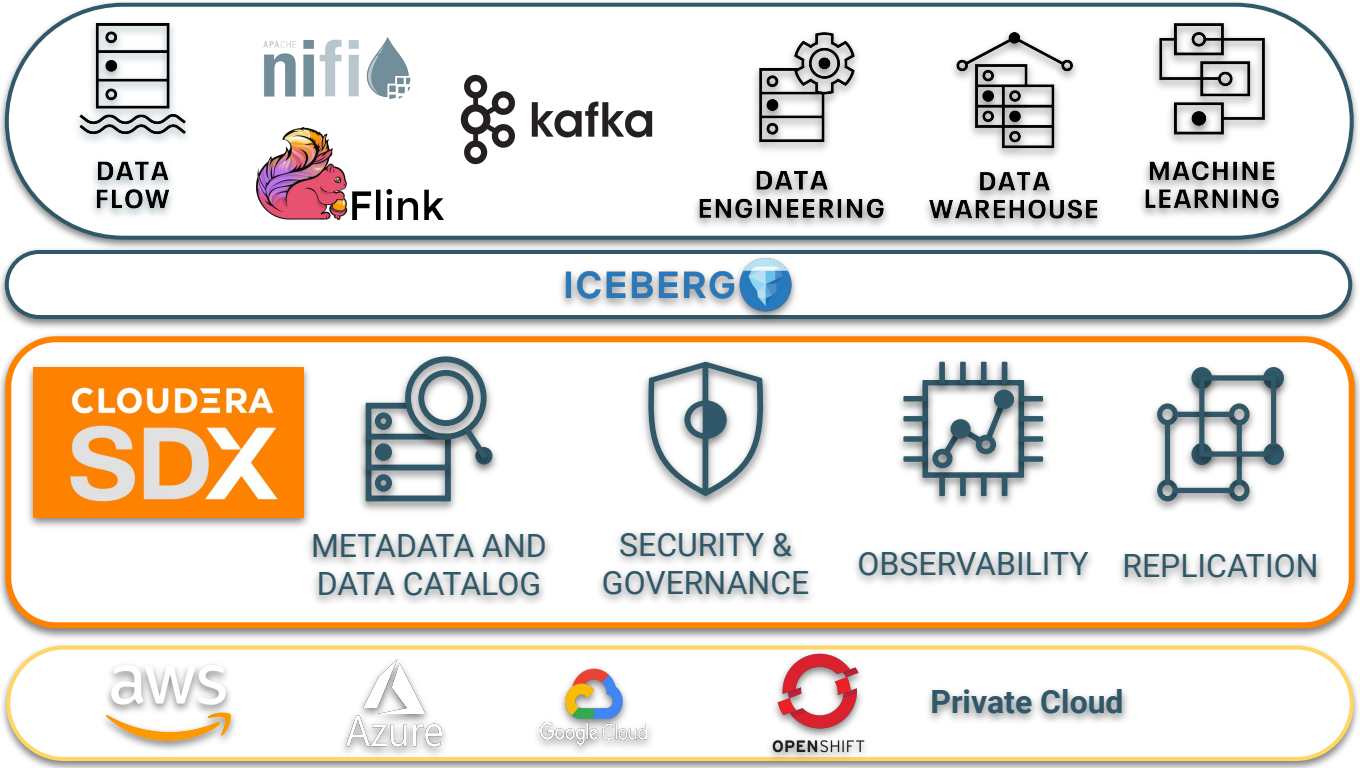
**CSP Community Edition**

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

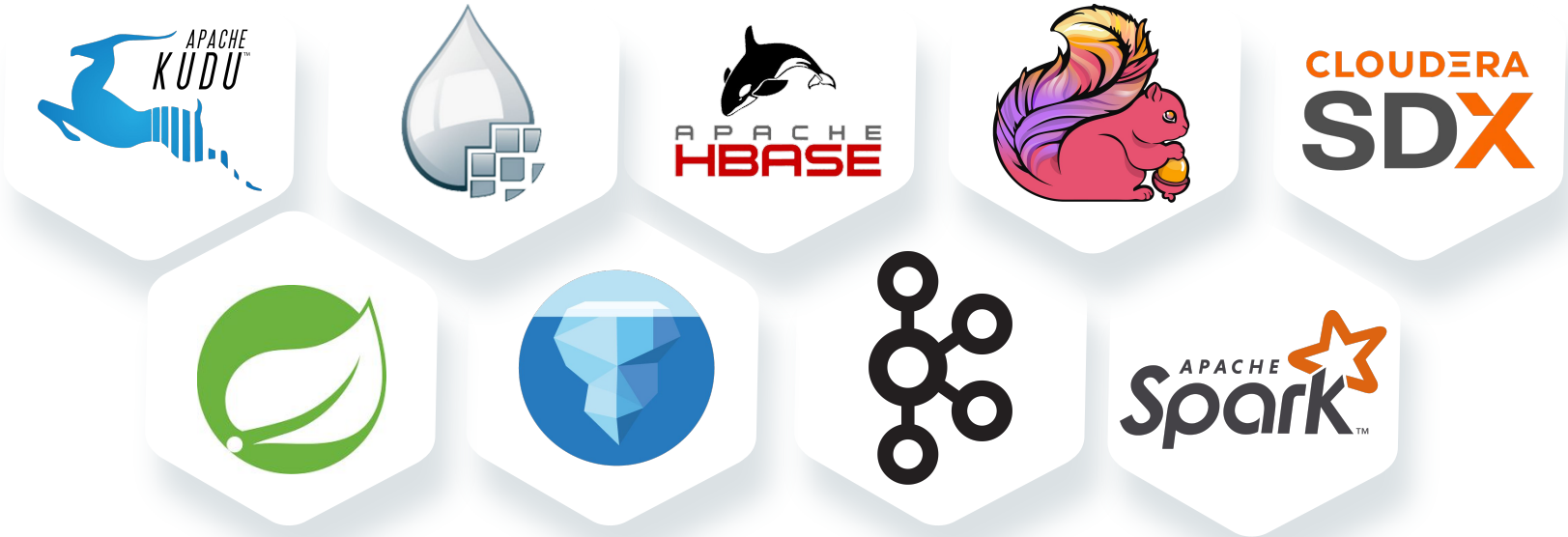
Learn More

# STREAMING

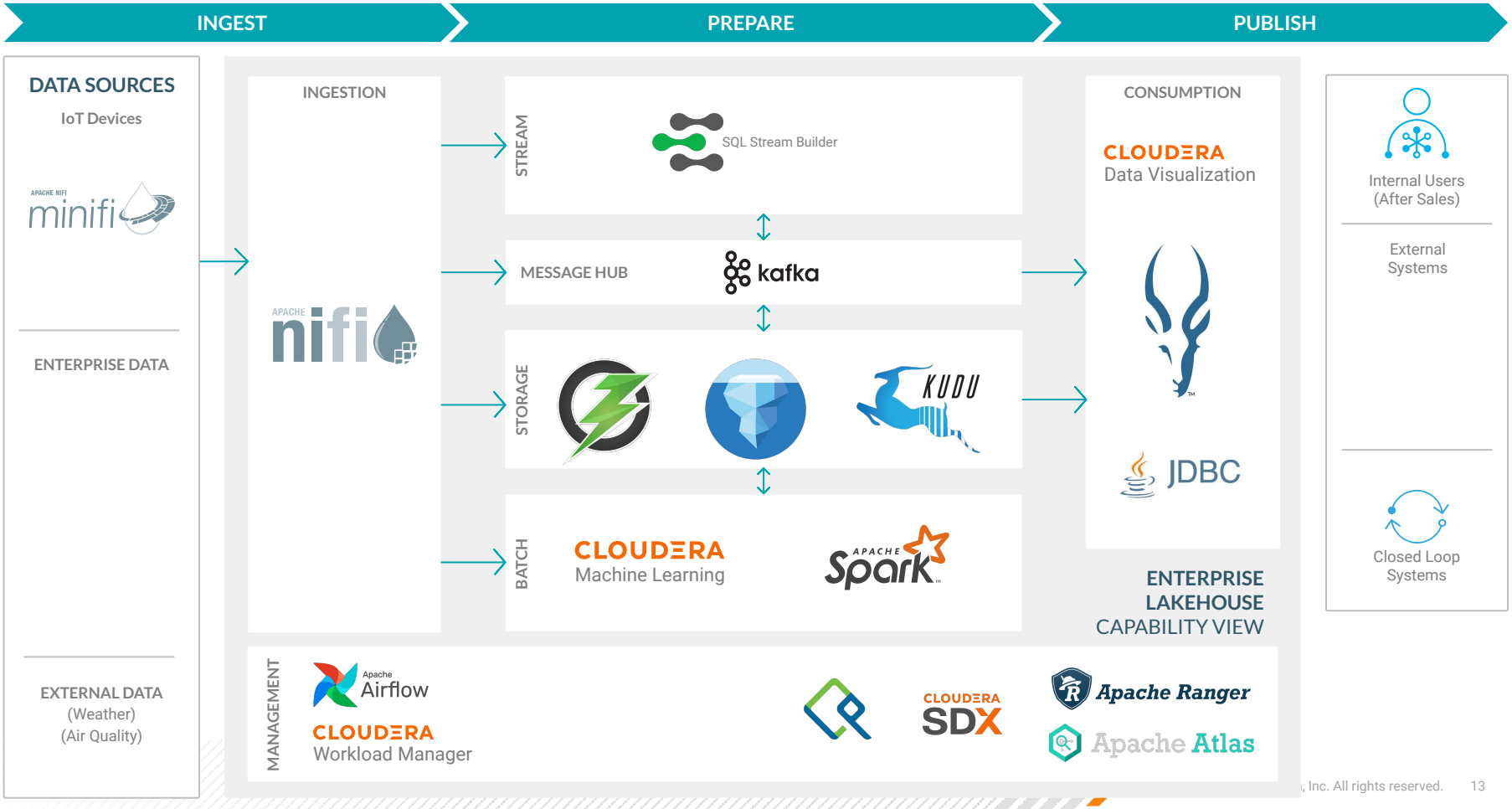
# CDP: AN OPEN DATA LAKEHOUSE



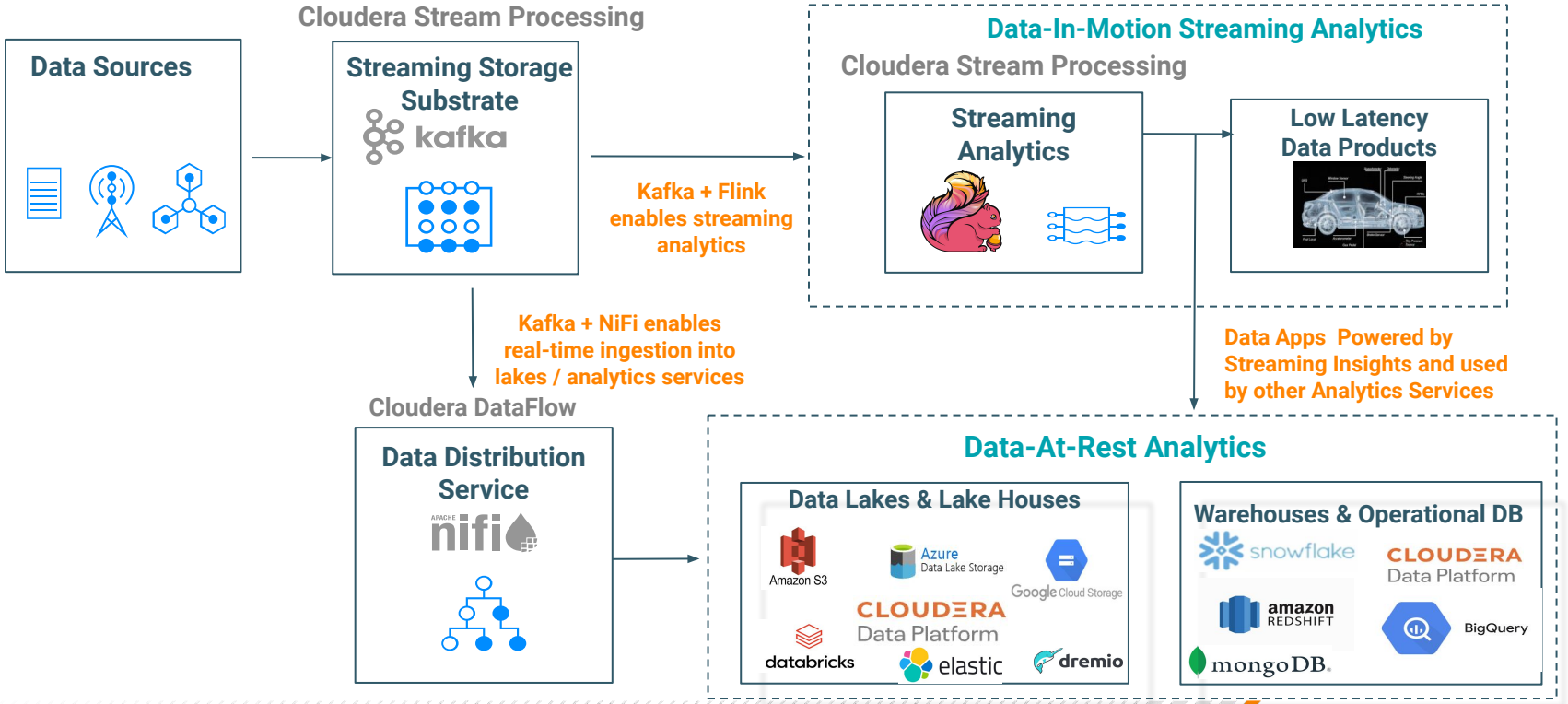
# BUILDING REAL-TIME REQUIRES A TEAM



# DATA INGESTION ARCHITECTURE - Using NiFi to pull the Data out of anything in near-real time



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



**Sploot**

**Nano**

**Shaggy**

**Stormy**

# Spark, NiFi, Flink? Which engine to choose?

## Already using Spark?

Want unified Batch/Stream?

Want highest Throughput?

Don't need low latency?

Large files?

Scheduled batches?

Replacing Sqoop, ETL


## Need NiFi?

Simple JDBC queries?

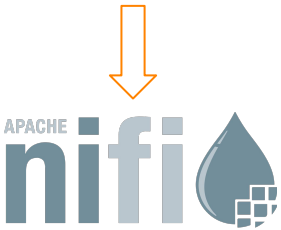
Transform individual records?

Want easy development?

Lots of small files, events, records, rows?

Continuous stream of rows

Support many different sources




## Need Flink?

Need Microservices, Batch and Stream?

Want high Throughput?

Want Low Latency?

Want Advanced Windowing and State?

Happy with a New Solution that is best-in-class?

