CLOUDERA

Partner SkillUp:
Implement a Universal Data Distribution Architecture to Manage All Streaming Data

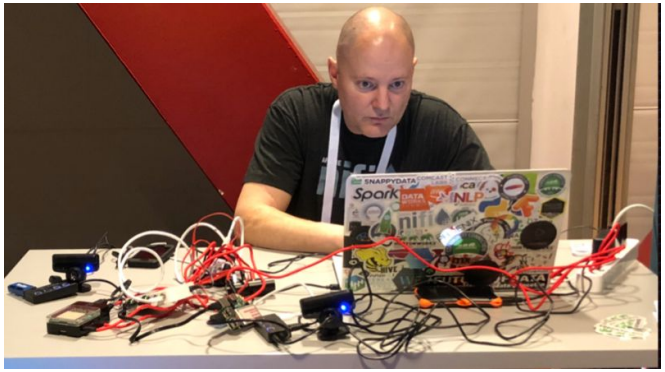
Tim Spann
Principal Developer Advocate in Data In Motion for Cloudera
tspann@cloudera.com

Salvador Almazan
Partner Solutions Engineer, US
salmazan@cloudera.com

Cloudera Data Services Workshop Registration

TODAY'S LEAD

Who am I?



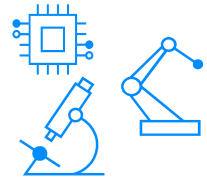
Principal Data-in-Motion Developer Advocate

DZone Zone Leader and Big Data MVB

Princeton NJ Future of Data Meetup
ex-Pivotal Field Engineer ex-StreamNative ex-PwC

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

CHALLENGES IN DATA COLLECTION & MOVEMENT



specialized tools

vendor specific data aggregation hubs & interfaces

vendor specific monitoring & management tools

Multiple Point Solutions Required

Data Architects & Executives

edge devices

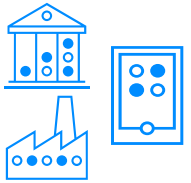
```
while running:
    consumer.subscribe([args.topic])

    msg = consumer.poll(1)
    if msg is None:
        continue
```

custom code to collect data from edge devices

custom management tools

Lack Agility

App/Integration Developers



public & private sector data

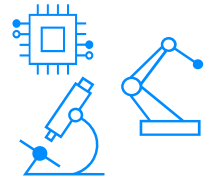
public or **private clouds**

cloud service provider specific data flow monitoring, management & security tools

Operational Challenges

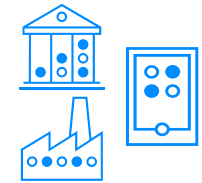
Platform/Flow Administrators

SOLUTION: UNIVERSAL DATA DISTRIBUTION



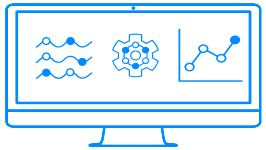
specialized tools

edge devices



public & private sector data

multi-public & hybrid clouds



Consistent data flow design, deployment, management & monitoring.
Any data. Any source. Any destination.

Standardize development tools for reusability.

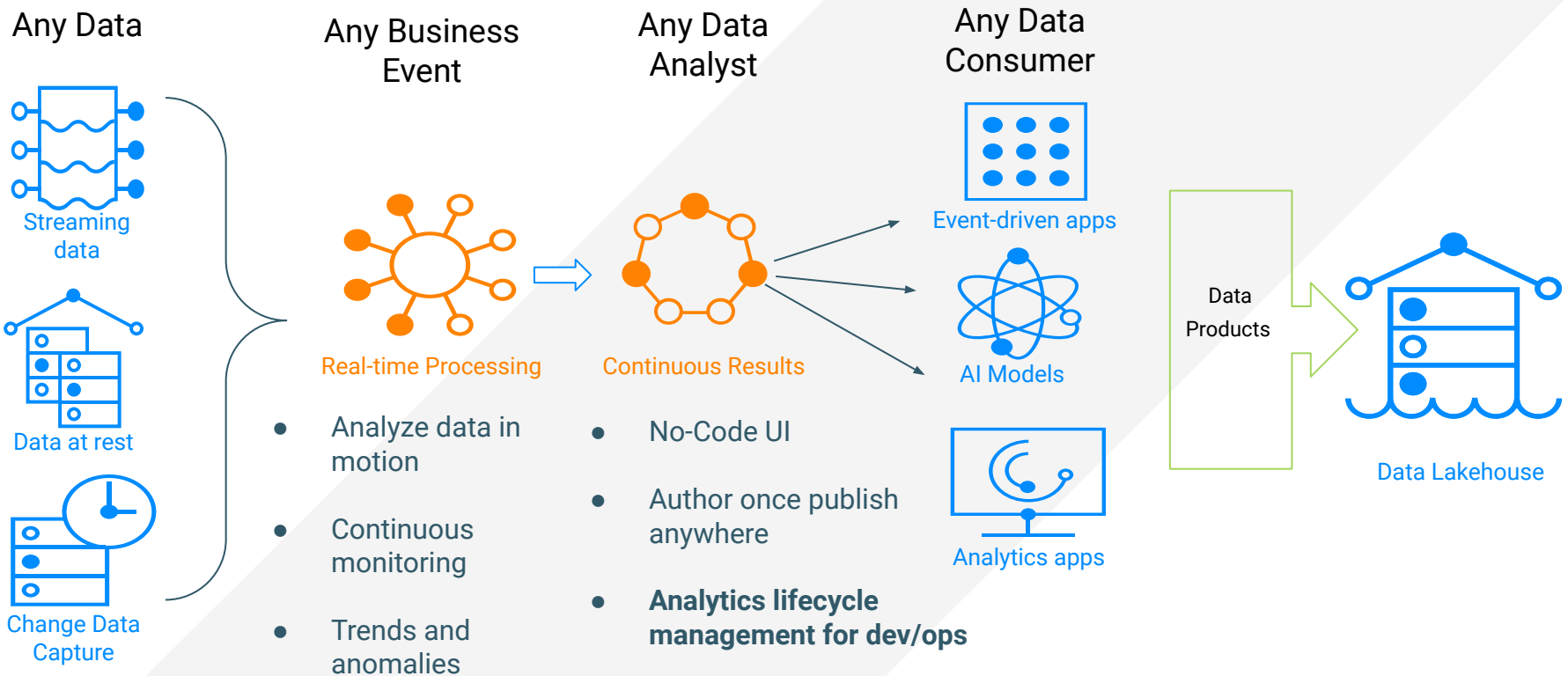
Data Architects & Executives

Simplify & **Accelerate** onboarding and managing data sources. Use no code tools.

App/Integration Developers

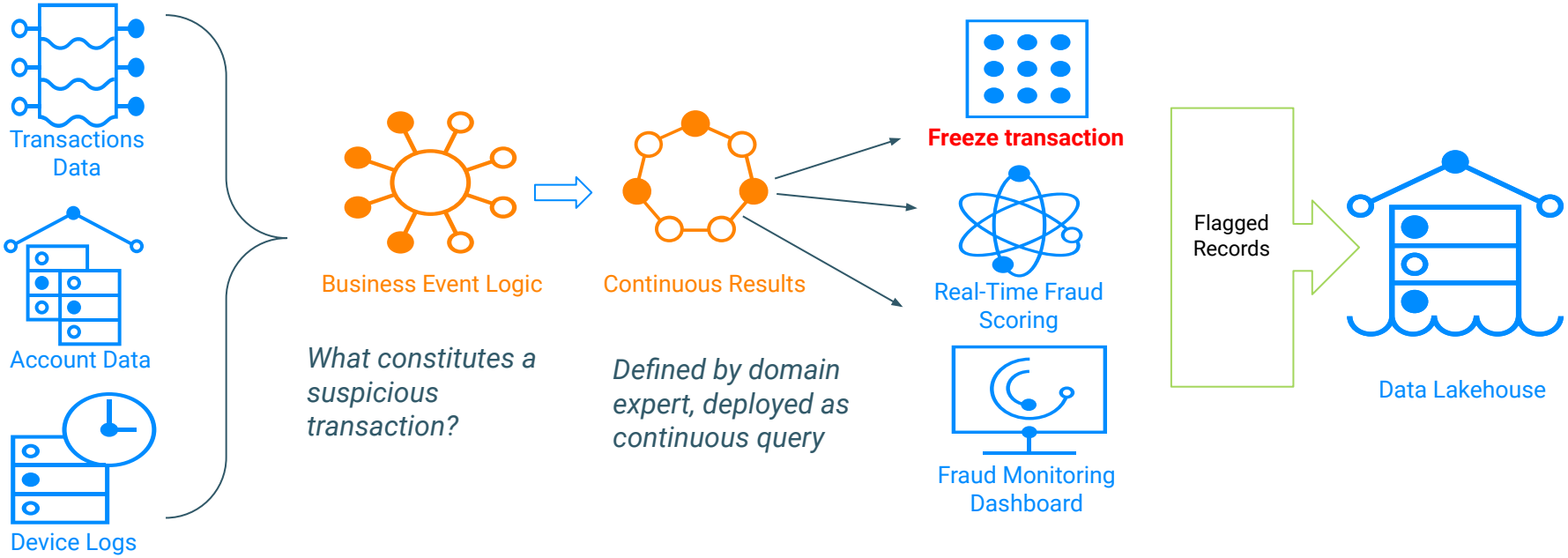
Streamline Ops to deploy, monitor, manage, secure and auto-scale data flows.

Platform/Flow Administrators


Any Data
Any Business Event
Any Data Analyst
Any Data Consumer
Streaming data
Data at rest
Change Data Capture
Real-time Processing
Continuous Results
Event-driven apps
AI Models
Analytics apps
Data Products
Data Lakehouse
Analyze data in motion
Continuous monitoring
Trends and anomalies
No-Code UI
Author once publish anywhere
Analytics lifecycle management for dev/ops

Stop Fraud when it happens

Simplified example of deployed use case



What Challenges Do We Address

SQL Stream Builder Makes it Easy

Simplified data architecture

Unified Processing for reduced complexity

Platform tools built to scale

CDO / CIO / VP of Data Mgmt

- Control over resources and expenses
- Coherent data architecture for maximum insight and agility
- *Accelerate digital transformation*

Devs/Analysts/Data Science

- No-code single interface for all real-time processing
- Click and deploy clusters, analytics and pipelines
- *Domain experts focused on high value analysis*

Platform Administrator

- Integrated security and governance
- Click and deploy clusters with automated reliability and performance tools
- *Scale with confidence*

Responding to Critical Events

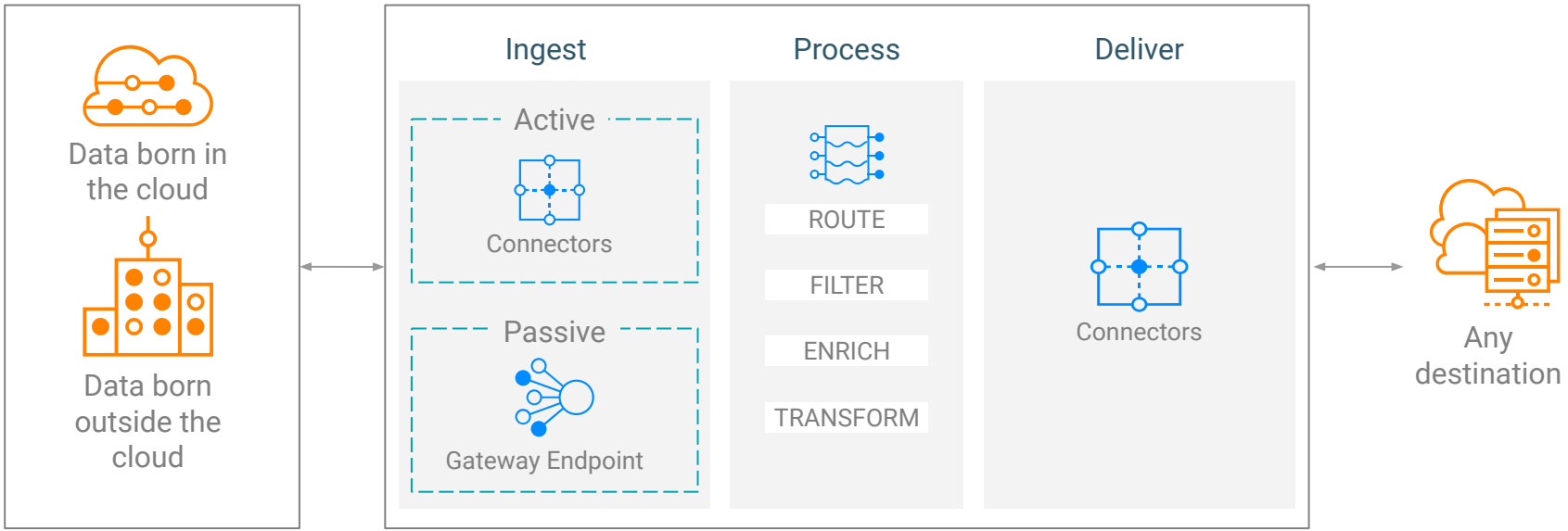
Stream Processing Use Cases Have 4 Criteria

