



Introduction to Cloudera Data Flow (CDF)

Data In Motion Field Engineering

Agenda



Introduction

Flow management

Edge Management

Stream processing

Enterprise Services

Cloud Services

Conclusions

Introduction

Low Latency Streaming Analytics Use cases



Telco Network Monitoring



Content Recommendations



Search Optimization



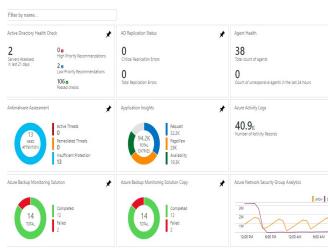
Clickstream Analysis



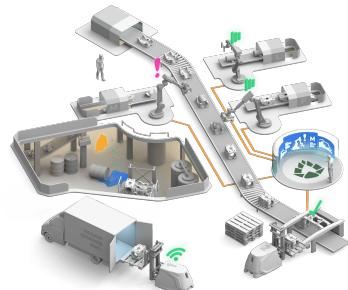
Fraud Detection



Gaming Analytics



Application Monitoring



Industrial IoT

Business Value Realized with Cloudera Data Flow

1

Faster insights

By ingesting and processing events in real-time. Whether it's part of an application, or a standalone pipeline, insights are available at low latency

2

Better customer experience

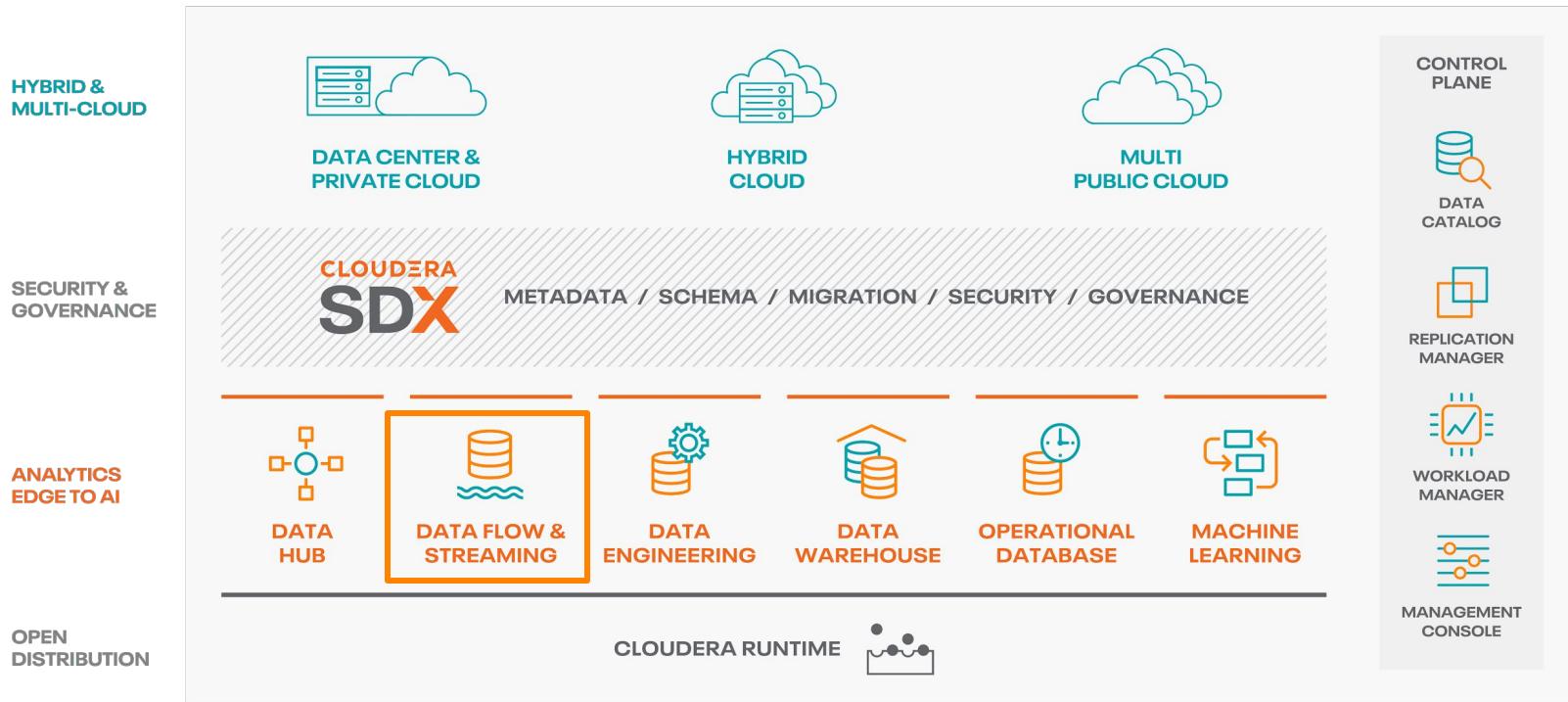
Reacting and answering the customer in real-time is a game changer in a world where we are used to smartphones, continuous connectivity and social networks.