# SOURCES AND SINKS

# APACHE ICEBERG

## A Flexible, Performant & Scalable Table Format

- Donated by Netflix to the Apache Foundation in 2018
- Flexibility
  - Hidden partitioning
  - Full schema evolution
- Data Warehouse Operations
  - Atomic Consistent Isolated Durable (ACID) Transactions
  - Time travel and rollback
- Supports best in class SQL performance
  - High performance at Petabyte scale

AMQP


AWS Lambda


Airtable


Amazon API Gateway


Amazon CloudWatch

Amazon DynamoDB


Amazon Kinesis Data Firehose


Amazon Kinesis Data Streams


Amazon SQS


Amazon Simple Notification Services (SNS)


Amazon Simple Storage Service (S3)


Apache Accumulo


Apache Cassandra


Apache HBase


Apache Hive


Apache Iceberg


Apache Ignite


Apache Kafka


Apache Kudu


Apache Solr

# COMMON USE CASES

# BE READY TO DETECT, RESPOND AND COMPLY

## Real-time alerting & Longer retention = Enhanced Readiness

**1**

### Detect

Collect event data at high events per second

Automatically identify high fidelity alerts

Analytics for proactive threat hunting

**2**

### Respond

Respond automatically to real-time alerts

Full context for complete forensics and incident response

**3**

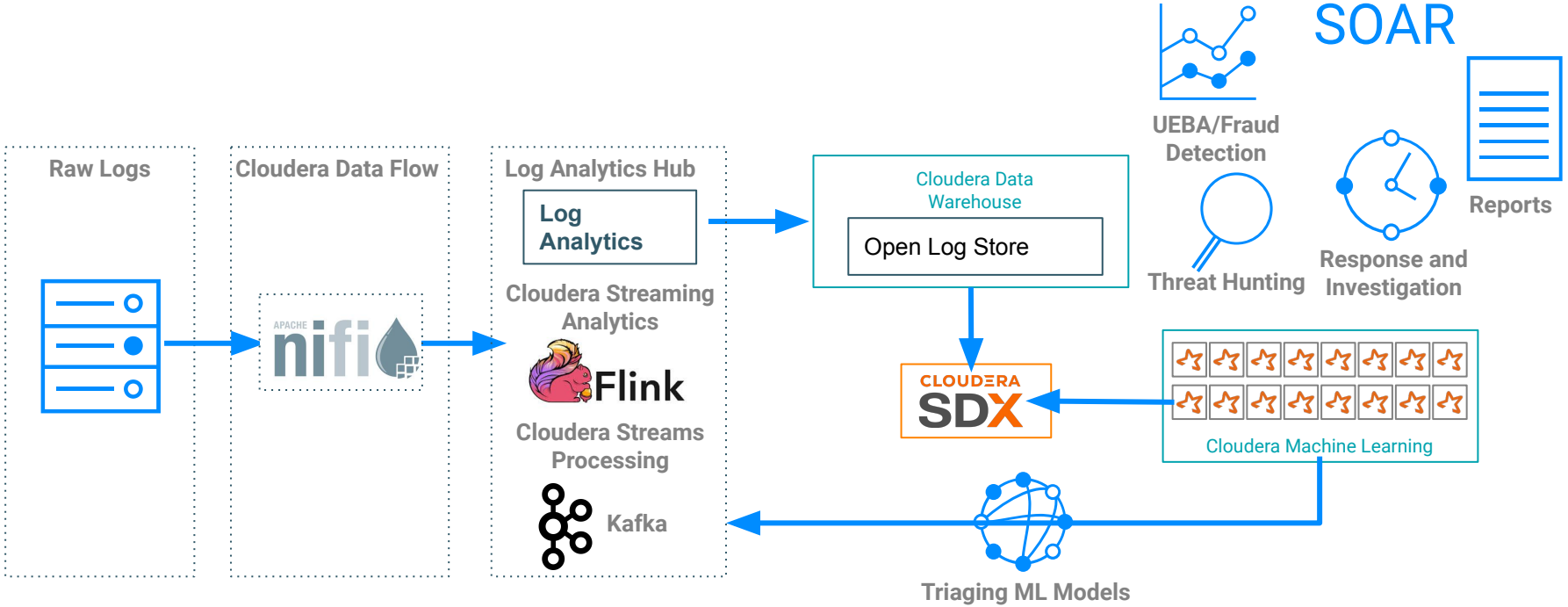
### Comply

Retain data as required

Automated compliance assessments

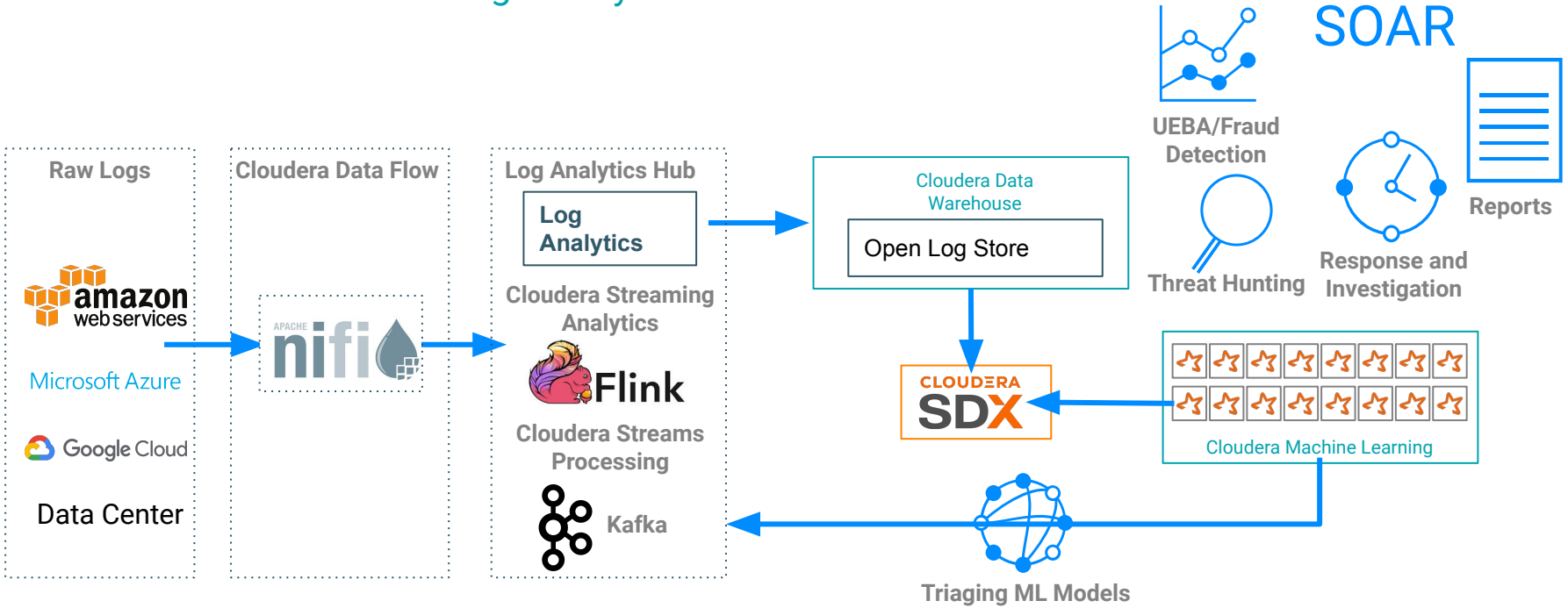
# STREAMING LOG ANALYTICS REFERENCE ARCHITECTURE

## Scalable and Secure Real-Time Log Analytics



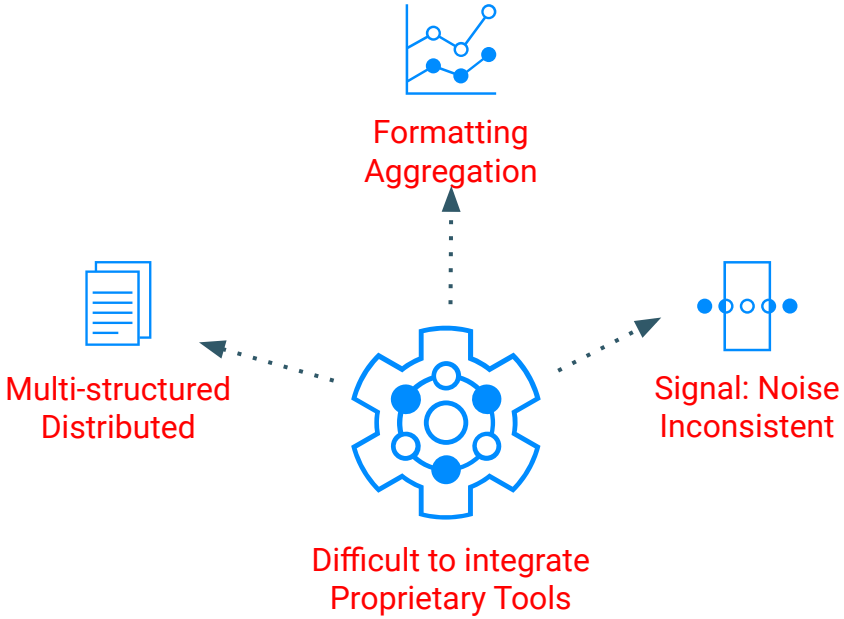
# CONSOLIDATE ACROSS CLOUDS

## Run collection and streaming on any cloud



# THE PROBLEM

## Making Observable Events *Actionable* can be Challenging




### Resulting in….

- Inability to **distribute** event data
- Data Silos
- Inefficient data movement
- **Situational awareness**

- License costs
- Consumption costs
- Development overhead
- **Resource utilization**

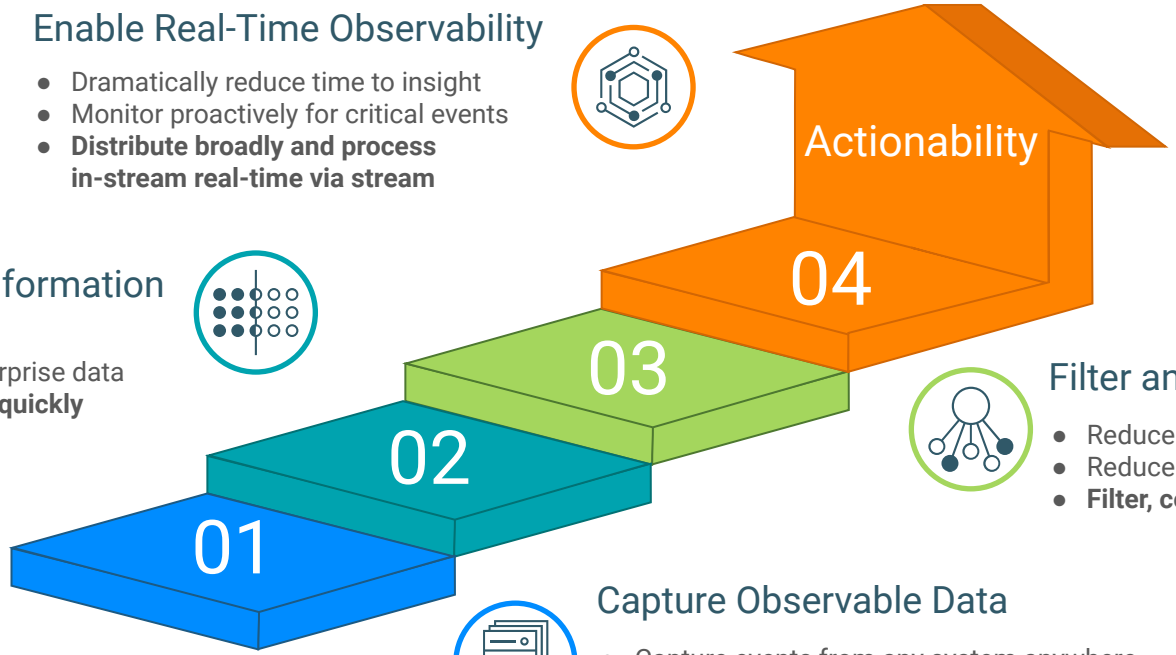
# CLOUDERA DATA-IN-MOTION SOLUTION

## Enable Real-Time Observability

- Dramatically reduce time to insight
- Monitor proactively for critical events
- **Distribute broadly and process in-stream real-time via stream**

## Create Useful Information

- Process raw data
- Integrate with enterprise data
- **Build, test, deploy quickly**




## Filter and Distribute

- Reduce noise, boost signal
- Reduce load on other system
- **Filter, compress, route anywhere**

## Capture Observable Data

- Capture events from any system anywhere
- Any data format, any protocol
- **Open- Not limited to proprietary tools**

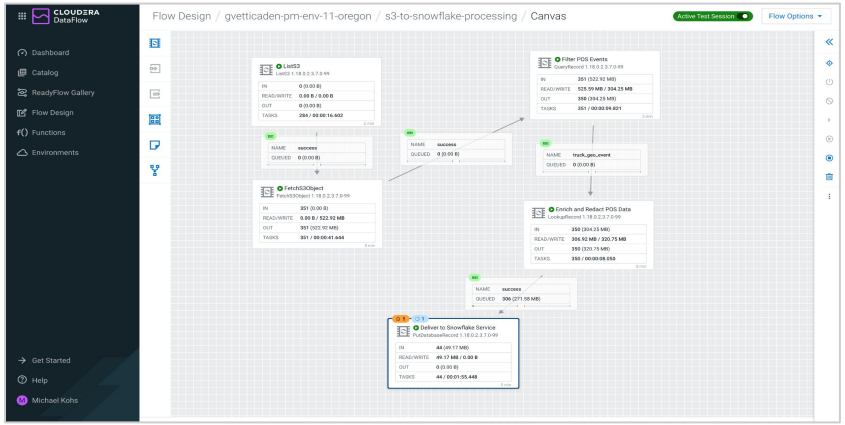
**Universal Connectivity**
**450+ Connectors**

**Hybrid Deployment-Ready**
**Author once, deploy anywhere**

**Self-Service across full lifecycle**
**Integrated no code UX**

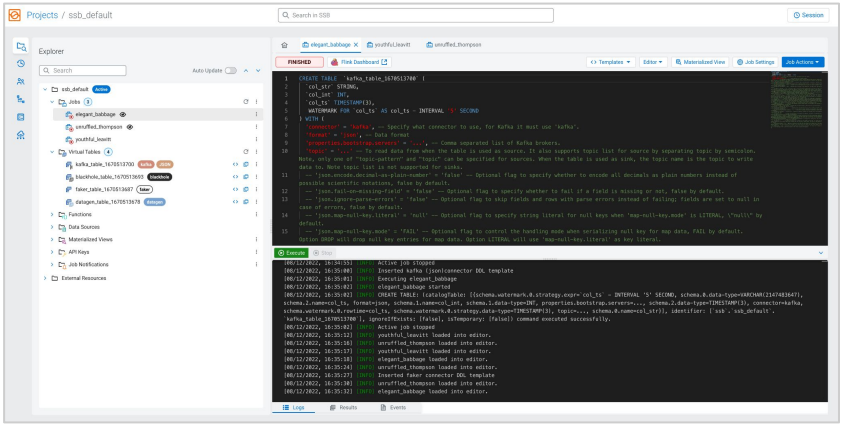
# NO-CODE/LOW-CODE USER EXPERIENCE

## DataFlow Designer




visual canvas + automatic provisioning=
***Self-service pipeline development***

## SQL Stream Builder




Instant data access + unified SQL processing =
***Self-service streaming event processing***

# KEY USE CASES AND OUTCOMES

## CYBER SECURITY

"**Cut log data by 60% resulting in >40% reduction in Splunk costs**. Delivered cyber logs to other teams analyzing logs resulting in a **decrease of MTTD by 90%.**"

Fortune 100 Energy Company

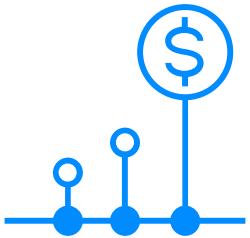
"Improved threat hunting by **eliminating 25% of false positive anomalies** and speeding up investigations"

Global Telco

## CUSTOMER INSIGHT

"**Improved Marketing Campaign effectiveness 15%** YoY. This allowed us to redirect funds for failing campaigns and allocate to new ones."

Global Telco



## INTELLIGENT OPS & SYSTEM RELIABILITY

"**Cut our data volume in half that translated to a 30%-40% reduction in public cloud cost**. With saving, expanded footprint & coverage of monitoring & visibility without a significant $$$ impact. This drove **improved situational awareness of app infrastructure availability & performance**"

Large US Airline

"**Faster resolution** of incidents, reducing exponentially the stops of trains in full operation."