1. **Urgent** need to react/respond
   - Rapidly cooling data and/or limited window of opportunity
   - Eg: fraud, cybersecurity, etc
2. Analytical **insight** is required
   - Raw data does not contain all necessary information
   - Eg: anomaly detection
3. Relevant data is **dispersed**
   - All necessary information is not in one stream
   - Eg: transaction streams + customer tables
4. Data Apps require **agility**
   - Embedding insights into business processes is a non-trivial exercise
   - Ie: Multiple endpoints, environments, formats or consumers

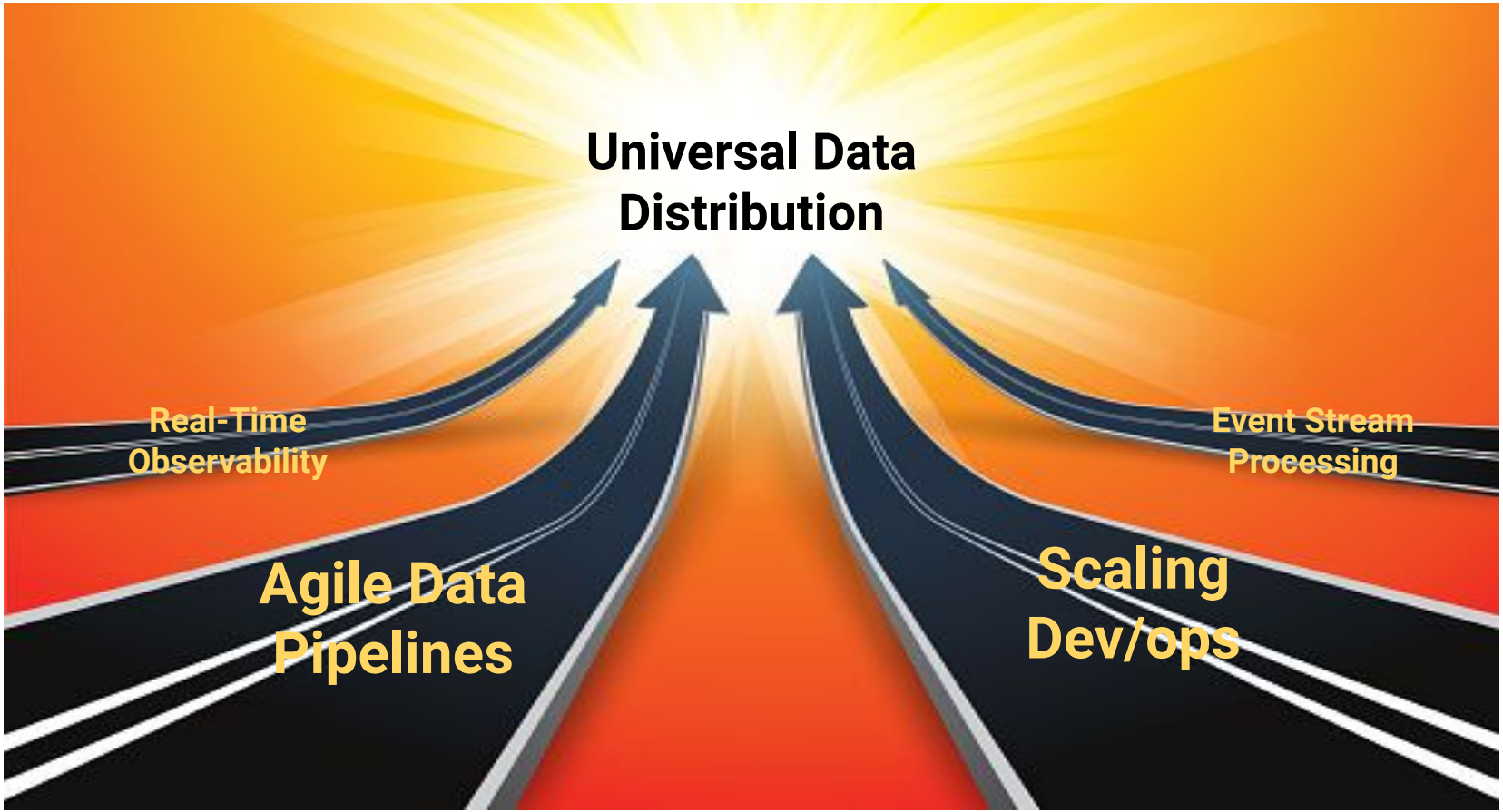
Strike while the iron is hot
-Ancient Proverb

UNIVERSAL DATA DISTRIBUTION - KEY REQUIREMENTS

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

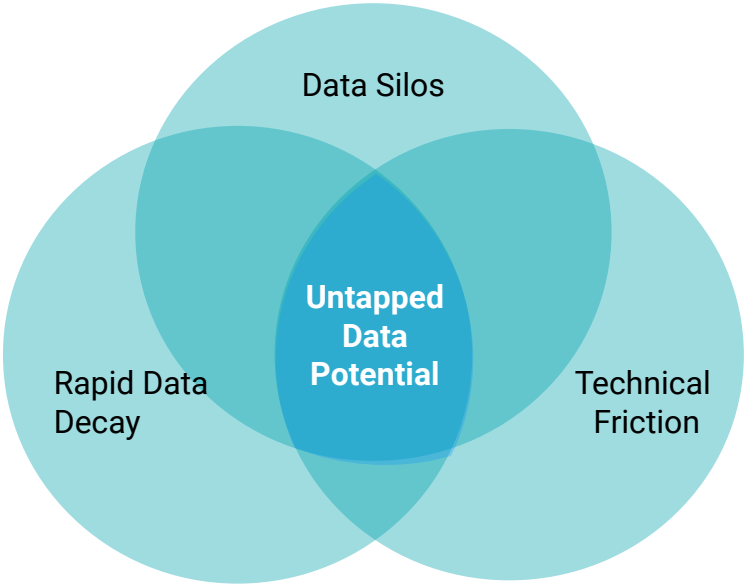


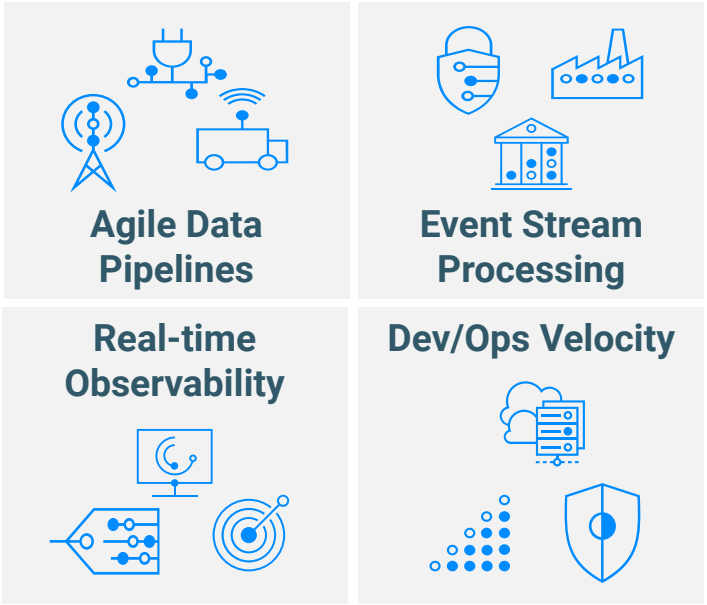
All Roads Lead to UDD



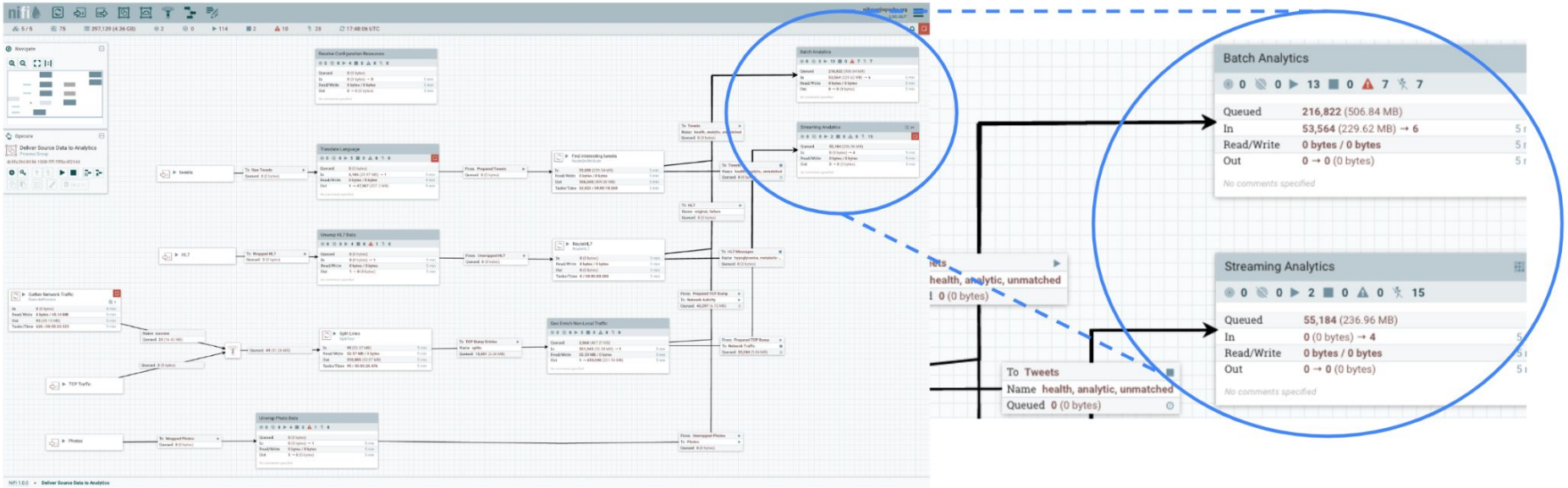
Maximize Impact of High Velocity Data

Solve for untapped data potential across hybrid environments





No Code, Visual Flow Designer for Data Collection & Movement



End-to-end data flow, from many sources to many destinations

DATA-IN-MOTION PLATFORM

CLOUDERA DATAFLOW DATA-IN-MOTION PLATFORM

EDGE & FLOW MANAGEMENT — Everything you need to build scalable data flows from the edge to the public cloud leveraging *Apache NiFi, Apache MiNiFi* and *Cloudera Edge Flow Manager*

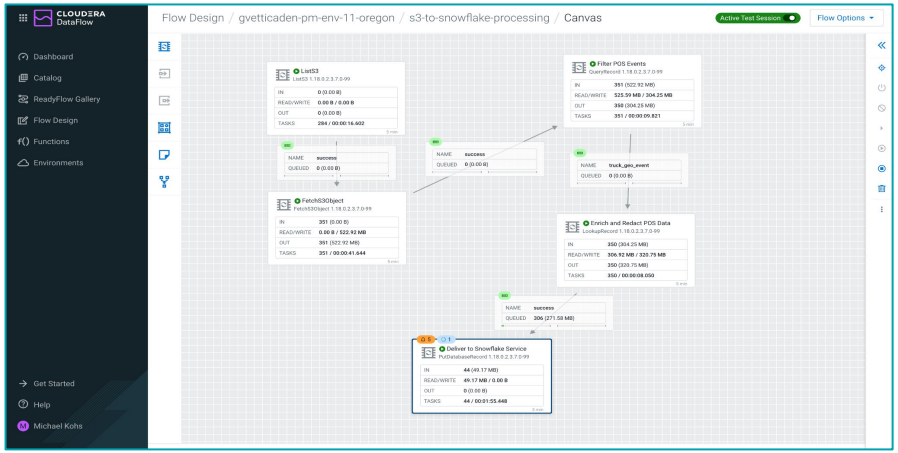
STREAMS MESSAGING — Enterprise grade messaging products for *Apache Kafka*. *Streams Messaging Manager* to monitor/operate clusters, *Streams Replication Manager* for HA/DR deployments, support for *Kafka Connect* and *Cruise Control*

STREAM PROCESSING & ANALYTICS — Powered by *Apache Flink with SQL StreamBuilder, it provides* low-latency stream processing capabilities with advanced windowing & state management

CLOUDERA SDX — Secure, Monitor and Govern your Streaming workloads with the same tooling around *Apache Ranger* & *Apache Atlas*.