3

Increased agility at lower cost

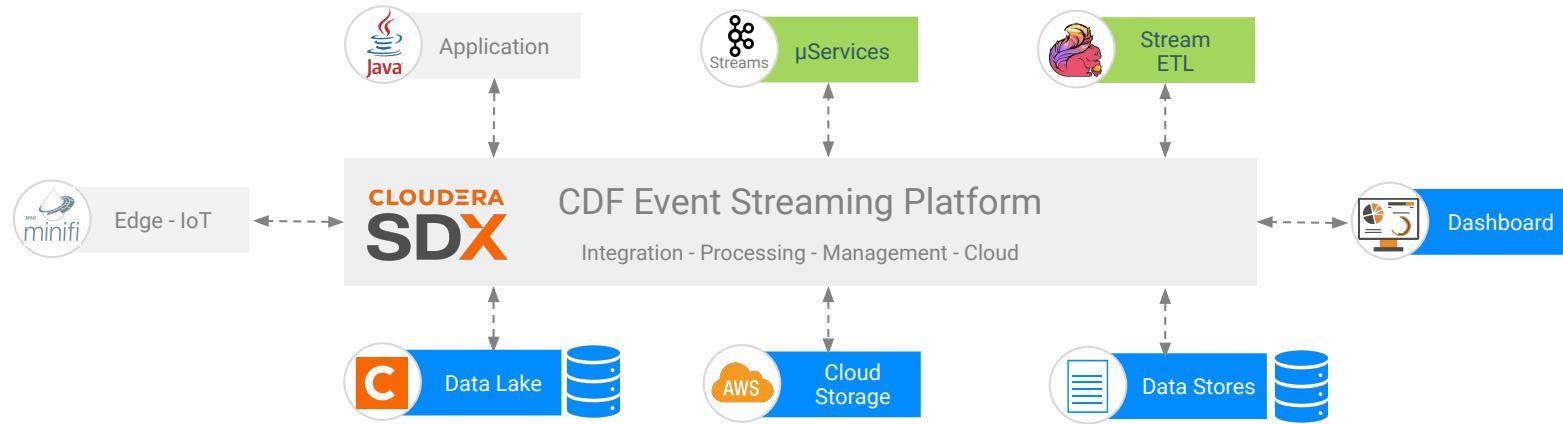
By putting events at the center of the data architecture, it becomes a living experimentation lab. new applications can be developed easily at lower cost