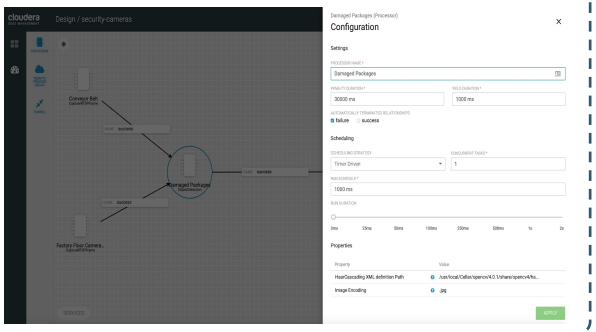
Metro de Madrid

# APACHE NiFi - MiNiFi Agents

# Cloudera Edge Management

## Edge device data collection and processing with easy to use central command and control

### Edge Flow Manager

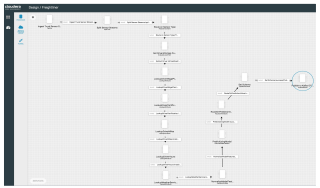



APACHE NIFI minifi

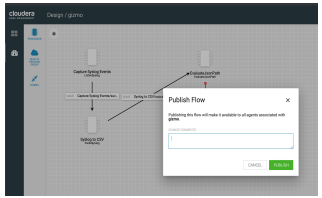
A lightweight edge agent that implements the core features of Apache NiFi, focusing on data collection and processing at the edge

- Small footprint agent with MiNiFi
- Java and C++ agents
- Rich edge processors (edge collection & processing)
- End to end lineage and security
- Kubernetes support

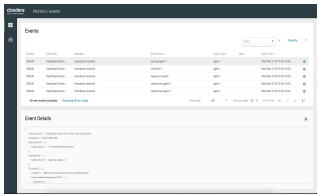
**Flow Authorship**



**Flow Deployment**



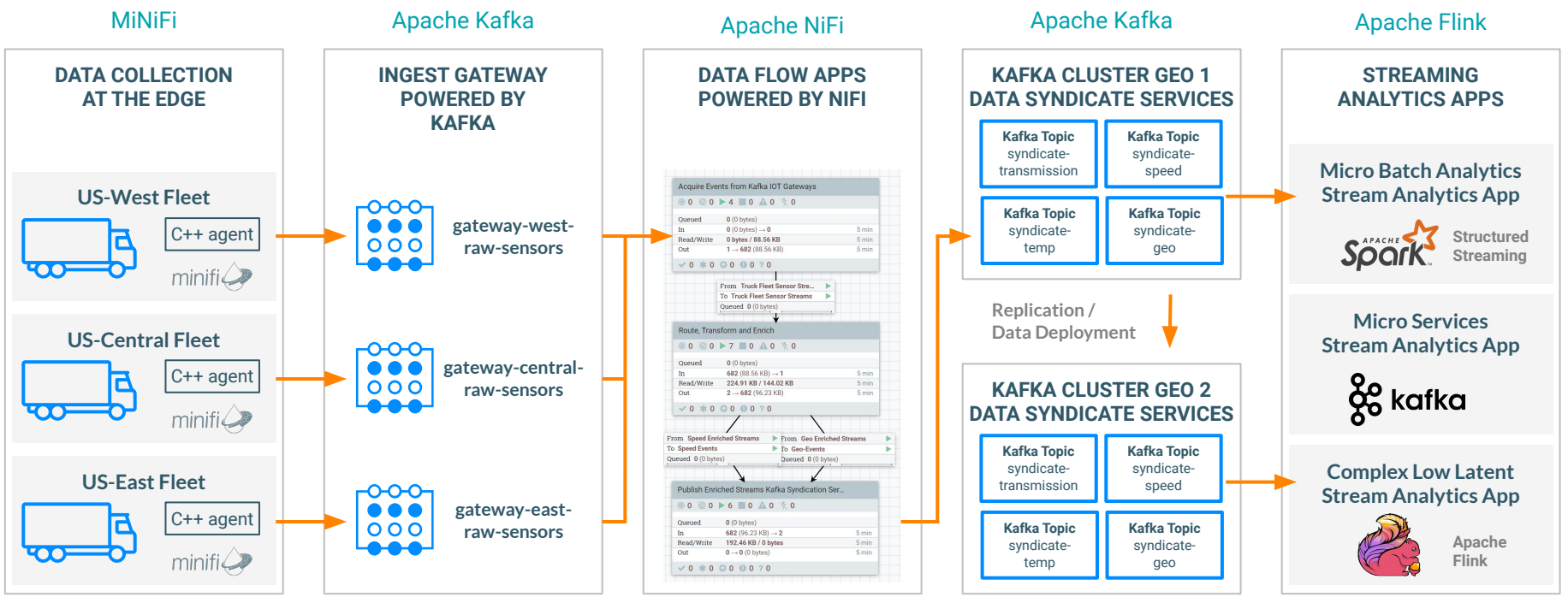

**Flow Monitoring**




- Central Command and Control (C2)
- Design and deploy to millions of agents
- Edge Applications lifecycle management
- Multitenancy with Agent classes
- Native integration with other CDF services

# APACHE KAFKA

# Apache Kafka

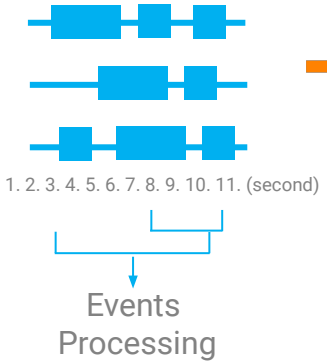


# APACHE FLINK

# DELIVERING STREAMING ANALYTICS

**Capture Events that Matter**

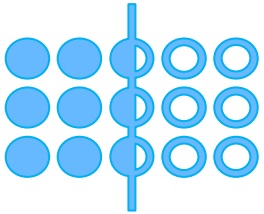
Low-latency analytics use cases




+

**Parsing and Blending Data**

Both offline and streaming data



+

**Data Analysts Can Write Queries**

Across the Lines of Business

SQL

=

**Streaming Analytics**

# SQL STREAM BUILDER (SSB)

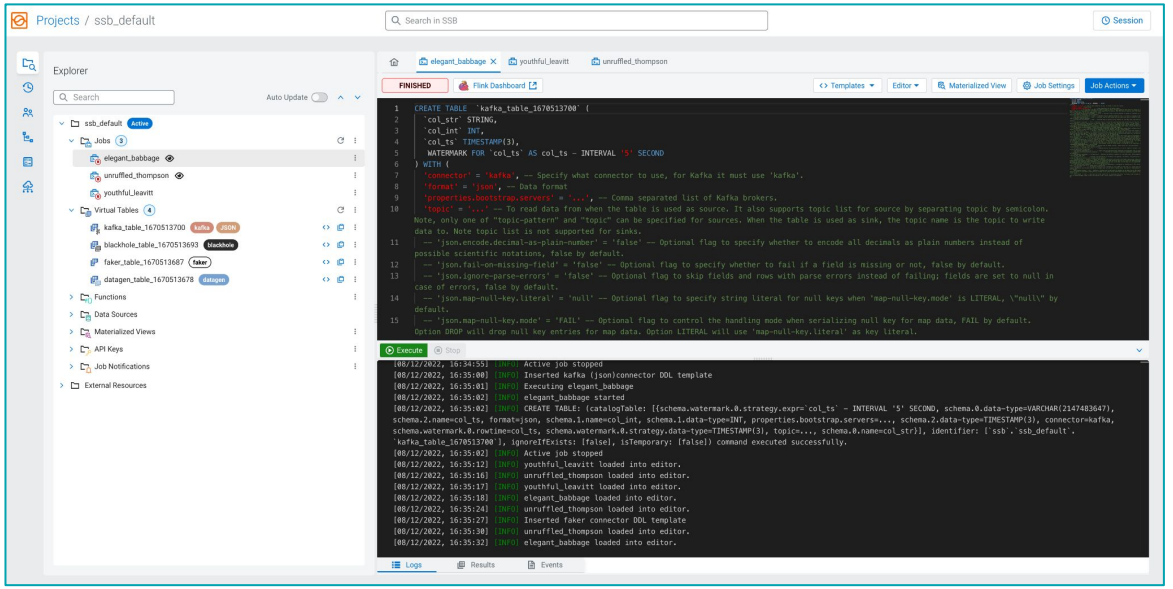
## Democratize access to real-time data with just SQL

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

*No Java or Scala code development required.*

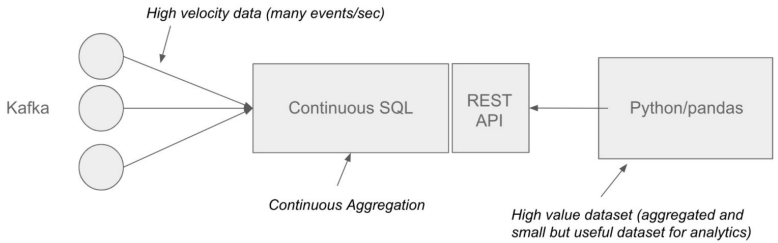
*Simplifies access to data in Kafka & Flink. Connectors to batch data in HDFS, Kudu, Hive, S3, JDBC, CDC and more*

*Enrich streaming data with batch data in a single tool*



# 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
```

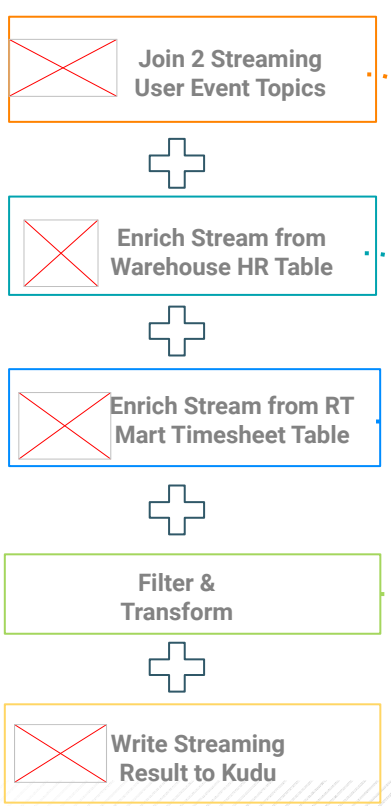
```
[90]: import pandas as pd
[91]: mv = "https://xxxyyyyzzz"
[92]: df = pd.read_json(mv)
[93]: len(df.keys())
[93]: 5
[95]: df['ts'] = pd.to_datetime(df['ts'])
[97]: df.dtypes
[97]: max_amount              int64
      sum_amount              int64
      thecount                int64
      ts             datetime64[ns]
      userid                  int64
      dtype: object
[98]: df.set_index('userid').sort_values(by=['thecount'], ascending=False).head()
```

[98]:

| userid | max_amount | sum_amount | thecount | ts |
|---|---|---|---|---|
| 787 | 34911 | 57304 | 10 | 2020-06-16 19:52:15 |
| 744 | 77407 | 95407 | 9 | 2020-06-16 19:52:15 |
| 78 | 86761 | 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 |

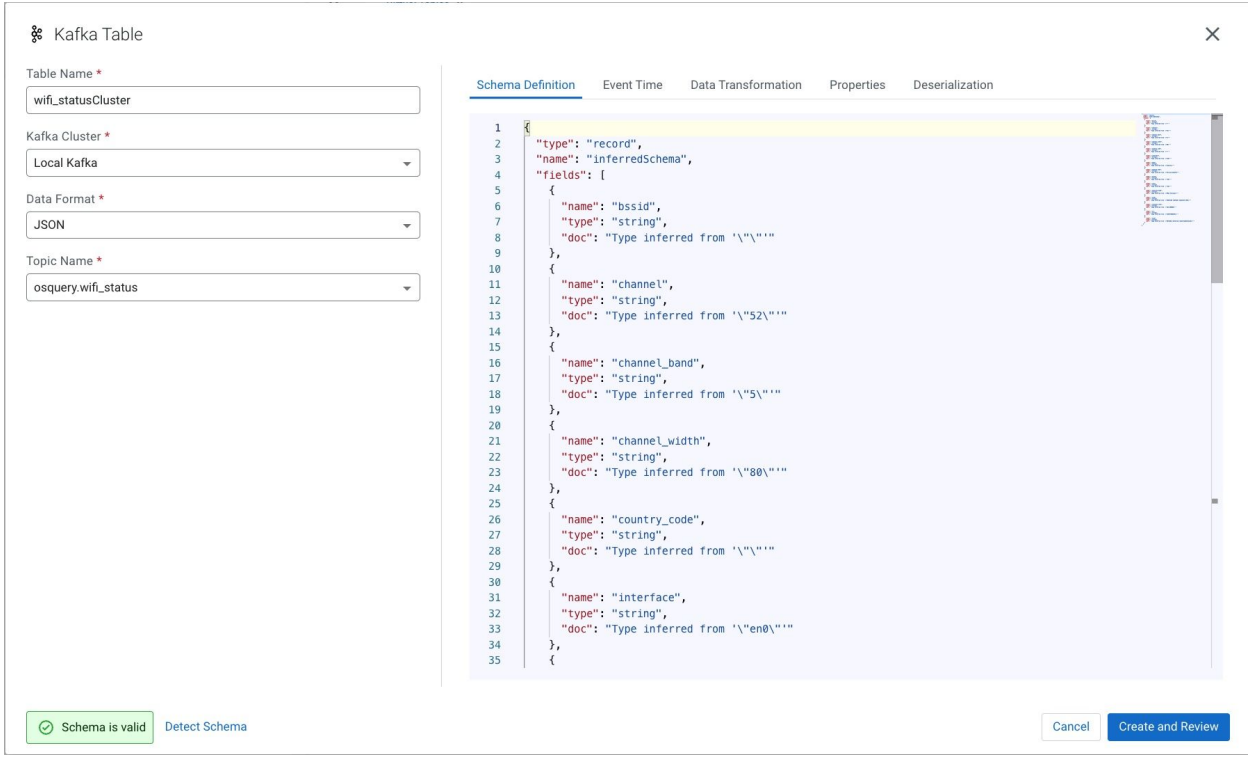
```
[ ]:
```

# Streaming ETL Data Pipeline Made Simple with SQL StreamBuilder



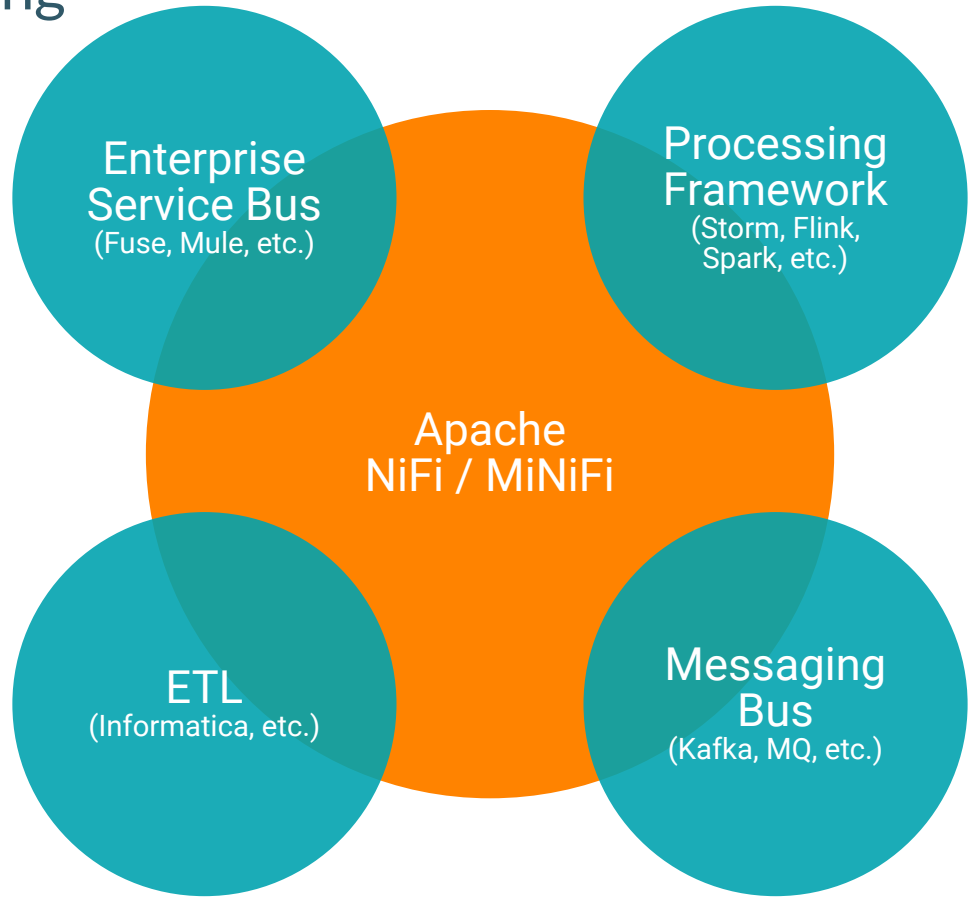