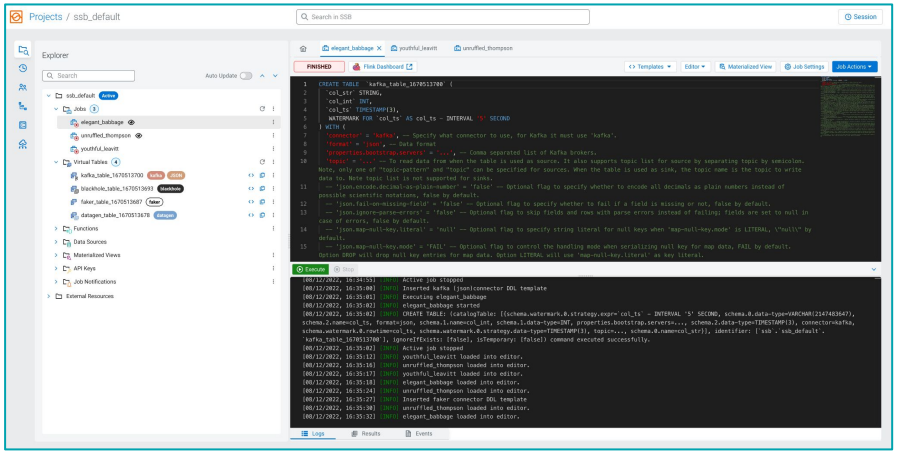
Key Features

DataFlow Designer



ease of administration + intuitive GUI =
Self-service flow development

SQL Stream Builder



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

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?

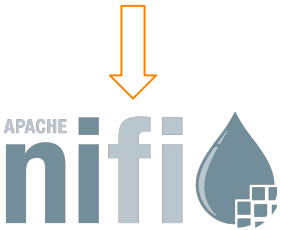
Transform individual records?

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?

USE CASES

UNIVERSAL DATA DISTRIBUTION USE CASES

Data Lakehouse & Warehouse Ingest

Modernize data pipelines with a single tool that works with any data lake, lakehouse or warehouse.

Collect from any source with any type including unstructured data and transform the data into the format that the lakehouse, data lake or warehouse requires.

IoT & Streaming Data Collection

Send data from IoT devices at the edge to a central data flow in the cloud that scales up and down as needed.

Process streaming data at scale, allowing organizations to start their IoT projects small, but with the confidence that their data flows can manage data bursts caused by adding more source devices as well as handle intermittent connectivity issues.

MEDICAL DEVICE MANUFACTURING

700K
Messages Per Second

Processed with Flink for actionable real-time alerts to reduce scrap

CHALLENGE

- Devices must be built to exact specifications
- Reduce scrap and failures by detecting deviations from specifications during manufacturing in real time
- Messages need to be split to perform analytics
- 700,000 Records Per Second with streaming analytics taking place within 2-4 seconds

SOLUTION

- Use SQL Stream Builder to inspect complex nested structures to measure and alert on out-of-spec optic resolutions and color balance.
- Non-Programmer Analysts able to successfully develop and implement algorithm in SQL

OUTCOMES

- Immediately know when devices miss specification
- Improve cost-efficiency via scrap reduction
- Operates in real time with manufacturing producing actionable alerts in seconds

MINING: IOT DATA COLLECTION WITH CLOUDERA DATAFLOW

100K

Tonnes of additional metal processing per year

CHALLENGE

Millions of data points generated by factories and refineries required operational modernization

SOLUTION

Using NiFi and MiNiFi to collect millions of messages per minute from IoT devices and sensor data from the machinery producing the steel

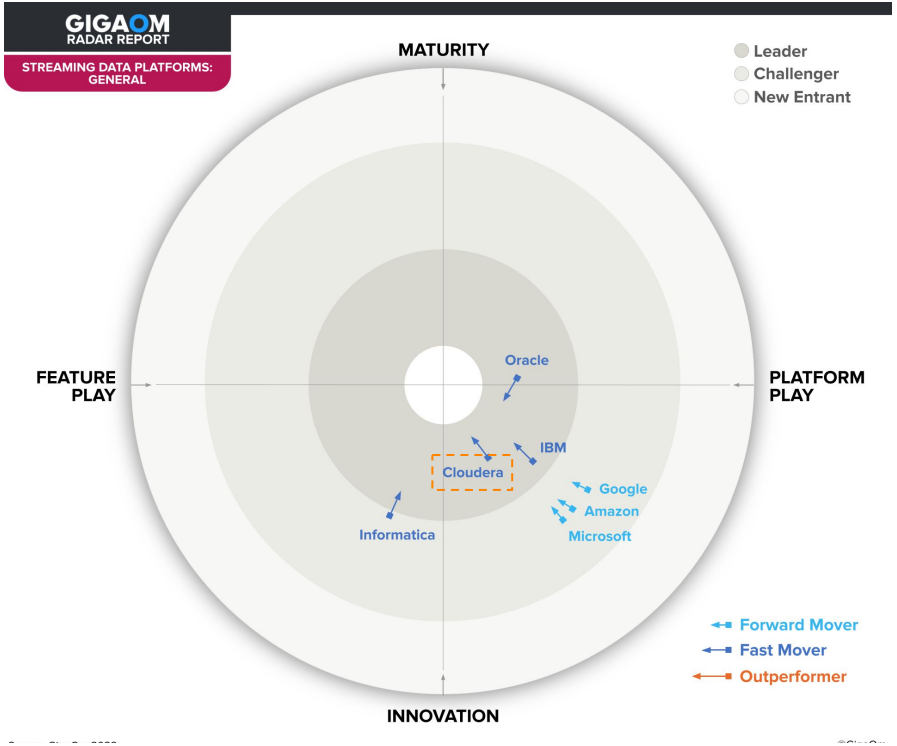
OUTCOMES

AI has automated speed and adjustments of the assembly line - increasing speed by more than 5%

LEADER IN GIGAOM REPORT FOR STREAMING DATA PLATFORMS

Cloudera is a **fast mover** in the GigaOm **innovation / platform play quadrant**.

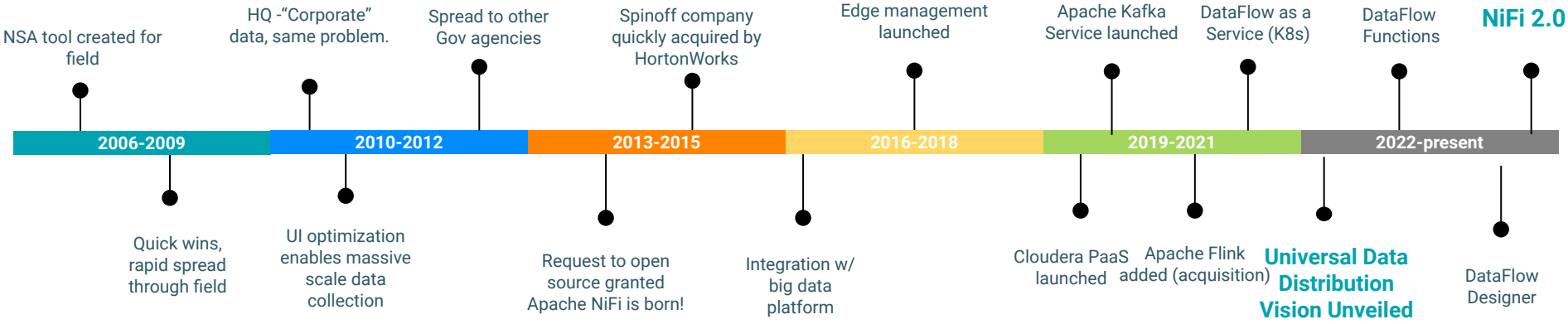
CDF is “an impressively enterprise-ready streaming solution that combines open source Apache Kafka, Flink, Spark, and NiFi technologies … extensive data governance; multiple stream processing engines.”



GigaOm Radar for Streaming Data Platforms

DATAFLOW APACHE NIFI

The Evolution from NiFi to Data-in-Motion



Solving for 1st mile
data collection

Key Requirements:
- Speed & scale
- Ease of configurability
- Resilience

Solving for last mile
data ingestion

Key Requirements:
- Centralized management capabilities
- Flexibility in data types
- Remote delivery

Solving for end to end process
data distribution

Key Requirements:
- Development velocity
- Cloud optimized deployment
- Completeness of capabilities

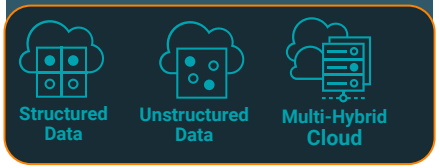
DATAFLOW COMPETITIVE DIFFERENTIATORS

Upleveled key differentiators

Ingest in a Hybrid World: From Any Data Source Anywhere With Any Structure

1

Ingest from **any data source (450+)** born inside and outside of the the cloud with **any structure**



Deliver to any Warehouse or Lake or Mesh on any Cloud

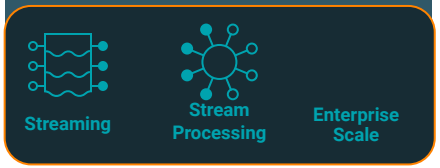
2

Deliver to any destination including data lakes, lake houses, data meshes and cloud services

Streaming Mode and Scale

3

Turn any data source into **streaming mode** and support **streaming scale** supporting hundreds of thousands of data generating clients.