```
SELECT
      geo_event.eventTimestamp, geo_event.driverId, geo_event.eventTime,geo_event.eventSource,
      geo_event.truckId,geo_event.driverName,geo_event.routeId,geo_event.route,geo_event.eventType,
      geo_event.latitude, geo_event.longitude, geo_event.correlationId, geo_event.geoAddress,
      speed_event.speed,
      driver.certified, driver.wage_plan,
      timesheet.hours_logged, timesheet.miles_logged
 FROM
   geo_events_json as geo_event
   join speed_events_json as speed_event
   on (geo_event.driverId = speed_event.driverId)
   left join CDP_Hive_Catalog.employees_hr_hive_db.driver
      FOR  SYSTEM_TIME AS OF PROCTIME() driver
      on driver.driverid = geo_event.driverId
   left join `CDP_Kudu_Catalog`.`default_database`.`impala::employees_hr_kudu_impala_db.timesheet`
      FOR  SYSTEM_TIME AS OF PROCTIME() timesheet
      on (timesheet.driverid = geo_event.driverId  and timesheet_week = 1)
    where
   geo_event.eventTimestamp BETWEEN
      speed_event.eventTimestamp - INTERVAL '1' SECOND AND
      speed_event.eventTimestamp + INTERVAL '1' SECOND
   AND geo_event.eventType <> 'Normal'
   AND driver.wage_plan = 'hours'
   AND timesheet.hours_logged > 45
```

# Infer Kafka Tables from JSON, AVRO, … Data



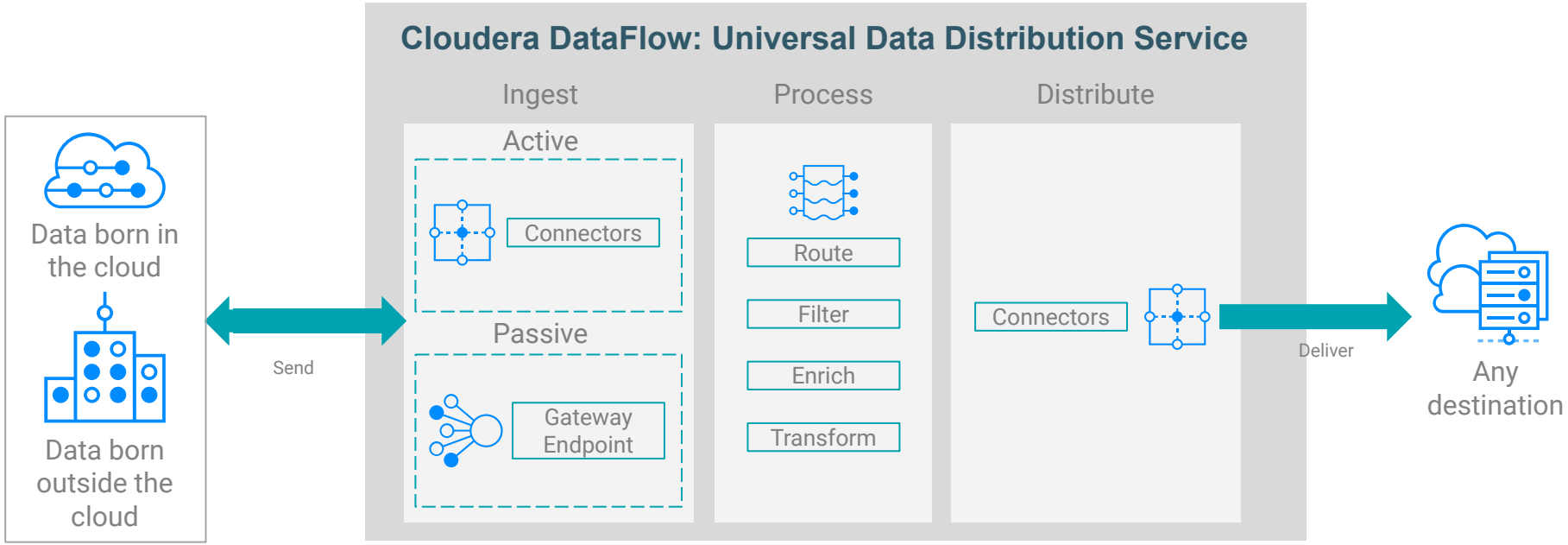
# DATAFLOW APACHE NIFI

# NiFi Positioning



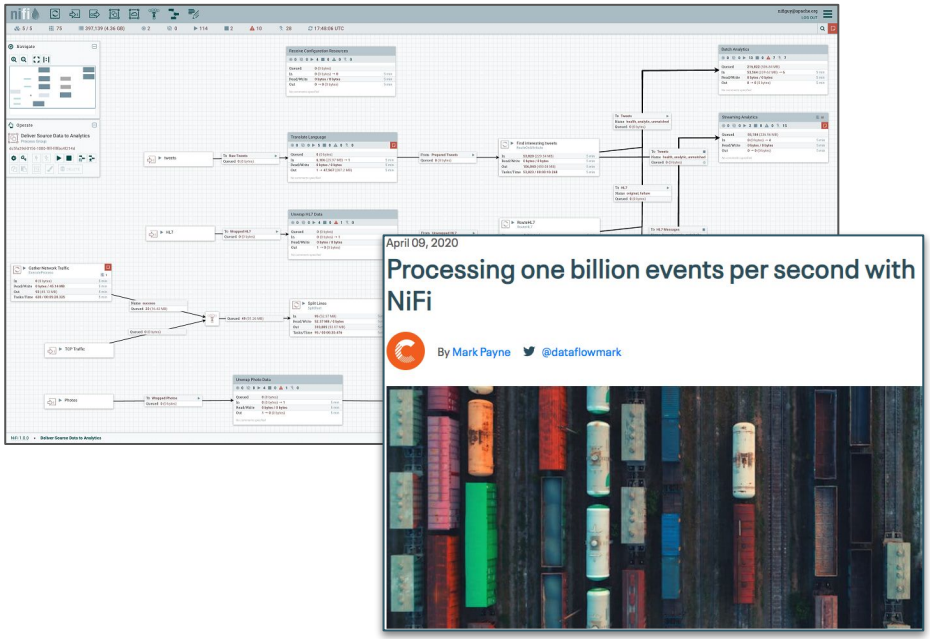
# UNIVERSAL DATA DISTRIBUTION WITH CLOUDERA DATAFLOW (CDF)

## 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

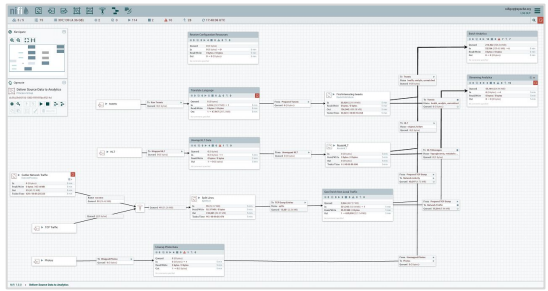



- #1 data ingestion/movement engine
- Strong community
- Product maturity over 11 years
- Deploy on-premises or in the cloud
- Over 400+ pre-built processors
- Built-in data provenance
- Guaranteed delivery
- Throttling and Back pressure

# 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

minifi APACHE nifi



APACHE NIFI registry

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

**ACQUIRE**

kafka, HIVE, Spark, HBase, APACHE STORM, Solr, SQL, Flink, DynamoDB, cassandra, amazon KINESIS, elastic, S3, mongoDB, slack

FTP
SFTP
HL7
UDP
XML
HTTP
EMAIL
HTML
IMAGE
SYSLOG

**PROCESS**

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

**DELIVER**

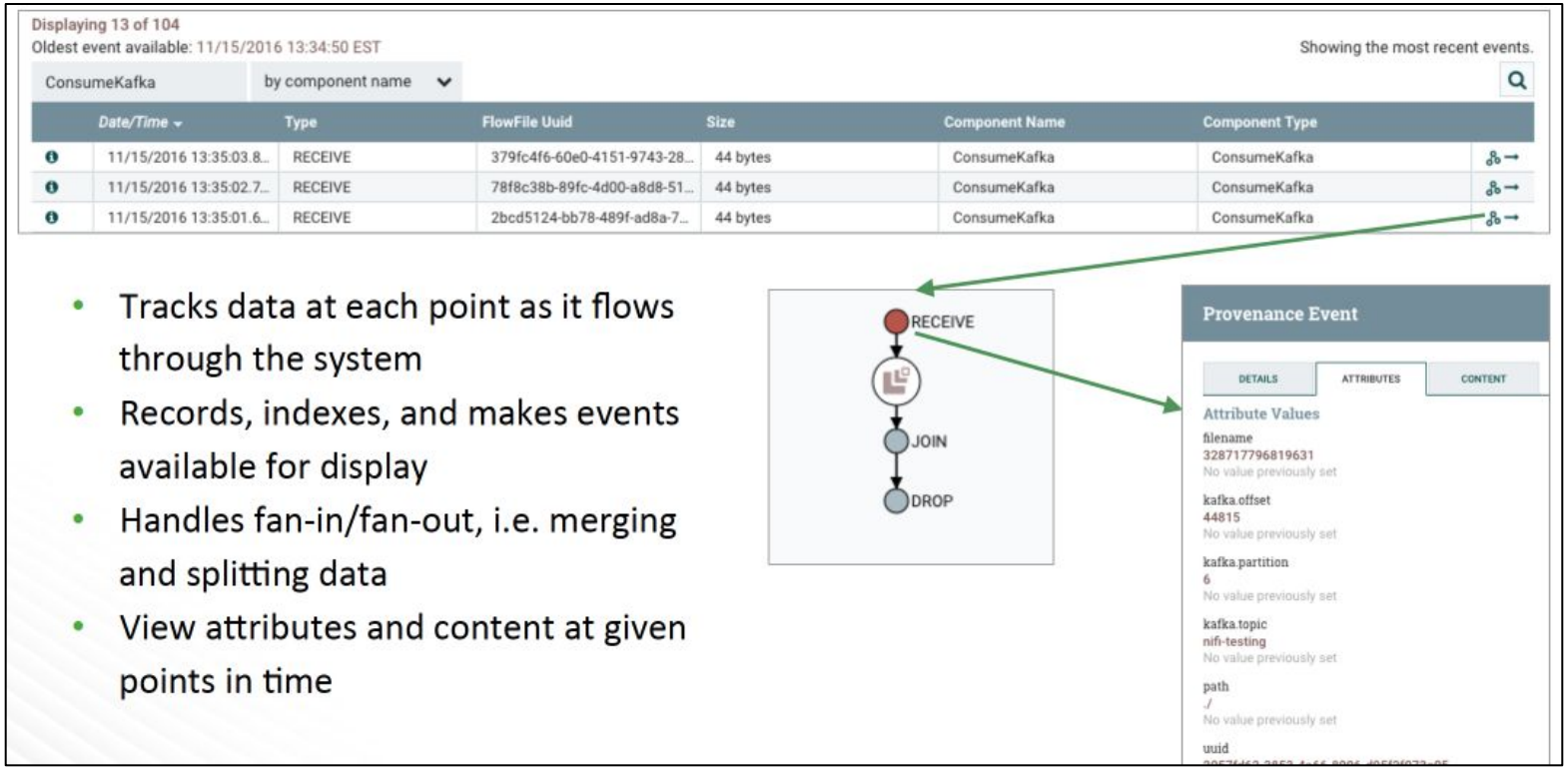
FTP
SFTP
HL7
UDP
XML
HTTP
EMAIL
HTML
IMAGE
SYSLOG

kafka, HIVE, Spark, HBase, APACHE STORM, Solr, SQL, Flink, DynamoDB, cassandra, amazon KINESIS, elastic, S3, mongoDB, slack

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

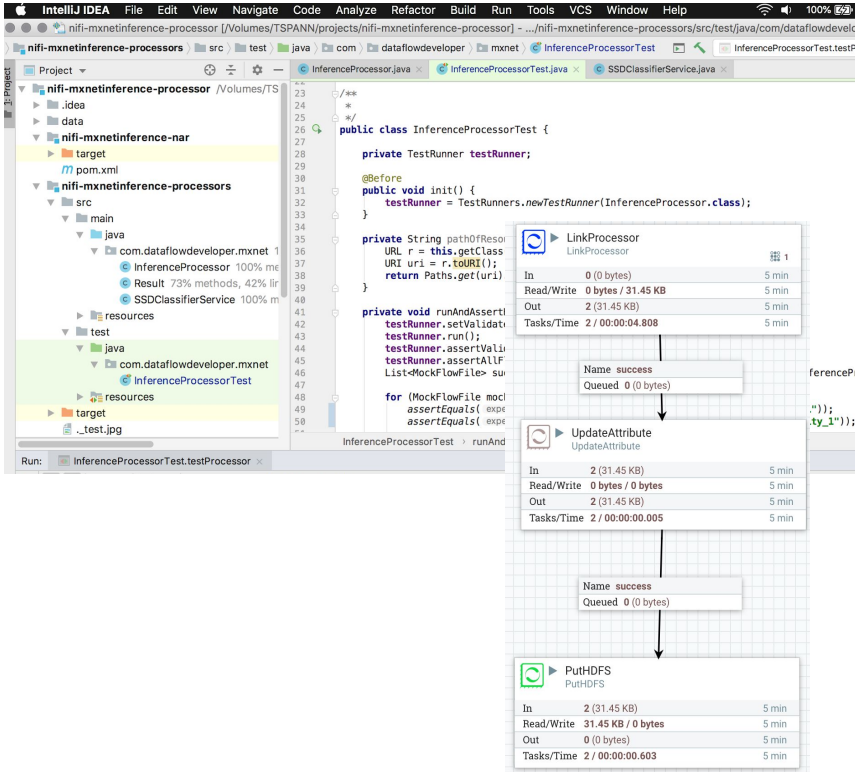
- Guaranteed Delivery
- Full data provenance from acquisition to delivery
- Diverse, Non-Traditional Sources
- Eco-system integration

# PROVENANCE



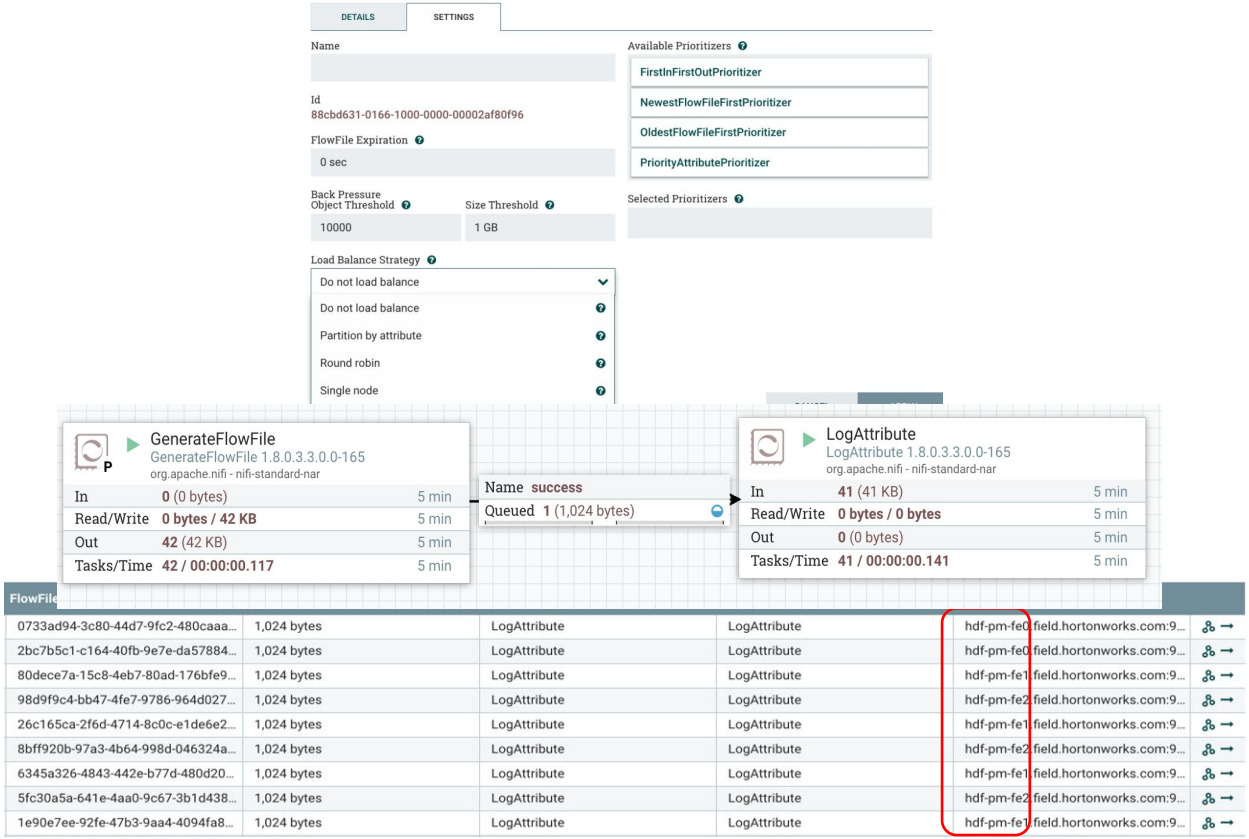
# EXTENSIBILITY

- 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

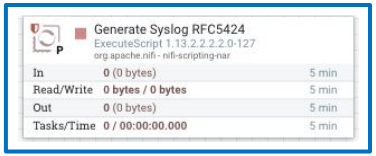


# NiFi Load Balancing

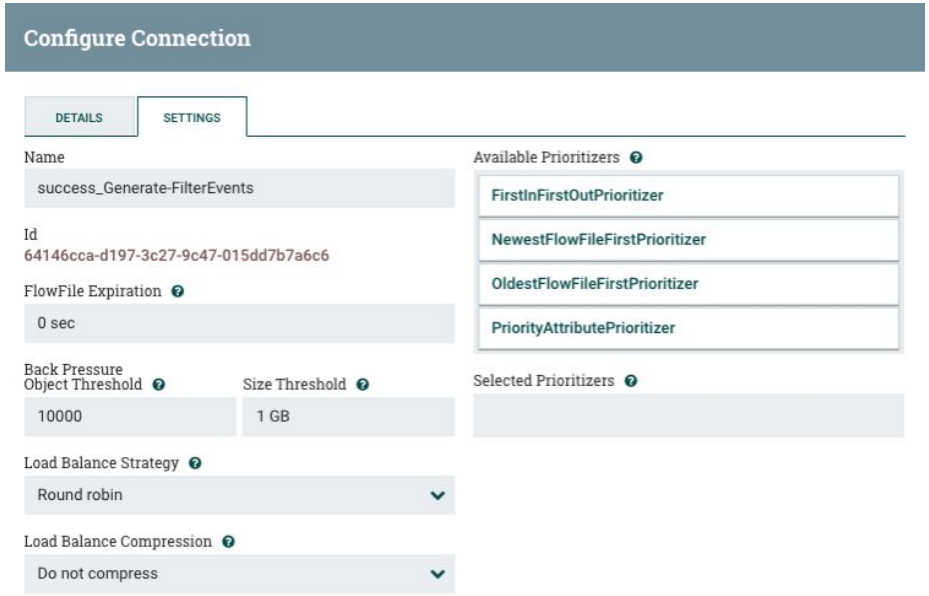
- Improve NiFi cluster throughput
- Defined at connection level
- Configurable balancing strategies
- Critical for scale up paradigm in Kubernetes
- Alleviates S2S balancing “hack” customers use



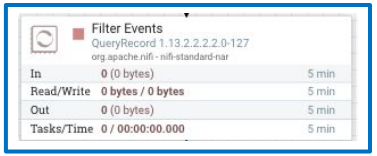
# QUEUE CONFIGURATION




- **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.

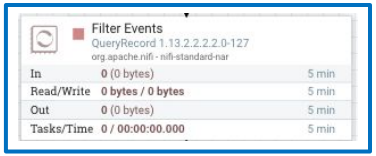


# RECORD-ORIENTED DATA WITH NIFI

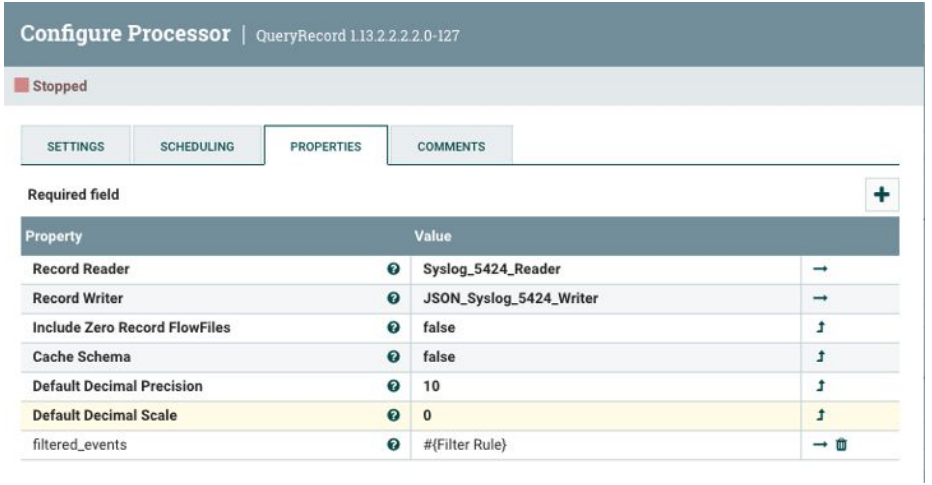



- **Record Readers** - Avro, CSV, Grok, IPFIX, JSAN1, JSON, Parquet, Scripted, Syslog5424, Syslog, WindowsEvent, XML
- **Record Writers** - Avro, CSV, FreeFromText, Json, Parquet, Scripted, XML
- Record Reader and Writer support referencing a schema registry for retrieving schemas when necessary.
- Enable 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.

# RUNNING SQL ON FLOWFILES



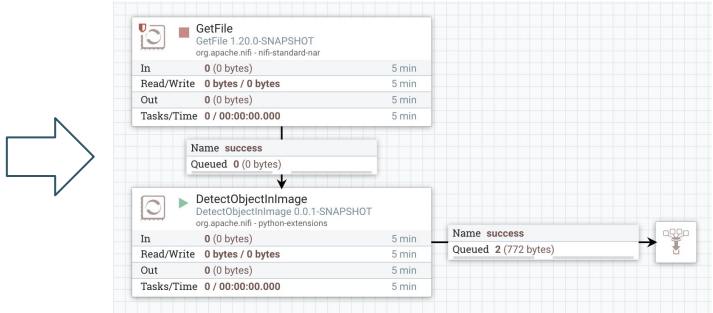

- Evaluates one or more SQL queries against the contents of a FlowFile.
- This can be used, for example, for field-specific filtering, transformation, and row-level filtering.
- Columns can be renamed, simple calculations and aggregations performed.
- The SQL statement must be valid ANSI SQL and is powered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



# 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), ' +
                          'DLDT (*.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:
```