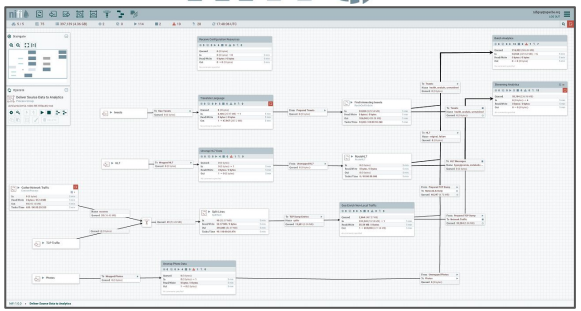
Low Code Extensible Developer Tooling

4

Development tooling that **solves real-world data integration needs** with a flow based **low-code** development paradigm with **extensibility**

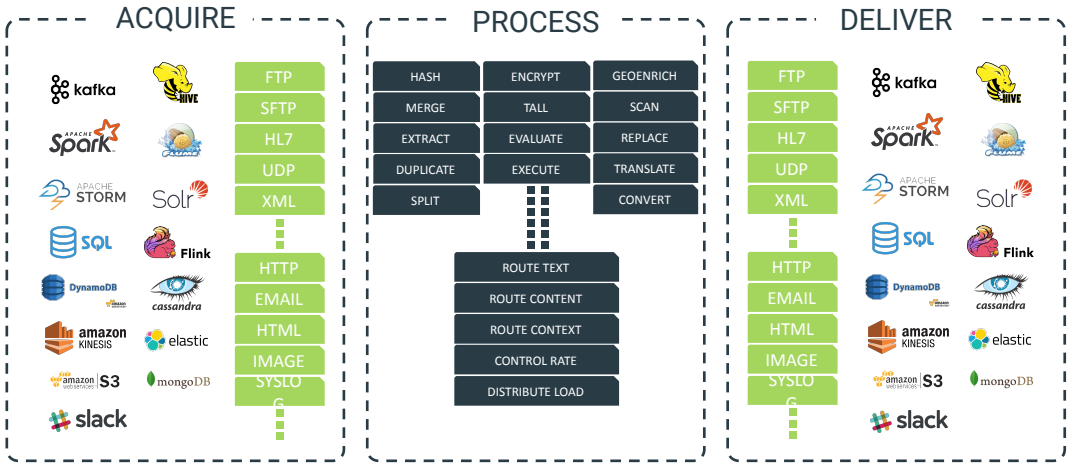
Apache NiFi

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



APACHE NIFI registry

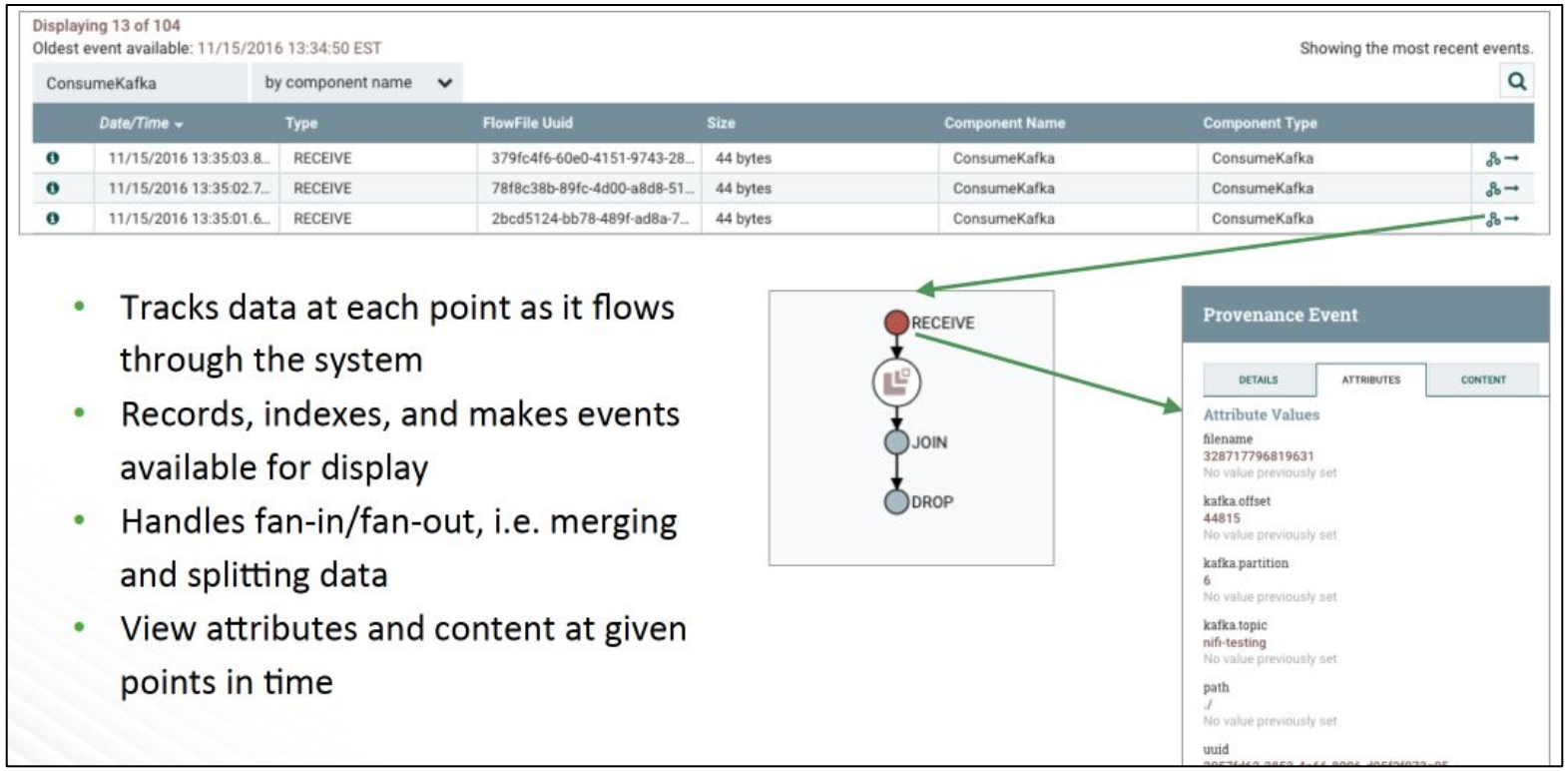
Advanced tooling to industrialize flow development (Flow Development Life Cycle)



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

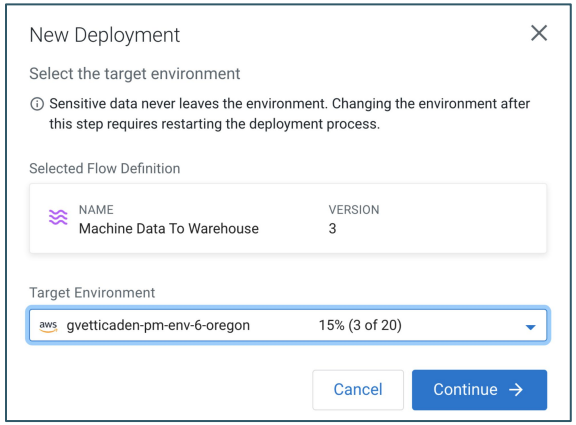
- Guaranteed Delivery
- Full data provenance from acquisition to delivery
- Diverse, Non-Traditional Sources
- Eco-system integration

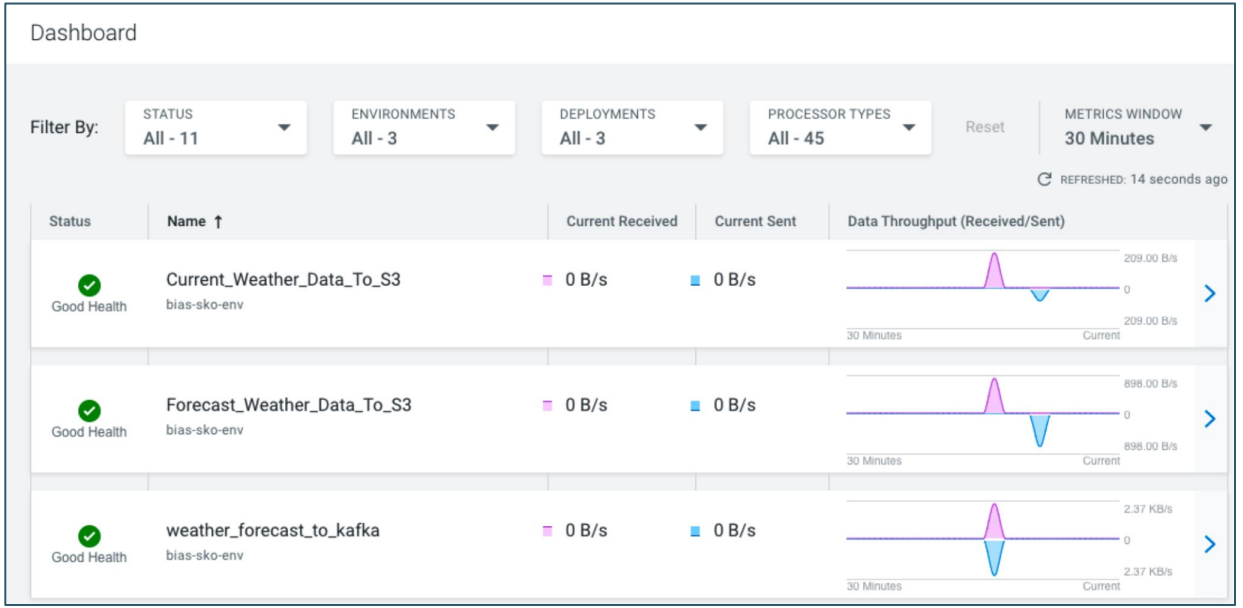
Provenance



Flow Deployment & Monitoring

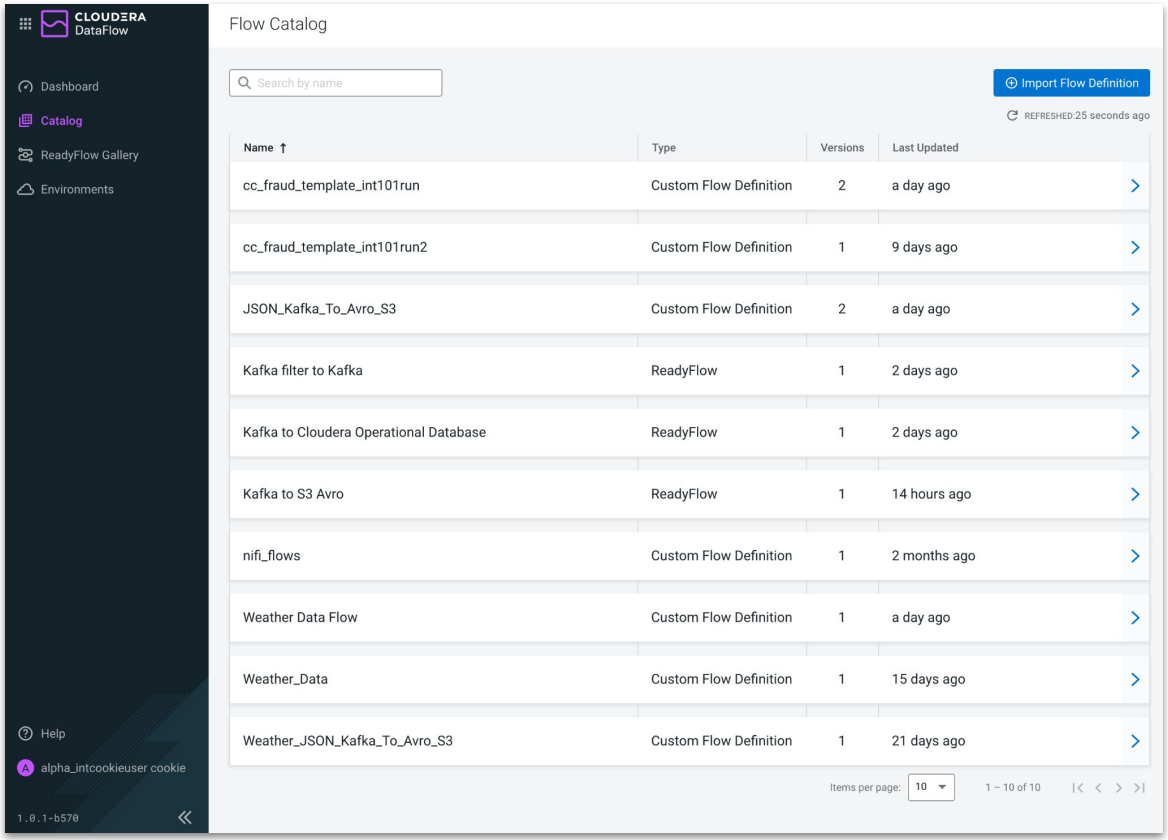
Easily Deploy, Manage, Monitor, and Auto-Scale Flows