https://www.youtube.com/watch?v=9Oi_6nFmbPg
https://github.com/apache/nifi/blob/614947e4ac6798ad80817e82514c39349d5faacb/nifi-docs/src/main/asciidoc/python-developer-guide.adoc

## Add Processor

Source

all groups

amazon attributes aws azure cloud database fetch get google hadoop ingest json listen logs message microsoft put python record restricted source storage test text update

Displaying 13 of 333

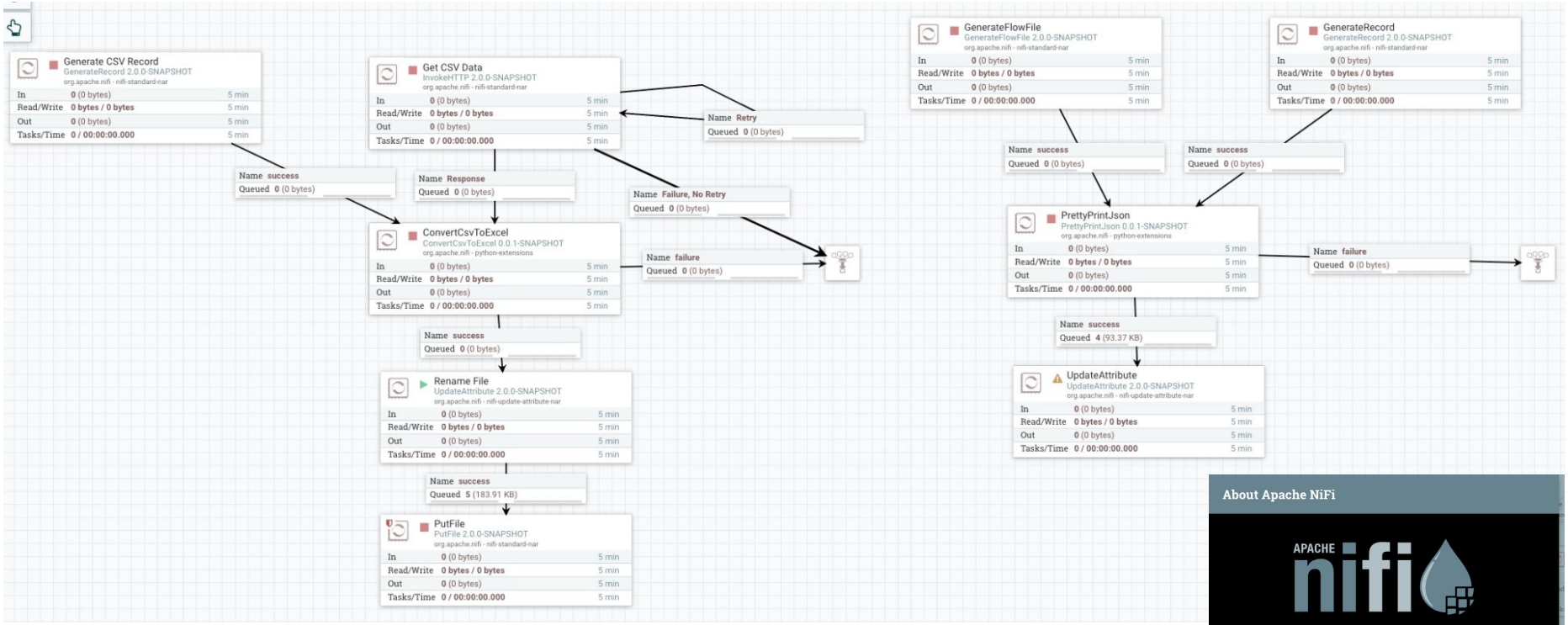
beta

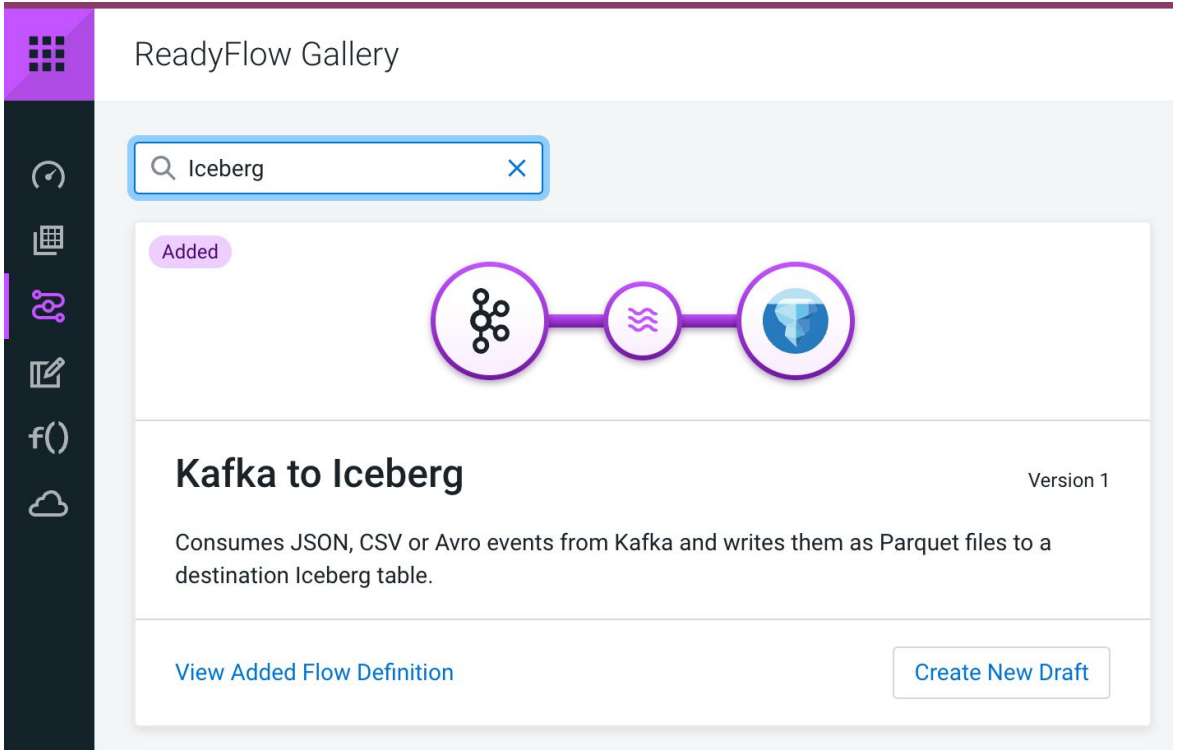
| Type | Version | Tags |
|---|---|---|
| ConvertCsvToExcel | 0.0.1-SNAPSHOT | excel, python, test, csv, beta |
| DetectObjectInImage | 0.0.1-SNAPSHOT | python, test, beta |
| GenerateRecord | 0.0.1-SNAPSHOT | python, test, beta |
| LogContents | 0.0.1-SNAPSHOT | python, test, beta |
| LookupAddress | 0.0.1-SNAPSHOT | python, test, beta |
| PopulateRecord | 0.0.1-SNAPSHOT | python, test, beta |
| PrettyPrintJson | 0.0.1-SNAPSHOT | python, test, beta |
| SetRecordField | 0.0.1-SNAPSHOT | python, test, beta |
| WriteMessage | 0.0.1-SNAPSHOT | python, test, beta |
| WriteMessage | 0.0.2-SNAPSHOT | python, test, beta |
| WriteNumber | 0.0.1-SNAPSHOT | python, test, beta |
| WriteNumpyVersion | 0.0.1-SNAPSHOT | python, test, beta |

**ConvertCsvToExcel 0.0.1-SNAPSHOT** org.apache.nifi - python-extensions

Converts a CSV file into a Microsoft Excel file

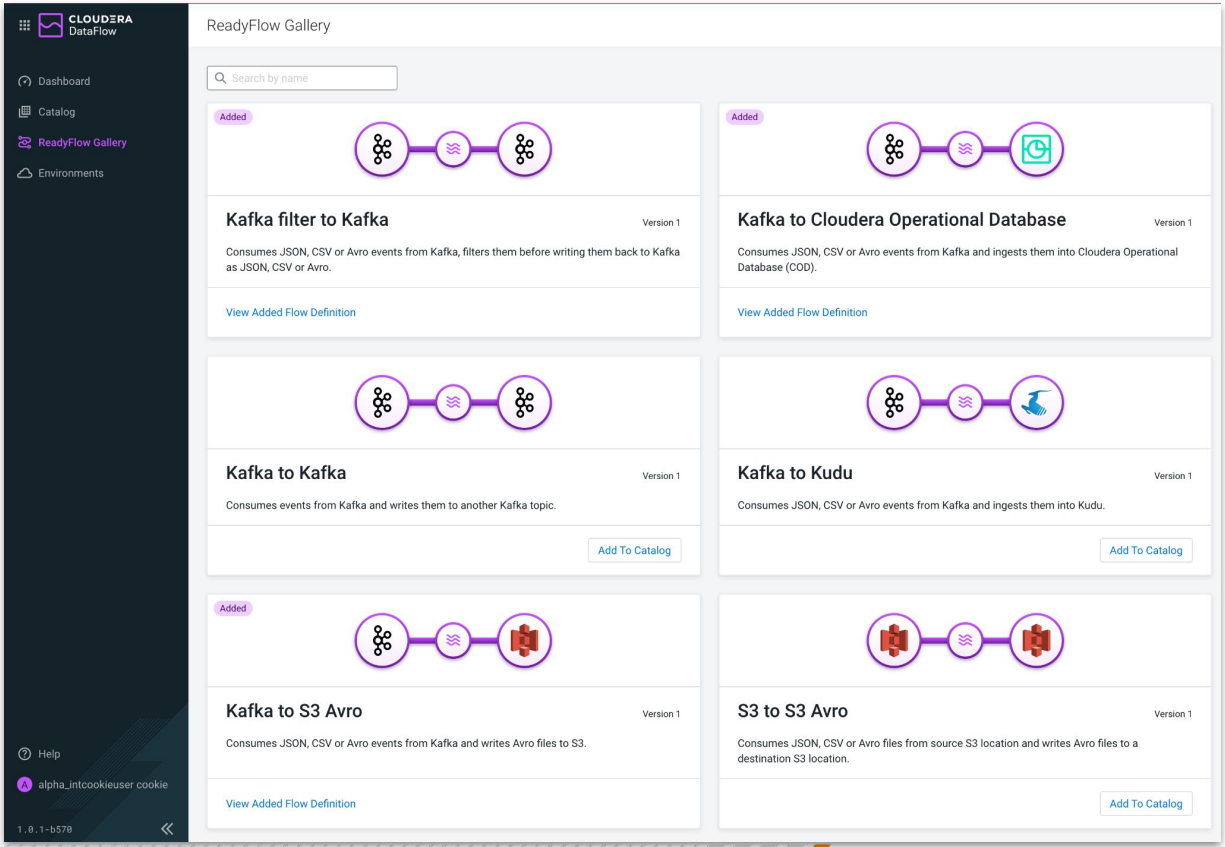
CANCEL ADD



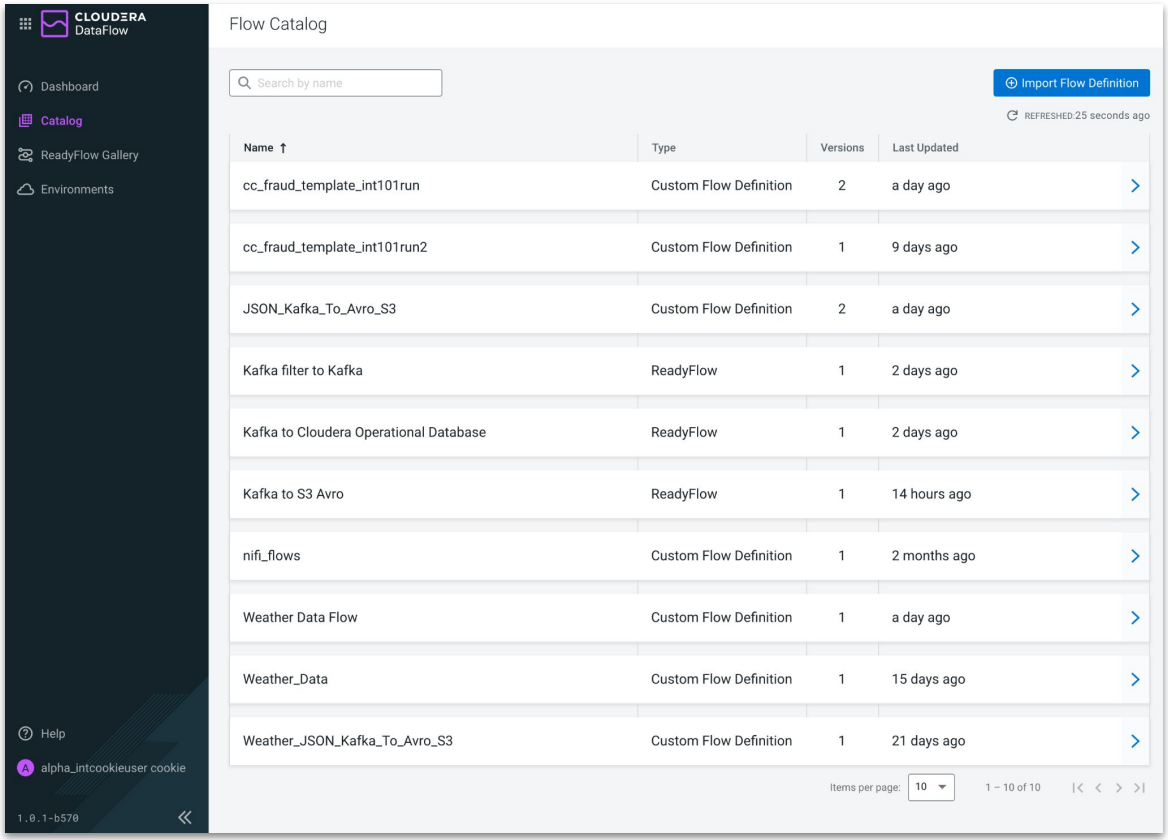
# READYFLOW GALLERY

- 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



# FLOW CATALOG

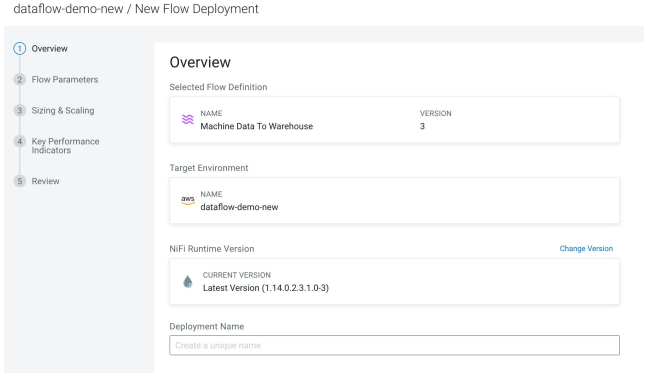
- Central repository for flow definitions
- Import existing NiFi flows
- Manage flow definitions
- Initiate flow deployments