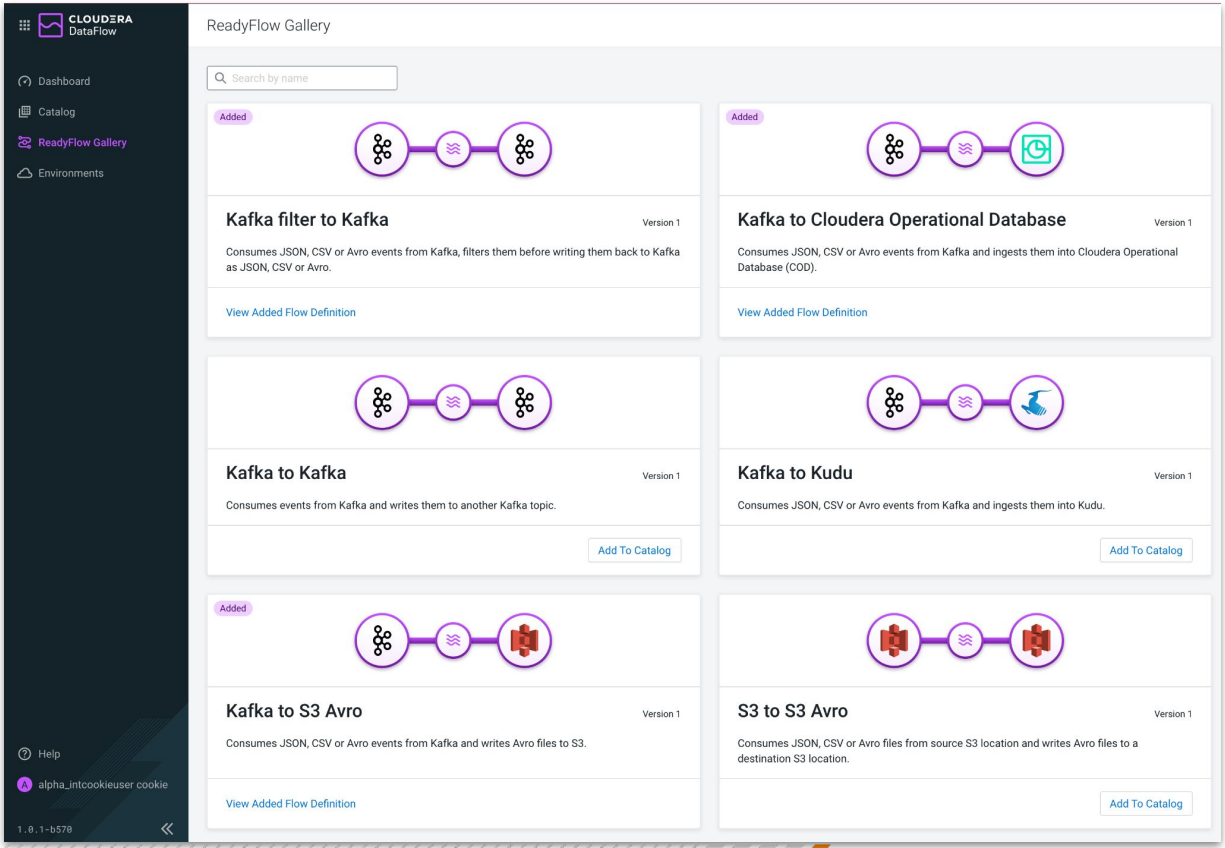
Flow Catalog

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



ReadyFlow Gallery

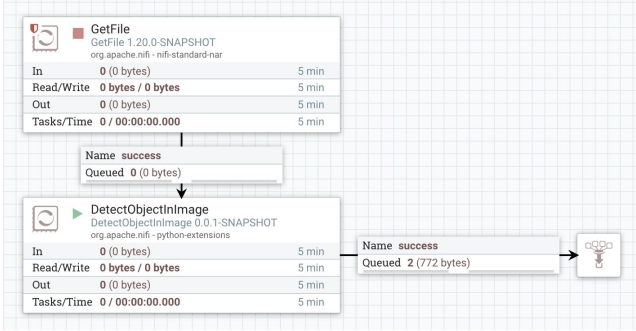
- Cloudera provided flow definitions
- Cover most common data flow use cases
- Optimized to work with CDP sources/destinations
- Can be deployed and adjusted as needed



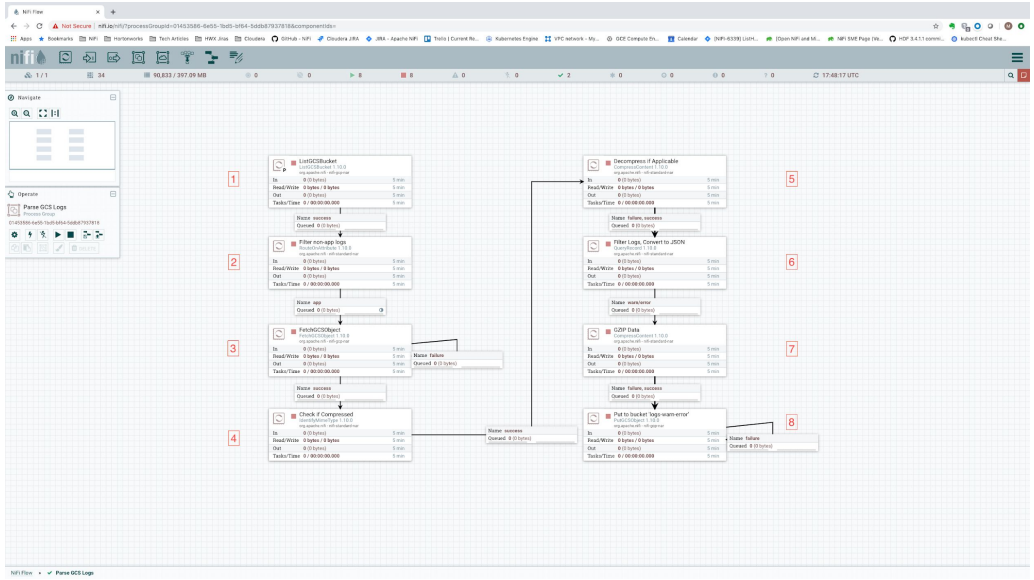
Apache NiFi with Python Custom Processors

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:
```



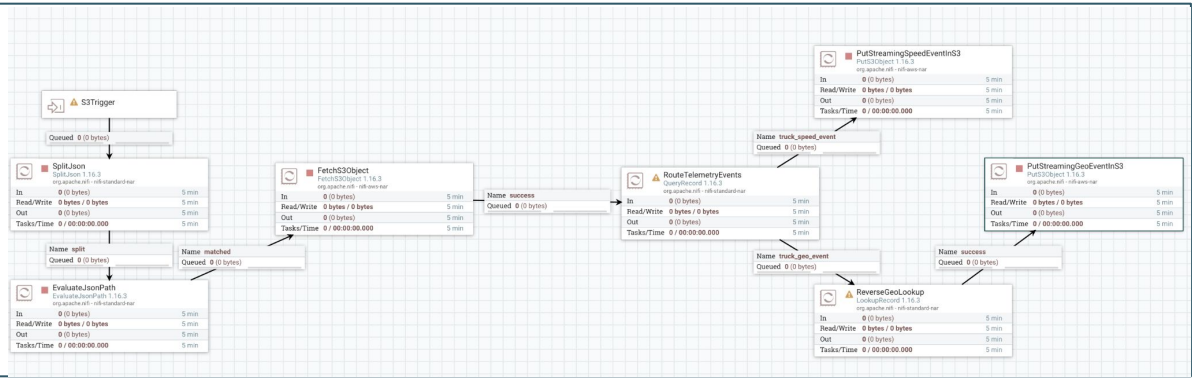
Processing one million events per second with NiFi



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

DEVELOP & DEPLOY SERVERLESS LOW CODE FUNCTIONS

Step1. Develop functions on local workstation or in CDP Public Cloud using **no-code**, UI designer



Step 2. Run functions on serverless compute services in AWS, Azure & GCP

AWS Lambda

Azure Functions

Google Cloud Functions

Highlights - Cloudera DataFlow for the Public Cloud (CDF-PC)

Ingest & move data in a hybrid world with an ecosystem of 450+ connectors that connect to and distribute data from the edge and on-premises data centers, to SaaS solutions, and public clouds.

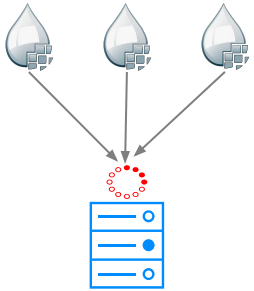
Two NiFi runtimes optimized [1] for real-time, always running workloads with **DataFlow deployments** and [2] for event-driven, short-lived batch jobs with **DataFlow Functions** for any public cloud.

Low-code developer tooling to simplify and accelerate onboarding and managing of new data sources with a low-code, extensible visual flow designer.

Enterprise-grade security & governance – Deploy your data flows with confidence and trust with Cloudera's offering unified security and governance across the entire platform.

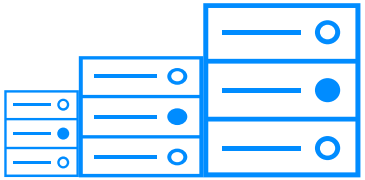
Integrated with CDP – Out of the box connectors and proven patterns for integration with other CDP services such as Cloudera Data Warehouse, Cloudera Operational Database or Cloudera Machine Learning.

OPERATIONAL CHALLENGES WITH NiFi FLOWS



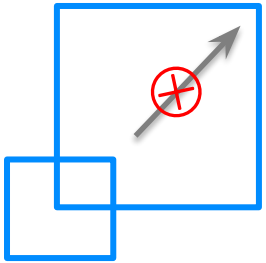
Resource contention

Impacts performance of all flows in the cluster



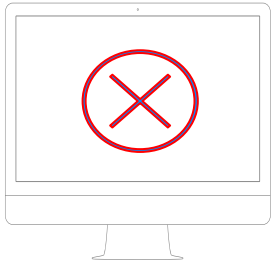
Oversizing clusters

High infrastructure costs



On-demand manual scaling

Operational nightmare



Comprehensive flow visibility

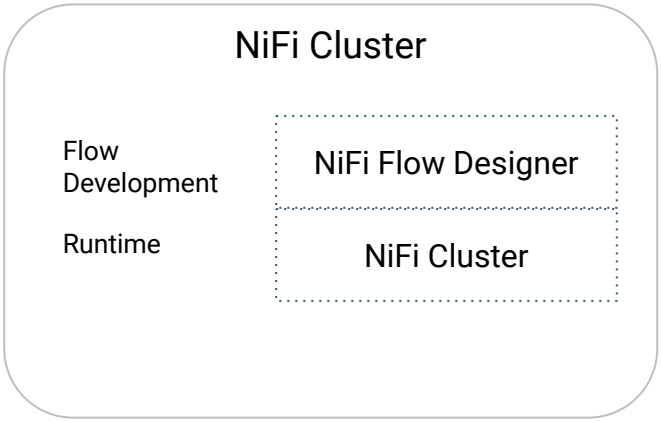
Monitoring NiFi flows across multiple clusters can be challenging

ARCHITECTURAL SHIFT IN DATAFLOW

Decouple NiFi Flow Designer and the NiFi Cluster Runtime to Support Diverse Runtimes

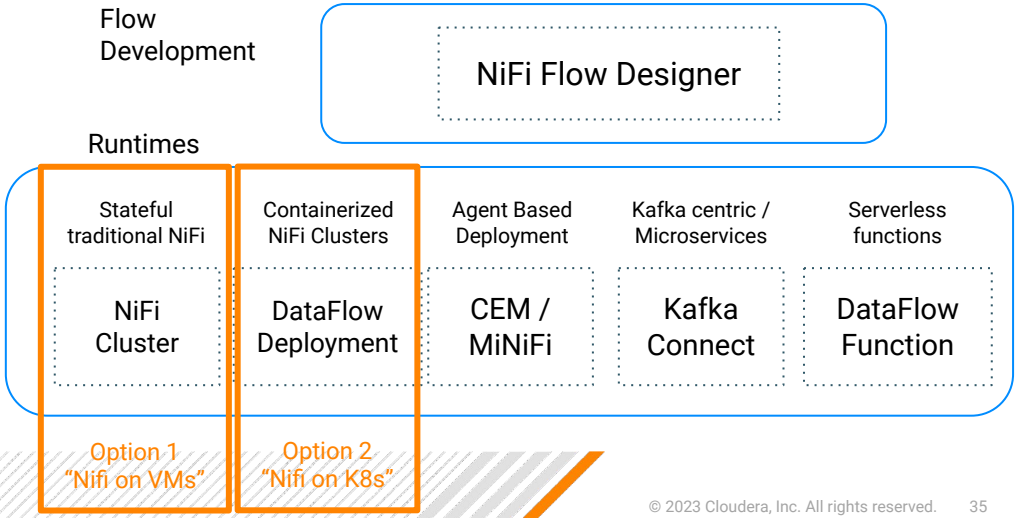
Classic NiFi Architecture

NiFi Flow Designer + NiFi Cluster Runtime are tightly coupled



New NiFi Architecture

Develop flows in designer and deploy in different runtimes based on use case



CONTAINER BASED DATAFLOW (AS OF TODAY PUBLIC CLOUD ONLY)

Flow Catalog

Keep track of your flow definitions and versions in a central catalog

Reuse your existing NiFi flows by uploading them to the catalog

Discover, search and reuse existing flows easily

Flow Deployment

Allows easy flow deployment based on NiFi 1.18 across CDP environments (Dev, QA, Prod)

Define and assign KPIs to your flows

Easy NiFi version upgrades

Update/Add KPIs, Update Parameters, Change sizing configuration

Automatic infrastructure scaling based on CPU utilization

Flow Monitoring