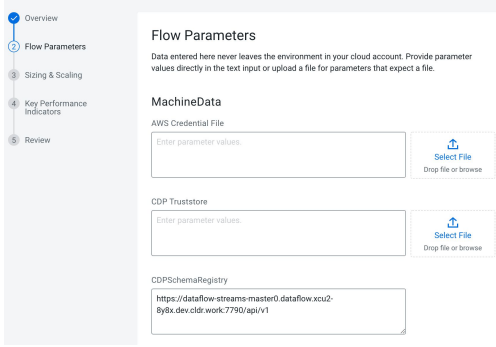
# DEPLOYMENT WIZARD

- 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

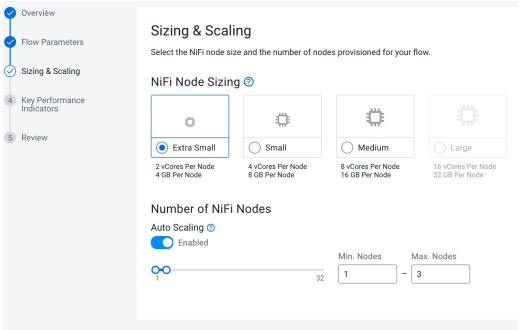
## Start Deployment Wizard



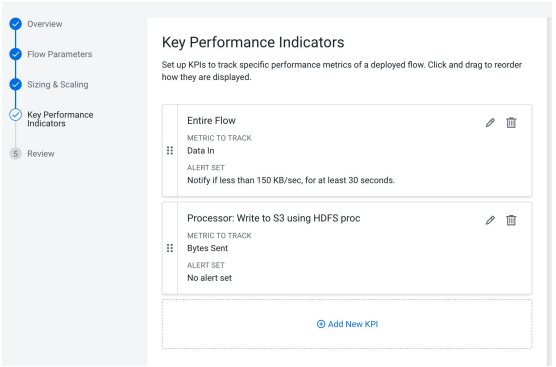

## Provide Parameters




## Configure Sizing & Scaling



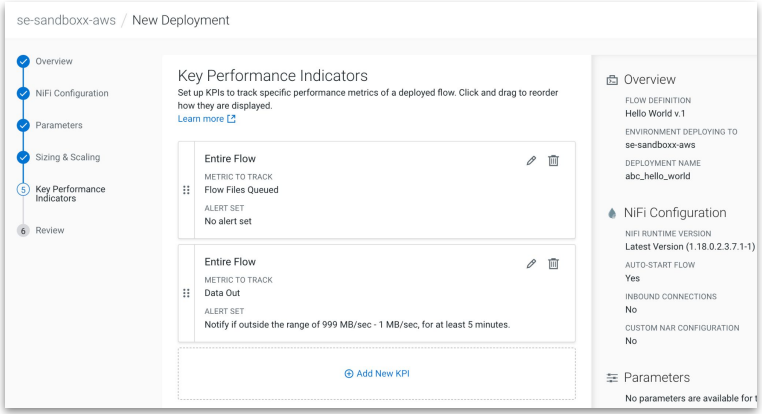

## Define KPIs



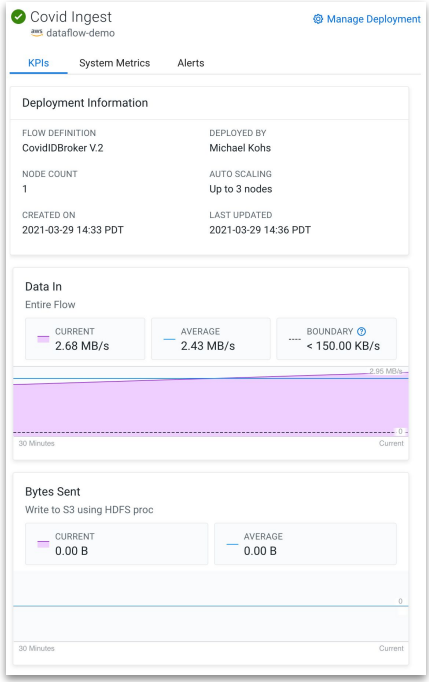
# KEY PERFORMANCE INDICATORS

- 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

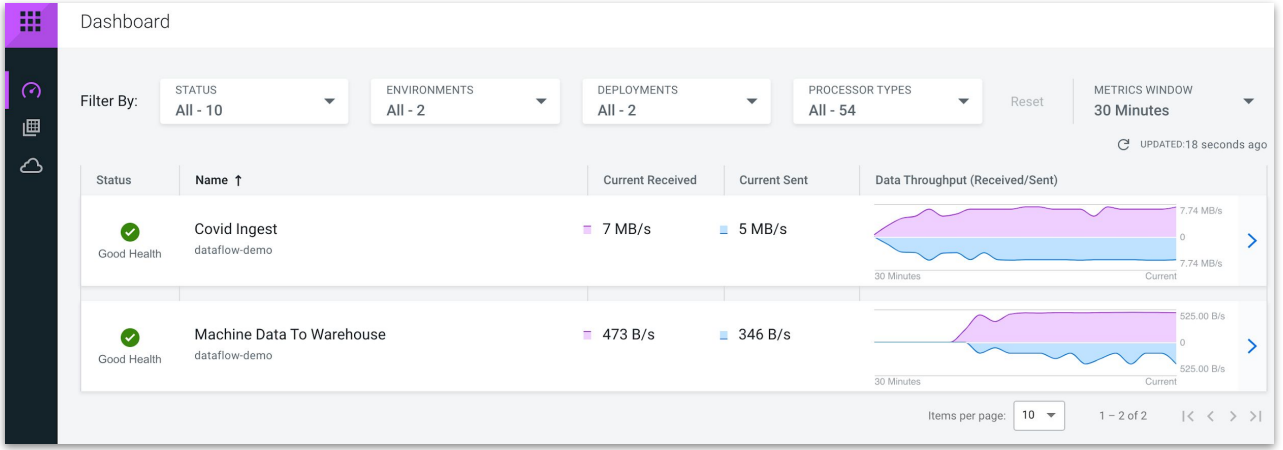


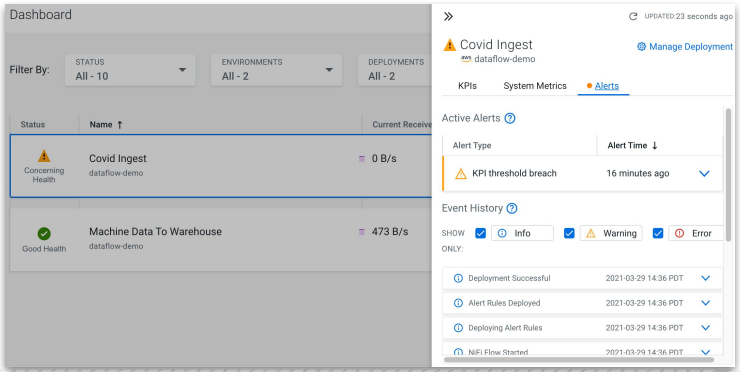

## KPI Monitoring

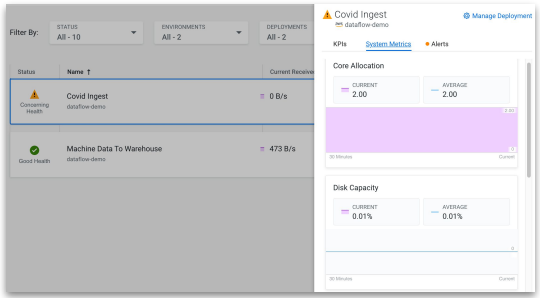


# DASHBOARD

- 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

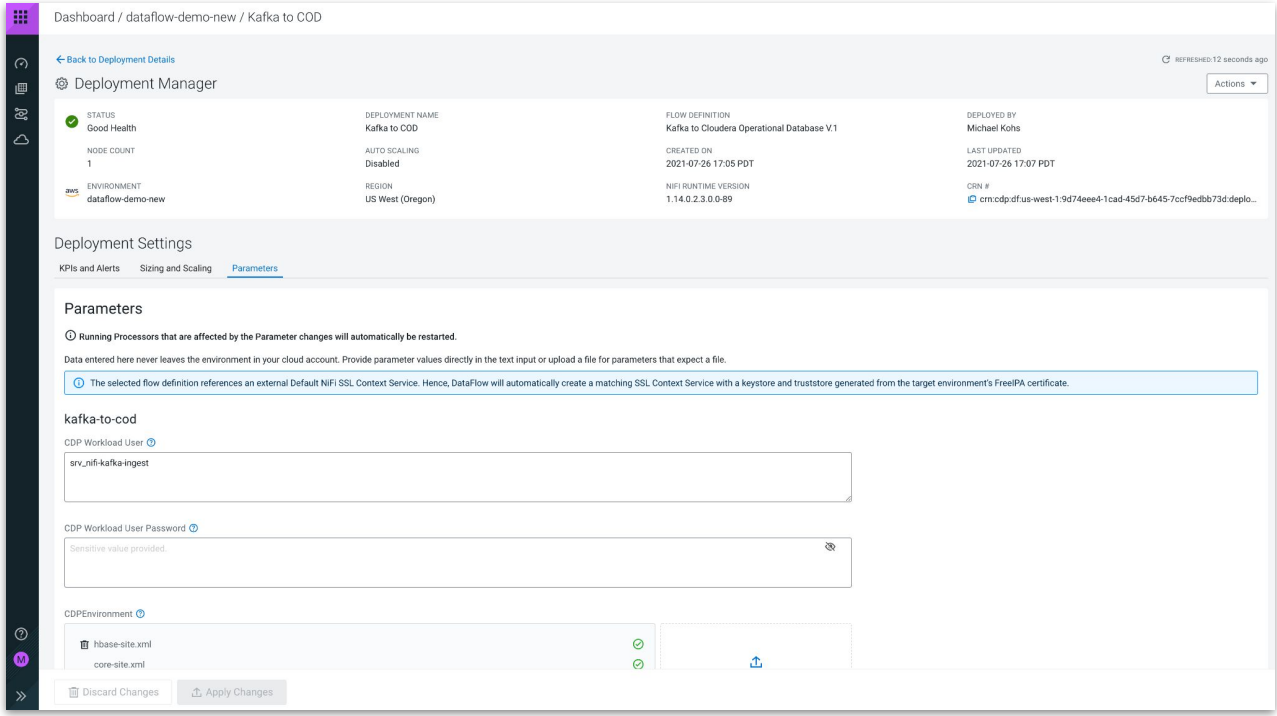


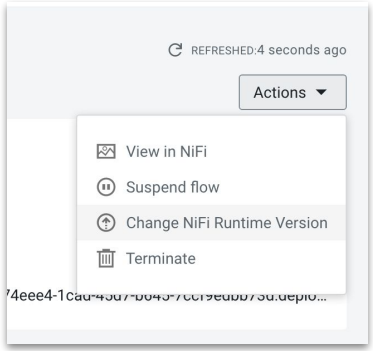
# DEPLOYMENT MANAGER

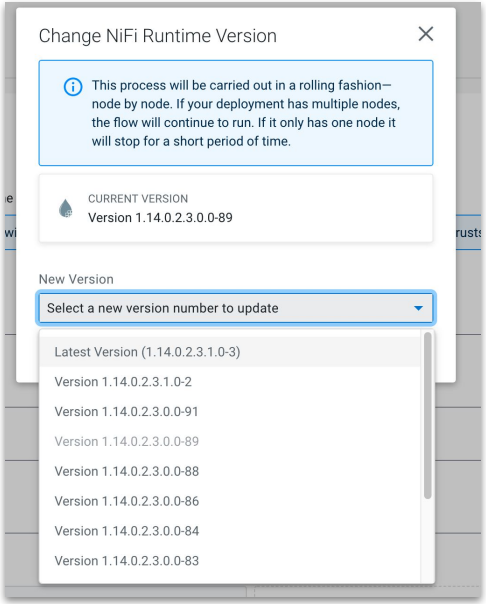
- Manage flow deployment lifecycle (Suspend/Start/Terminate)
- Add/Edit KPIs
- Change sizing configuration
- Update parameters
- Change NiFi version of the deployment
- Gateway to NiFi canvas



# 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)



# Apache NiFi - Release History Review

- *Soon: NiFi 2.0*
- *Apr 23: 1.21*
- Feb 23: 1.20

- Non-stop security improvements, vulnerability resolution, library updates. “Security by default”. Secure Zookeeper.

- Dec 22: 1.19
- Oct 22: 1.18
- Aug 22: 1.17
- Mar 22: 1.16

- Key capabilities: all repository on-disk encryption, classloader isolation for native libs, empty all queues, prometheus, Azure AD authz, ‘Run Once’, Framework level retries, Stateless NiFi in NiFi, Parameter Contexts with inheritance and sensitive values, interactive component validation, generate fake records, scripted record transforms

- Dec 21: 1.15
- Jul 21: 1.14
- Mar 21: 1.13

- New connectors for: IBM MQ, GCP, AWS, Azure, AMQP, MQTT, CDC/MySQL, Kudu, CEF, Geo Hashing, SNMP Traps, SMB, Airtable, Dropbox, Box, Hubspot, Workday, Zendesk, Iceberg, Snowflake, Salesforce, Google Drive

- Aug 20: 1.12
- Jan 20: 1.11

# NiFi 2.0 - Quick Look

- Moves to Java 11 and Java 17 as the base
- JSON for flow serialization
  - XML and Templates concepts gone
- Parameter Contexts for easier SDLC/Deployment
  - Variables removed.
- Technical Debt Reduction
  - Delete a bunch of legacy components.
- Focus on timer and cron based scheduling
  - Event driven scheduling removed / unnecessary.

# Key / Common Themes Of Change beyond the framework

## Stateless NiFi engine

- Faster execution (no disk)
- End to end transactional
- Run in FaaS or NiFi or command line

## Record Oriented Processors

- The proven best practice pattern for handling any ‘record’ pattern
- Same proven processors - just bring your format/schema

## Scripting / Python

- Data Engineers and Data Science user base growing
- Existing script processors always improving - heavy use.

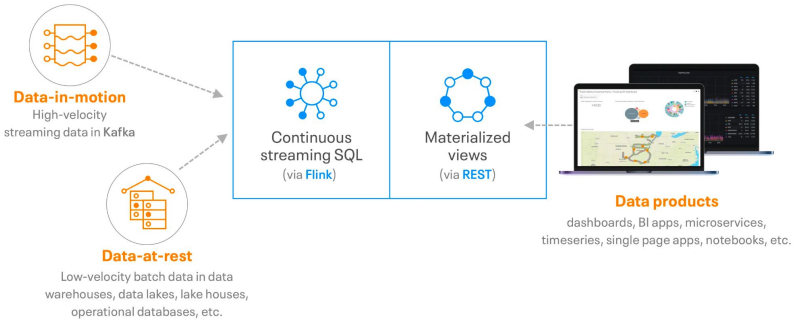
## All the connectors!

- Apache NiFi includes **362** processors by default with more available in Maven
- 132 controller services
- 18 Reporting Tasks

# BEST PRACTICES

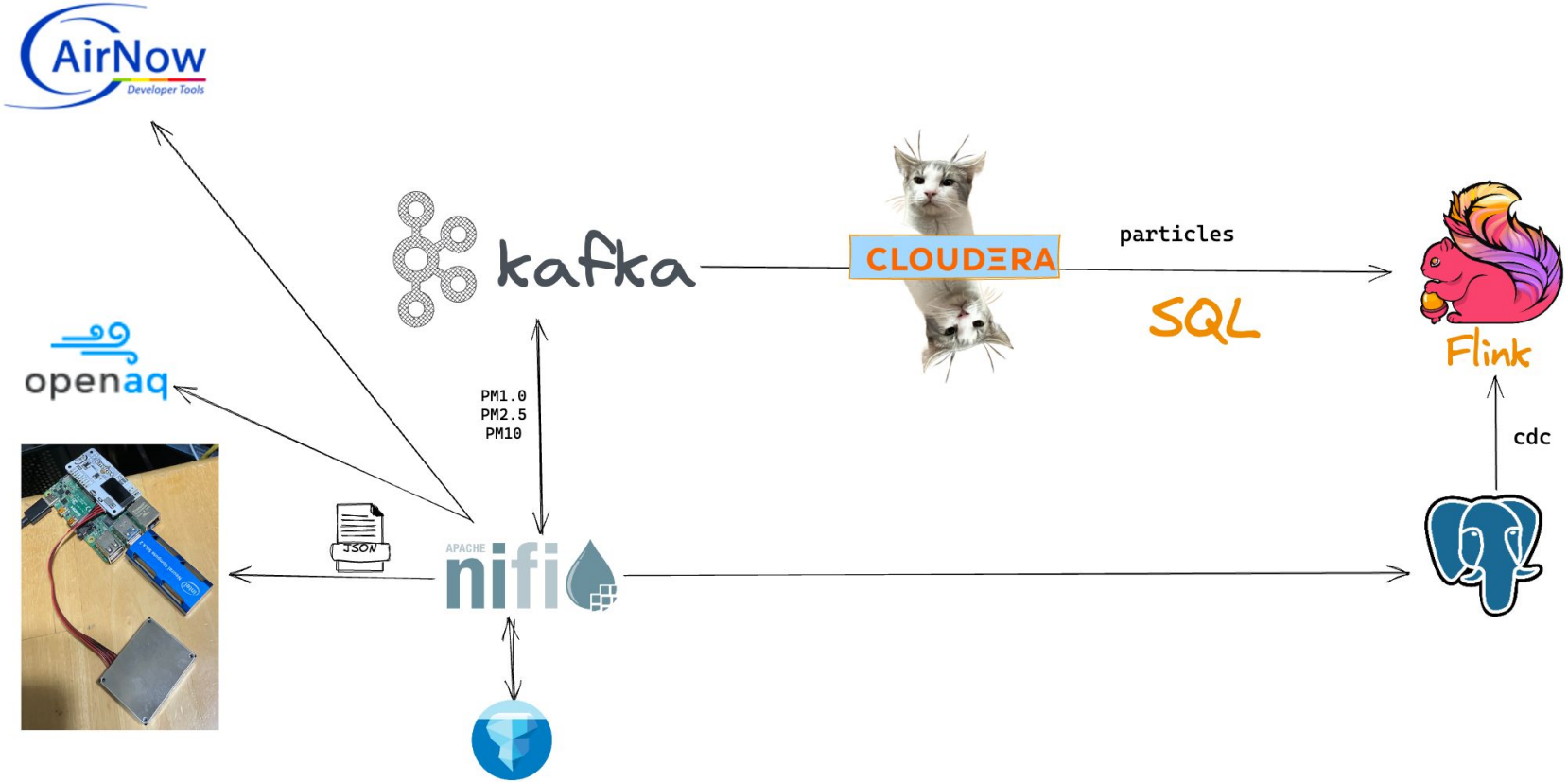
# STREAMING TECH DEBT TIPS

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



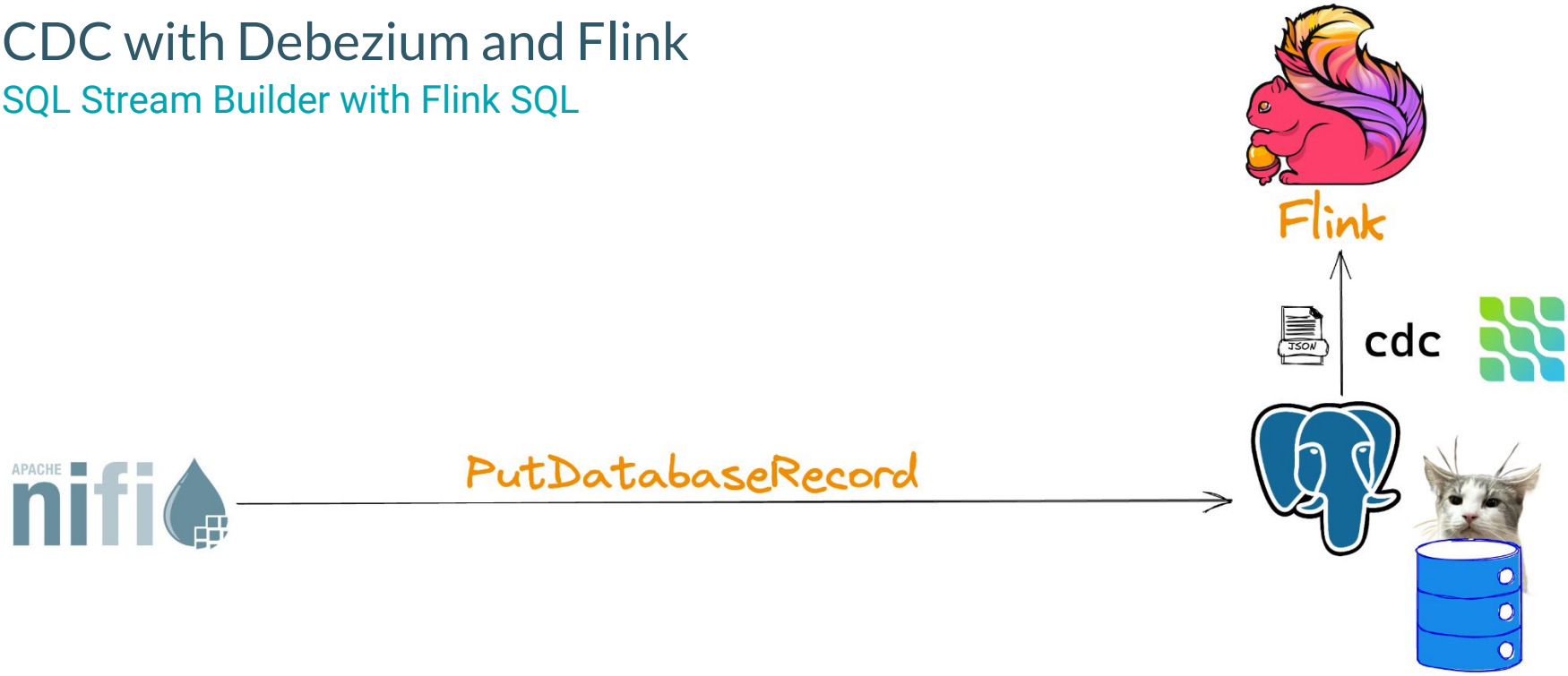

# EXAMPLES

# Particulate Matter Sensor Pipeline



# CDC with Debezium and Flink

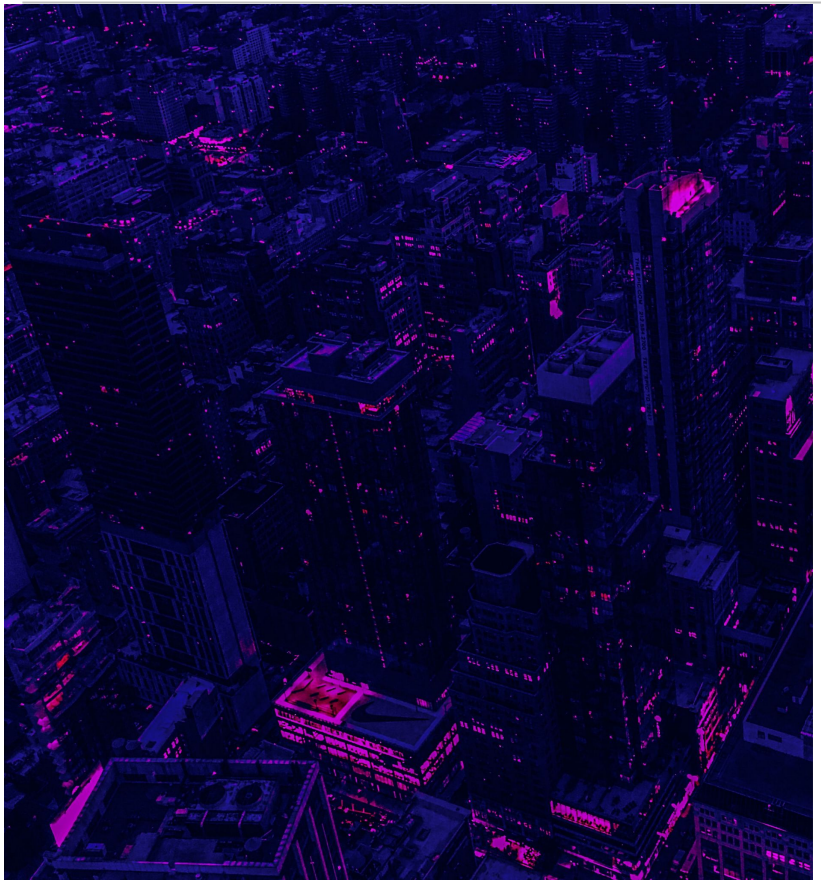
## SQL Stream Builder with Flink SQL

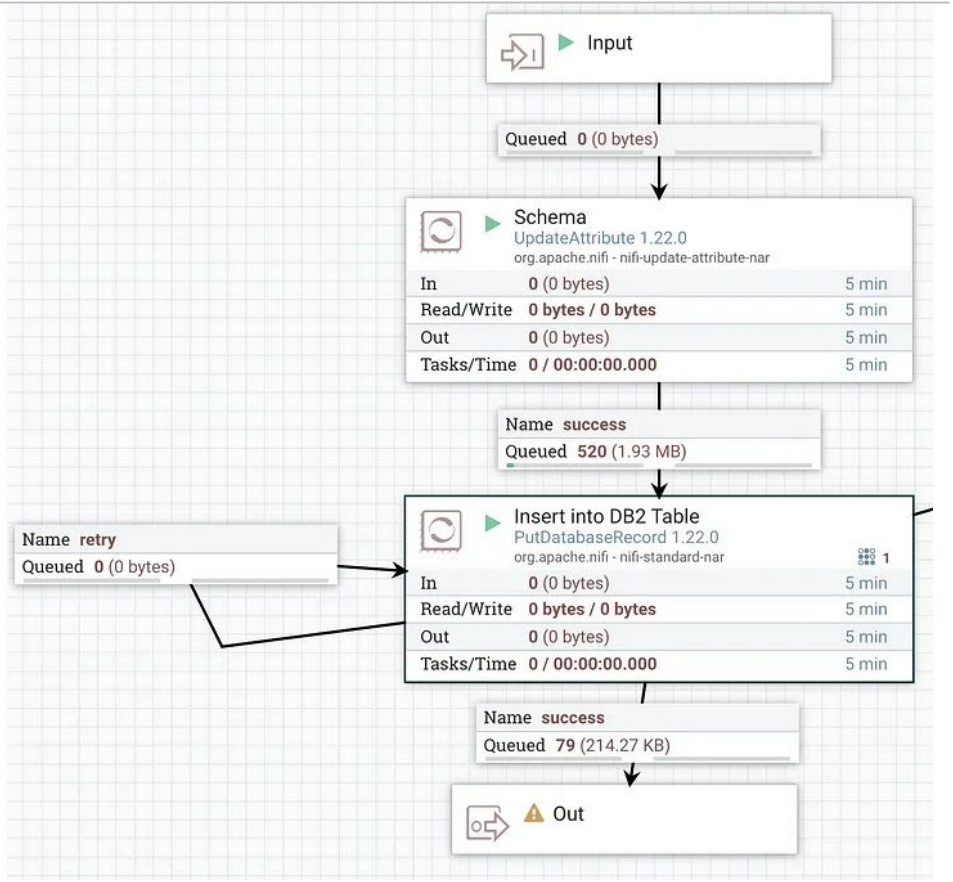


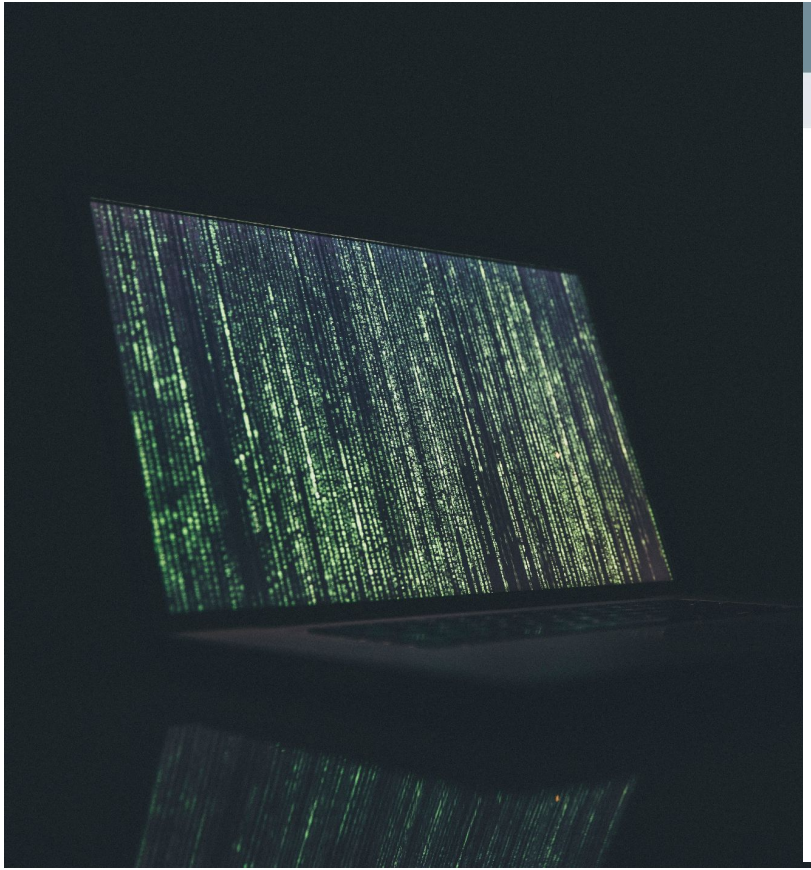

https://docs.cloudera.com/csa/1.10.0/how-to-ssb/topics/csa-ssb-cdc-connectors.html

# CDC with Debezium and Kafka

## Kafka Connect







## Configure Processor | PutDatabaseRecord 1.22.0

Stopped

SETTINGS | SCHEDULING | PROPERTIES | RELATIONSHIPS | COMMENTS

**Required field**

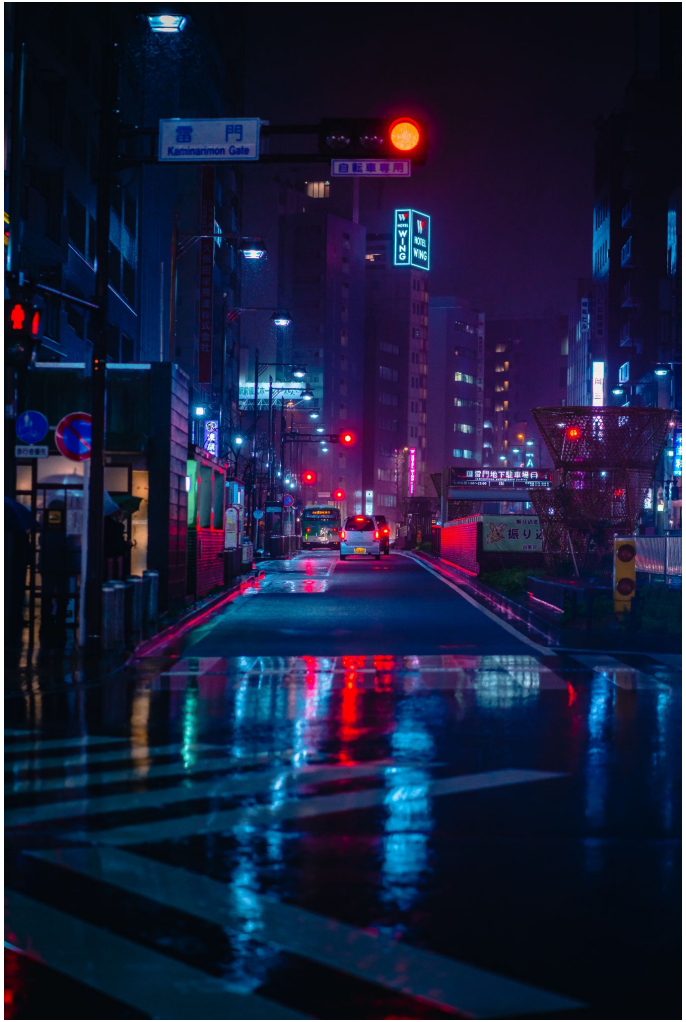
| Property | Value |
|---|---|
| **Record Reader** | **JsonTreeReaderSchema** |
| Database Type | Generic |
| **Statement Type** | **INSERT** |
| Data Record Path | No value set |
| **Database Connection Pooling Service** | **ibmdb2cluster** |
| Catalog Name | No value set |
| Schema Name | DB2INST1 |
| **Table Name** | **TRAVELADVISORY** |
| Translate Field Names | true |
| Unmatched Field Behavior | Ignore Unmatched Fields |
| Unmatched Column Behavior | Ignore Unmatched Columns |
| Quote Column Identifiers | false |

## Controller Service Details | DBCPConnectionPool 1.22.0

SETTINGS | PROPERTIES | COMMENTS

**Required field**

| Property | Value |
|---|---|
| **Database Connection URL** | **jdbc:db2://192.168.1.157:50000/SAMPLE** |
| **Database Driver Class Name** | **com.ibm.db2.jcc.DB2Driver** |
| Database Driver Location(s) | /Users/tspann/Downloads/db2/db2jcc4.jar |
| Kerberos User Service | No value set |
| Kerberos Credentials Service | No value set |
| Kerberos Principal | No value set |
| Kerberos Password | No value set |
| Database User | db2inst1 |
| Password | Sensitive value set |
| **Max Wait Time** | **500 millis** |
| **Max Total Connections** | **8** |
| Validation query | No value set |
| Minimum Idle Connections | 0 |
| Max Idle Connections | 8 |



## Processor Details

Running

STOP & CONFIGURE

SETTINGS | SCHEDULING | PROPERTIES | RELATIONSHIPS | COMMENTS

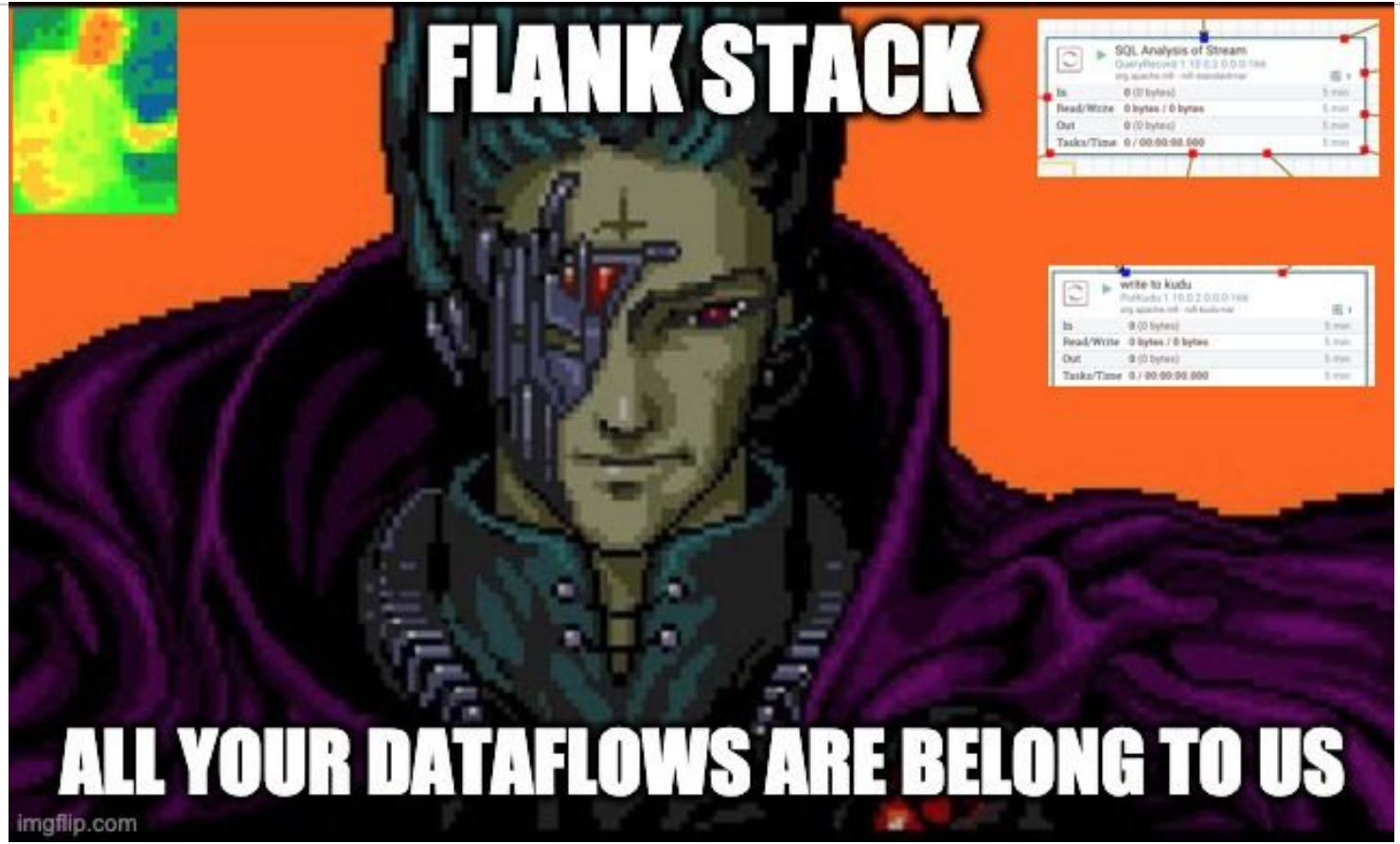
**Required field**

| Property | Value |
|---|---|
| **Record Reader** | **CDP Infer JsonTreeReader** → |
| Database Type | Oracle 12+ |
| **Statement Type** | **INSERT** |
| Data Record Path | No value set |
| **Database Connection Pooling Service** | **Oracle12DBCPConnectionPool** → |
| Catalog Name | No value set |
| Schema Name | TSPANN |
| **Table Name** | **NEWJERSEYBUS** |
| Translate Field Names | true |
| Unmatched Field Behavior | Ignore Unmatched Fields |
| Unmatched Column Behavior | Ignore Unmatched Columns |
| Quote Column Identifiers | false |

OK

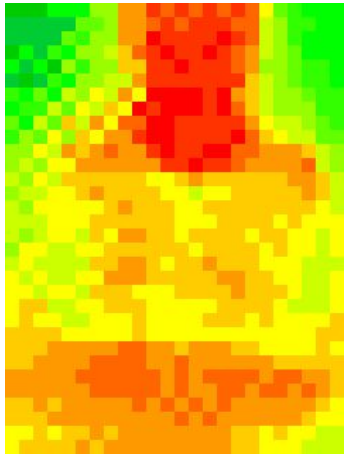
# RESOURCES AND WRAP-UP

# Resources



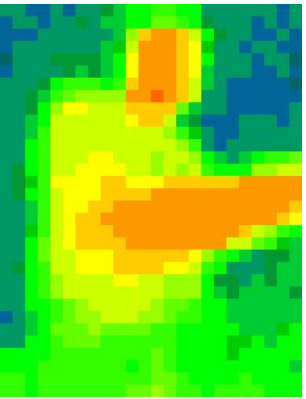
# Upcoming Events

**Nov 22**



https://events.pinetool.ai/3079/#sessions/101077

https://medium.com/@tspann/cdc-not-cat-data-capture-e43713879c03



# THANK YOU