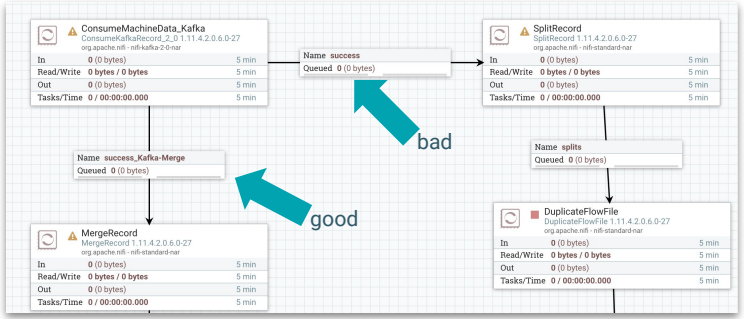
Central monitoring console for all your flows across environments

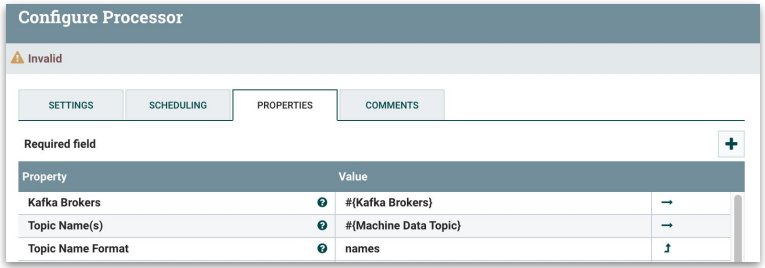
Monitor flow metrics and infrastructure usage

Define alerts for flows breaching assigned KPIs

FLOW DEVELOPMENT BEST PRACTICES

- Name your processors/ connections
- Parameterize connection information
- Tag sensitive properties as “sensitive”
- Define controller services on process group level (except *Default NiFi SSL Context Service*)





FLOW DEVELOPMENT BEST PRACTICES

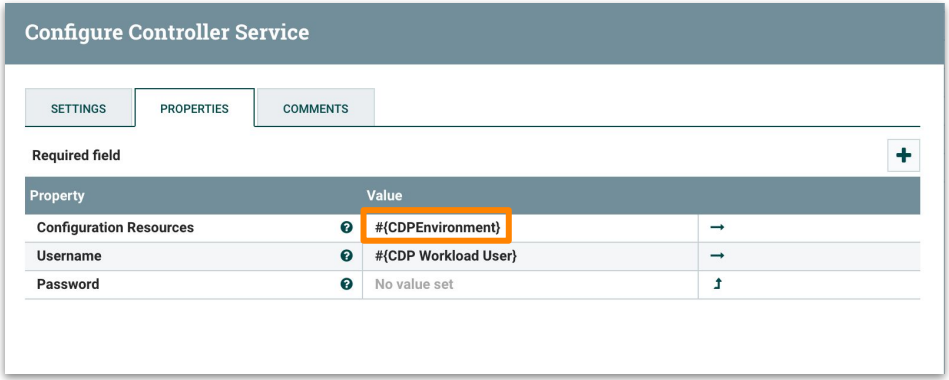
CDPEnvironment Parameter & Default SSLContextService

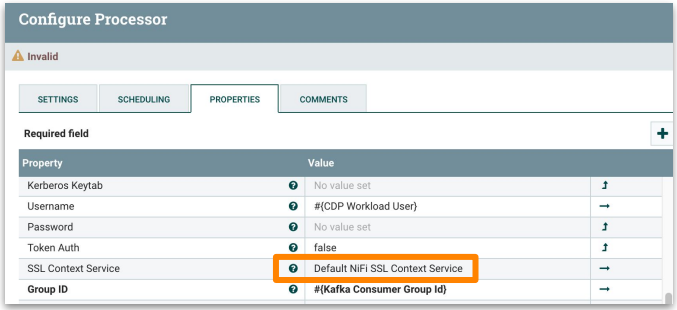
CDP Environment Parameter

- Use whenever Hadoop configuration files are needed
- CDF detects parameter usage, obtains the Hadoop configuration files from SDX, makes them available to NiFi pods and replaces parameter value accordingly
- No copying of config files required anymore

Default NiFi SSL Context Service

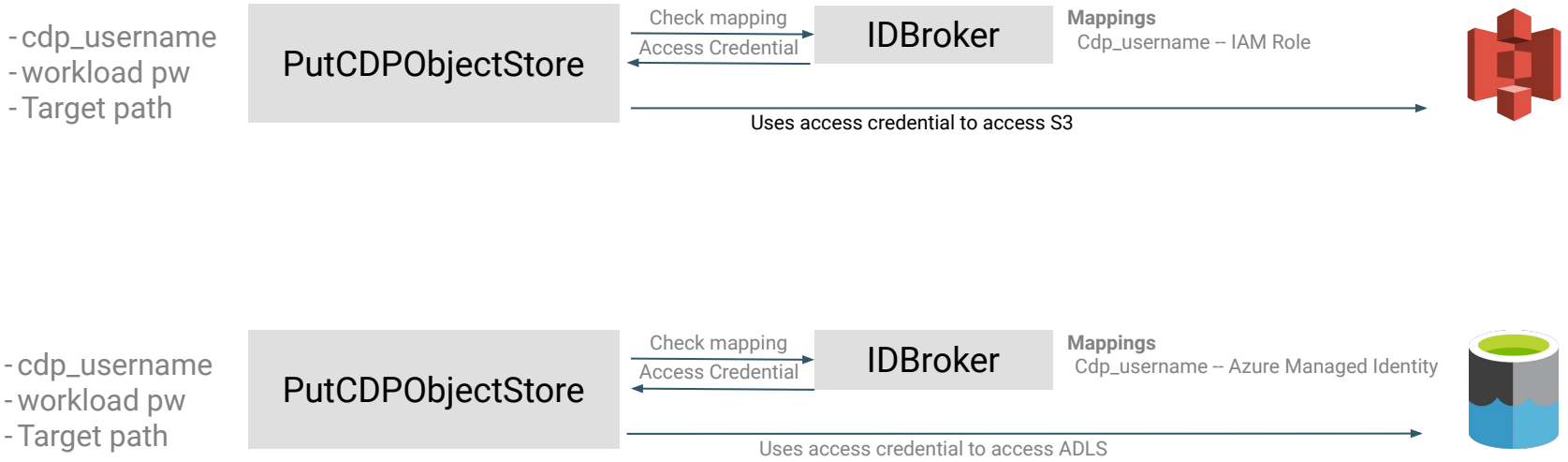
- Use whenever SSL Context Service is required to interact with CDP service in target environment
- CDF detects reference, creates a key and truststore for the target environment and configures a *Default NiFi SSL Context Service* accordingly
- No more manual creation of truststores and moving around of certificates to interact with CDP services
- *Default NiFi SSL Context Service* must be an *external controller service* i.e. defined **outside** of the process group that's exported





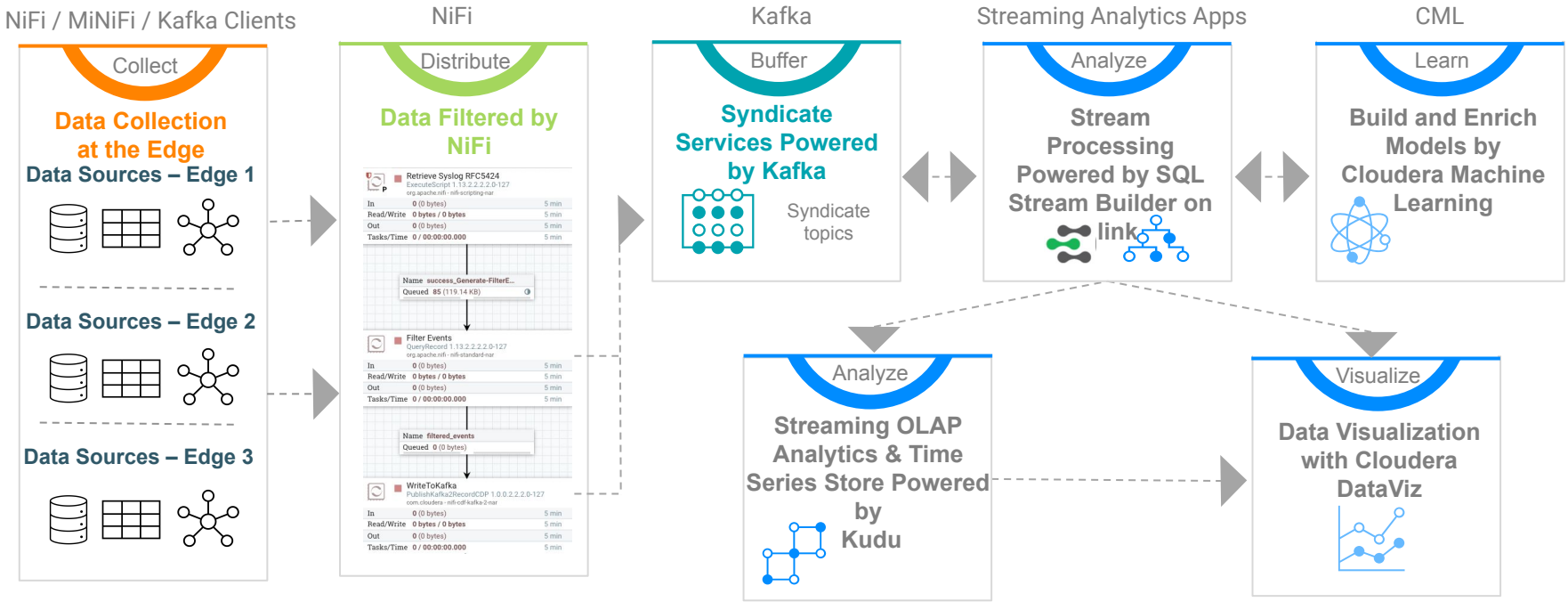
FLOW MANAGEMENT CLUSTERS - CDP DATA ACCESS

Object Store Access Governed by SDX



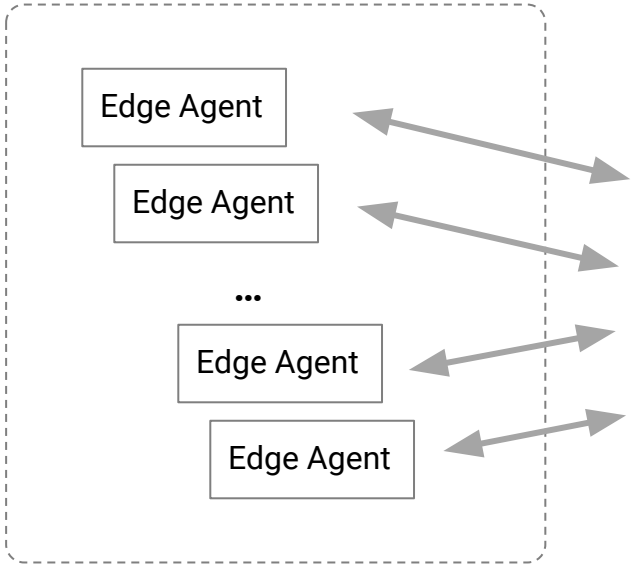
https://docs.cloudera.com/management-console/cloud/security-overview/topics/security_how_identity_federation_works_in_cdp.html

LOG ANALYTICS REFERENCE ARCHITECTURE

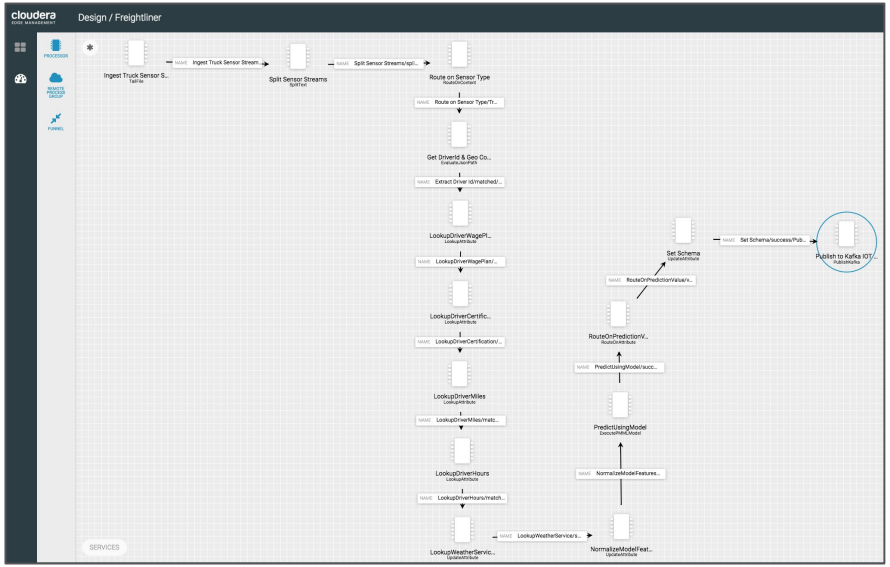


EDGE FLOW MANAGEMENT

Data Collection & Management at the Edge



Lightweight edge agents focused on data collection and processing at the edge

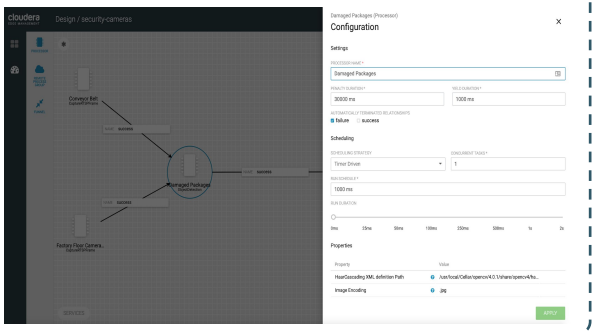


Central command & control to design and deploy to thousands of agents

Cloudera Edge Management

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

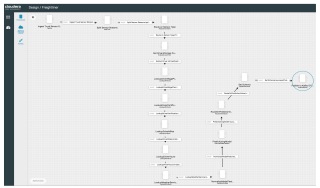
Edge Flow Manager



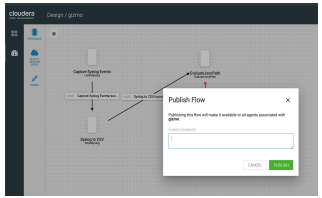
APACHE NIFI minifi

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

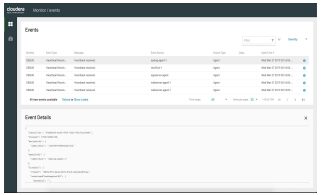
Flow Authorship



Flow Deployment



Flow Monitoring



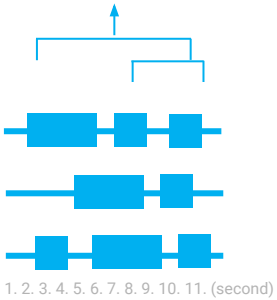
- Small footprint agent with MiNiFi
- Java and C++ agents
- Rich edge processors (edge collection & processing)
- End to end lineage and security
- Kubernetes support

- 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

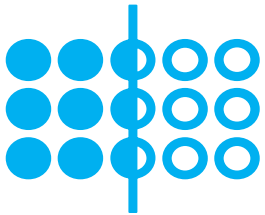
STREAM PROCESSING & ANALYTICS

STREAMING ANALYTICS ACCESSIBILITY ACROSS THE ENTERPRISE

Event Processing



\+



\+

\=

Streaming Analytics

Capture Events that Matter

Low-latency analytics use cases

Filtering and Joining Data

Both batch and streaming data

Data Analysts Can Write SQL Queries

Across the Line of Businesses

SQL STREAM BUILDER (SSB)

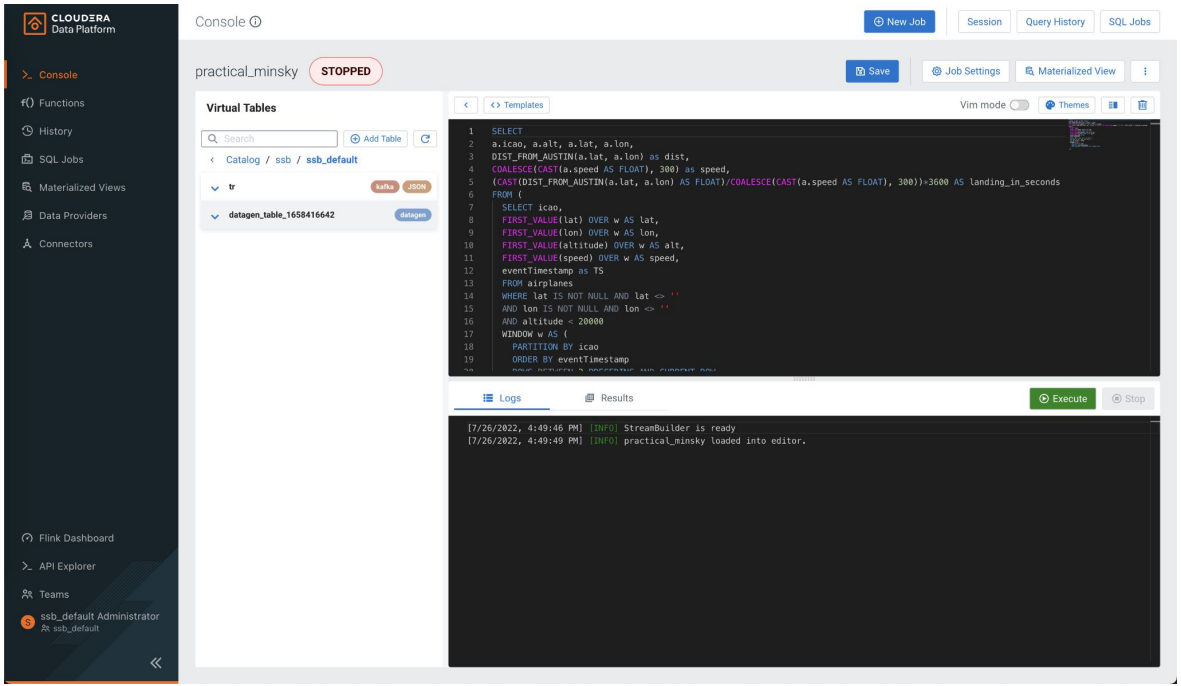
Democratize access to real-time data with just SQL

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

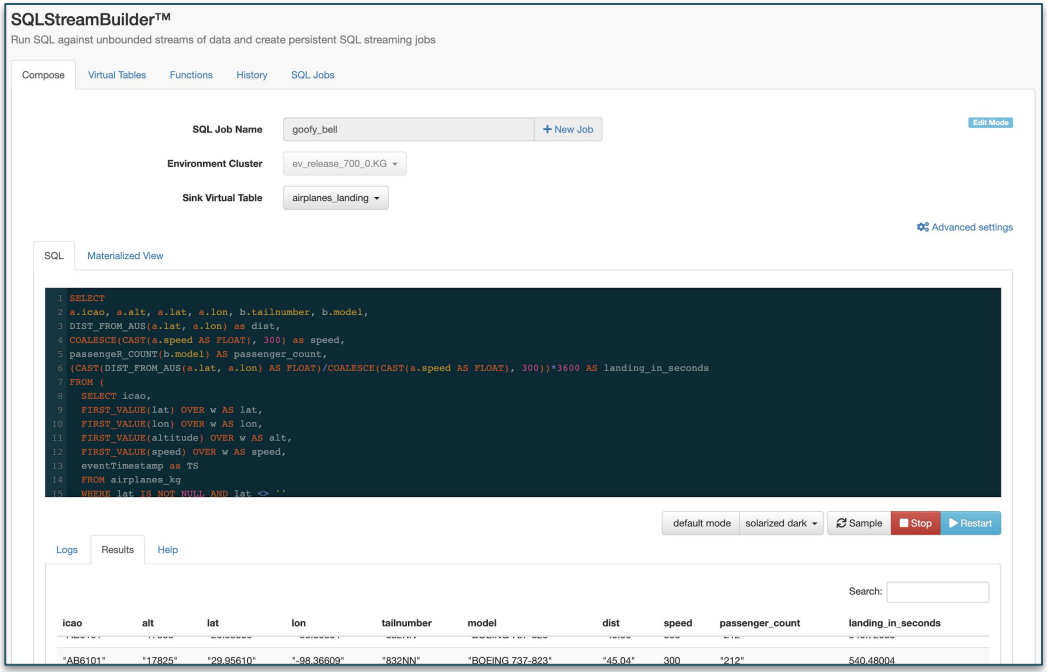


STREAMING SQL BUILDER ON FLINK

Fast deployment of streaming SQL queries on CDP Public & Private Cloud

Key Highlights

- SQL Stream Builder for fast curated event lists
- Automatic schemas for streaming events
- Materialized Views
- Instant APIs for streaming results
- Extensible with user defined functions



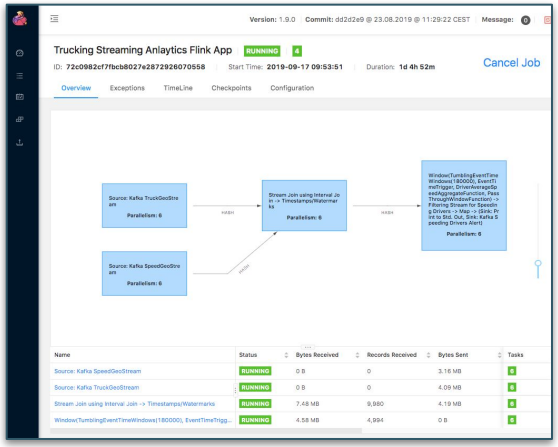
APACHE FLINK

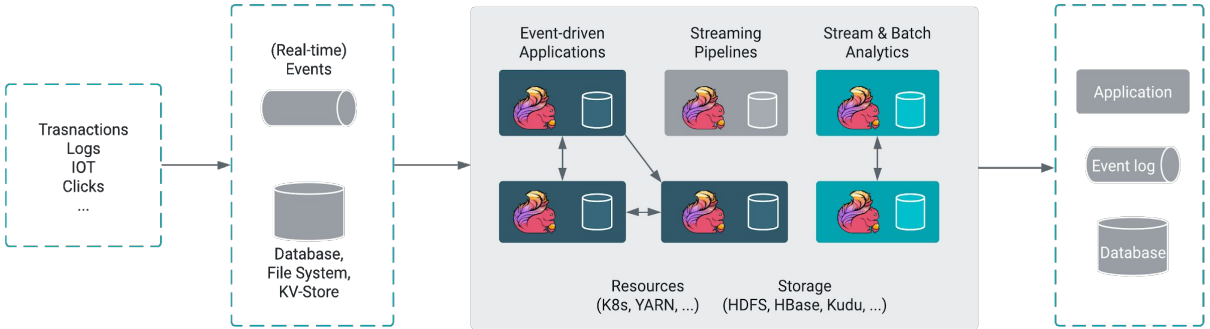
- Distributed processing engine for stateful computations
- 10s TBs of managed state
- Flexible and expressive APIs
- Guaranteed correctness & Exactly-once state consistency
- Event-time semantics
- Flexible deployment & large ecosystem (K8s, YARN, S3, HDFS..)
- Support for Flink SQL API

Low Latency

Event Processing

Real-Time Insights

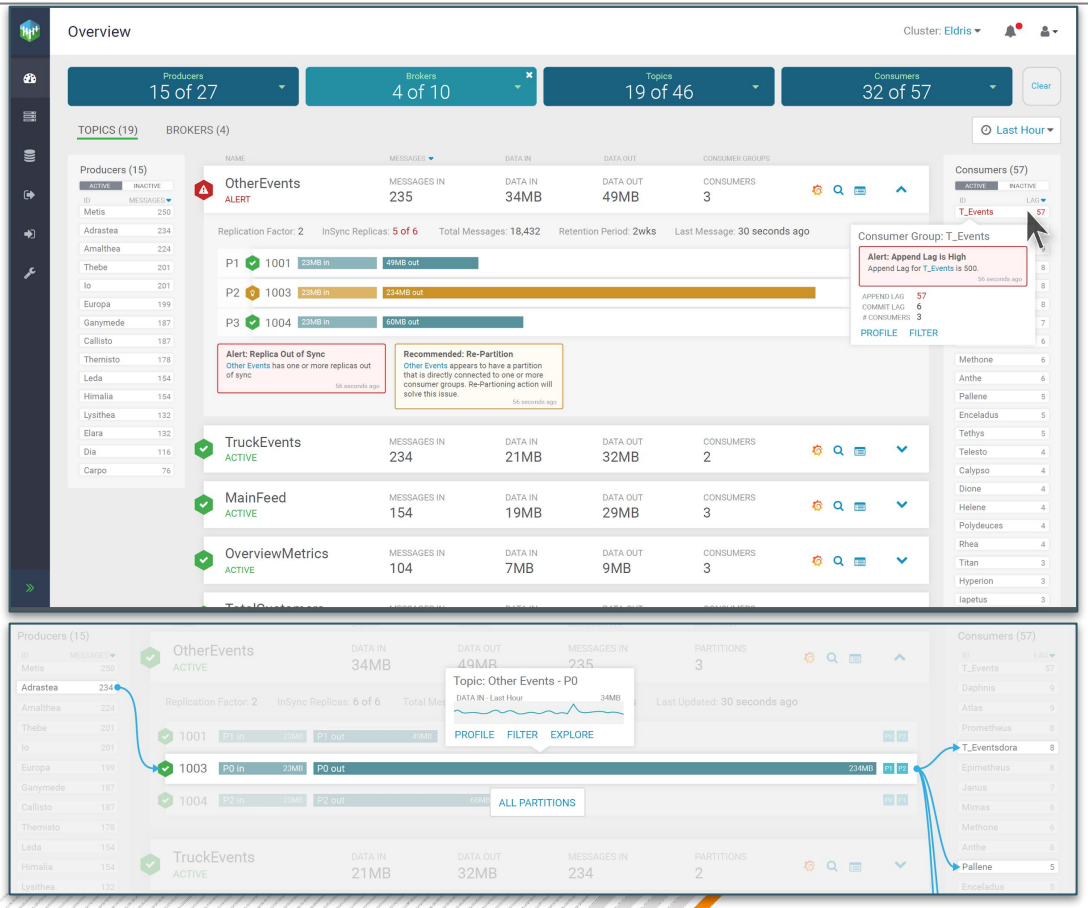




STREAMS MESSAGING

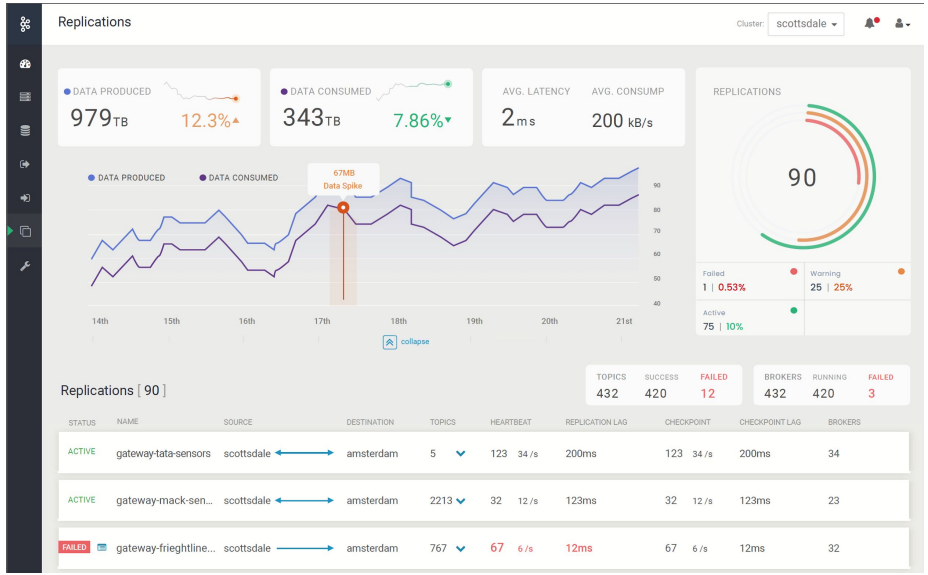
STREAMS MESSAGING MANAGER (SMM)

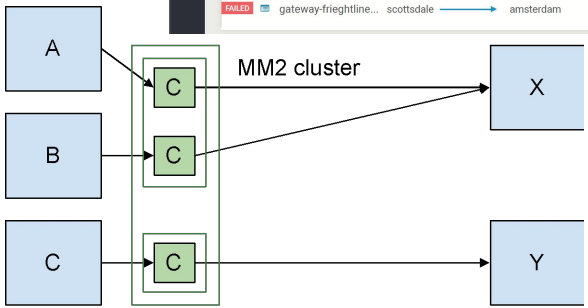
- **Single Monitoring Dashboard** for all your Kafka Clusters across 4 entities
- Data Explorer
- Topic **administration**: CRUD
- **Alerts** on available metrics
- **REST** as a First Class Citizen
- Designed for the Enterprise
  - Support for **Secure** Kafka cluster
  - Rich **Access Control Policies**



Streams Replication Manager (SRM)

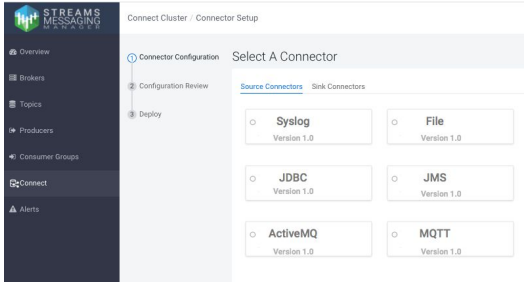
- Event Replication engine for Kafka
- Supports active-active, multi-cluster, cross DC replication scenarios
- Leverage Kafka Connect for scalability and HA
- Replicate data and configurations (ACL, partitioning, new topics, etc)
- Offset translation for simplified failover
- Integrate replication monitoring with SMM

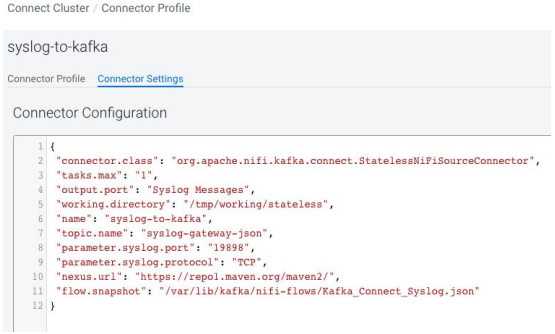




KConnect Connectors

- New Connectors
- CDC Debezium Connectors
- SMM UI Integration
- Reuse your Kafka Infrastructure
- Enterprise Security for Secrets, Authentication and Authorization





Sources

- ActiveMQ (via JMS)
- MQTT
- Syslog over TCP
- JDBC
- JMS
- HTTP

Sinks

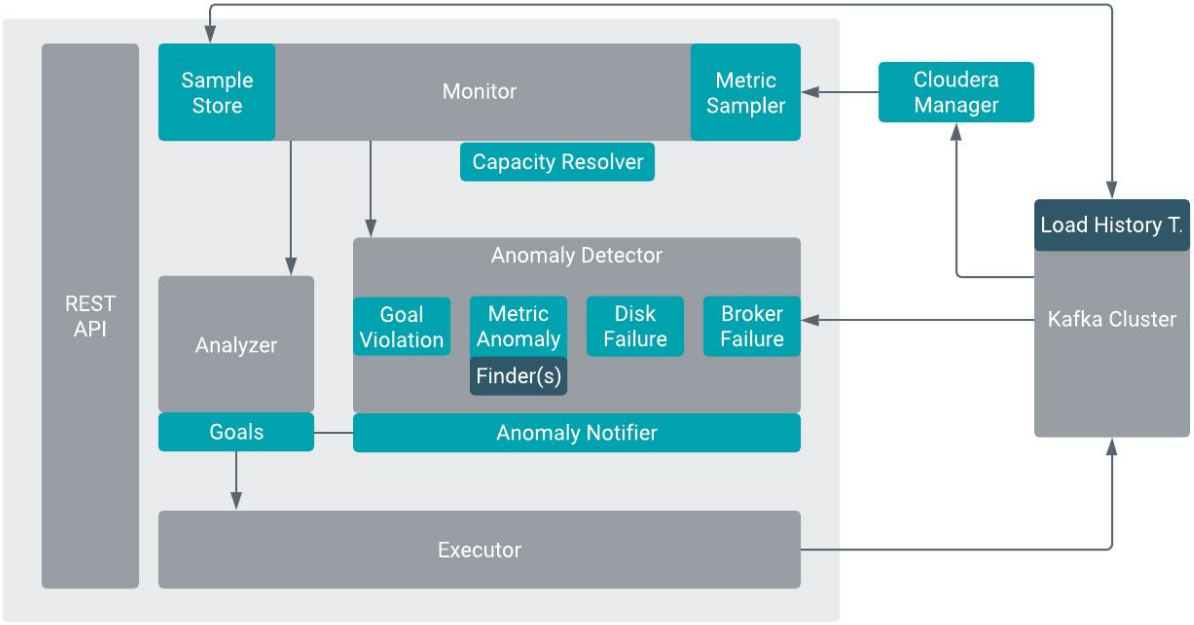
- AWS S3
- ADLS
- Kudu
- HDFS
- HTTP

CDC Connectors

- MySQL
- PostGreSQL
- Oracle
- SQLServer

Cruise Control

- Self- Healing from broker, hard-drive and other failures
- Goal based rebalancing based on replica and broker load and other metrics
- Simplify broker onboarding & decommissioning; no more JSON replica assignments



DEMOS

Future of Data - Princeton + Virtual

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

FUTURE OF DATA

AN OPEN SOURCE COMMUNITY

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

CLOUDERA