Cloudera Data Platform



Event Driven Organization with CDF

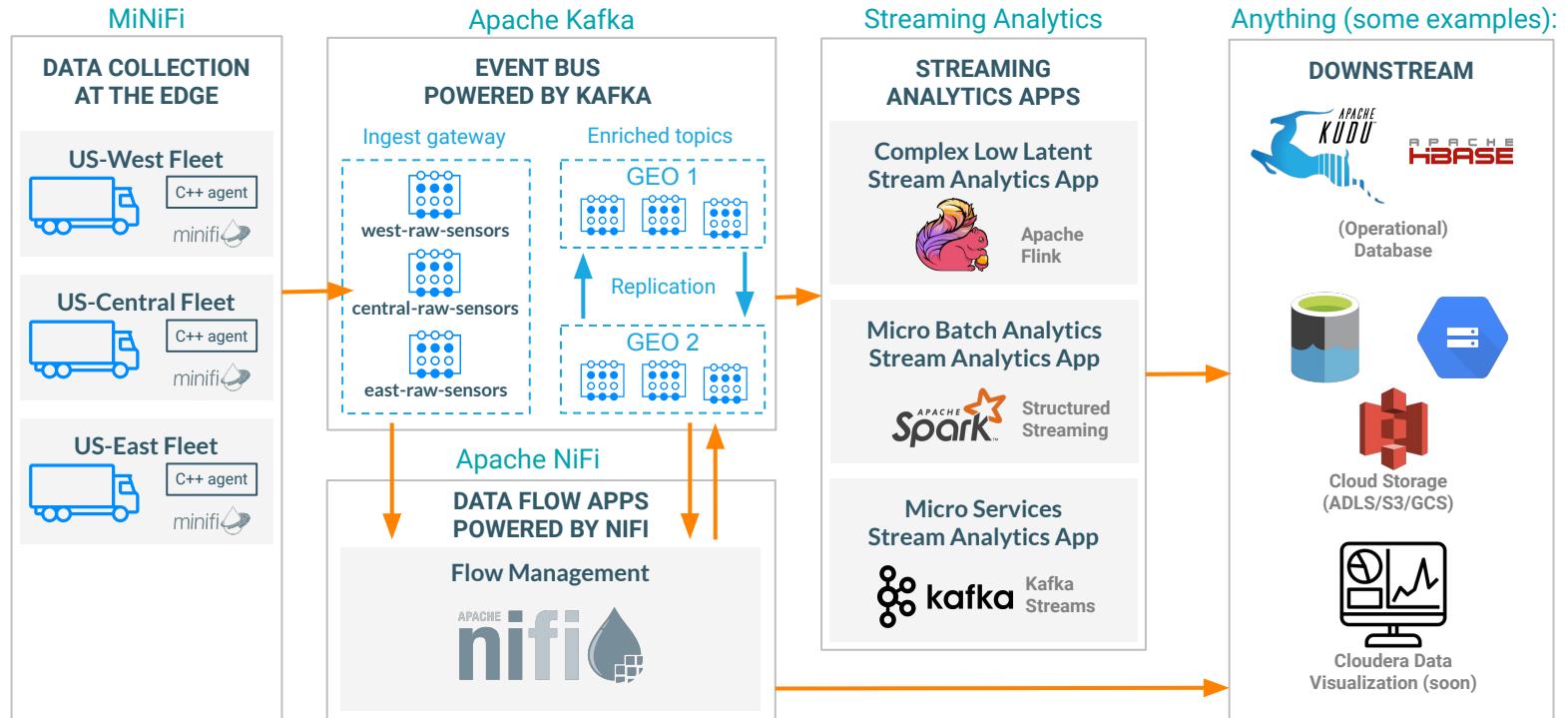
Modernize your data and applications



Cloudera DataFlow Data-in-motion Platform

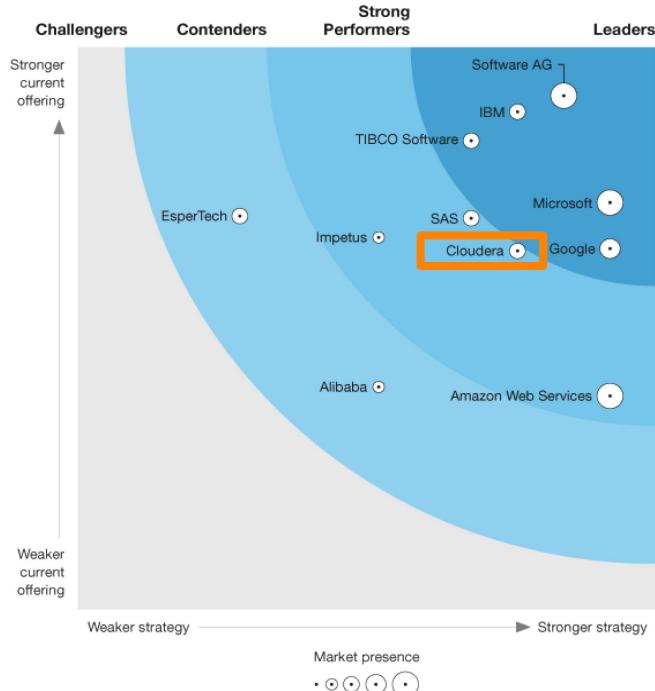


Cloudera DataFlow reference architecture



Strong Performer in Forrester Wave for Streaming Analytics

CDF is a Strong Performer in the Forrester Wave for Streaming Analytics, Q3 2019



ANALYST REPORT
Cloudera named a Strong Performer in The Forrester Wave™: Streaming Analytics, Q3 2019

Report recognizes Cloudera DataFlow as "more than streaming analytics"
Cloudera has been named as a Strong Performer in The Forrester Wave™: Streaming Analytics, Q3 2019. We're excited to make our debut in this Wave at, what we consider to be, such a strong position. We are proud to have been named as one of "The 11 providers that matter most" in streaming analytics. The report states that analytics prowess, scalability, and deployment freedom are key differentiators across 26-criterions.

Download the report
First Name _____ Last Name _____
Job Title _____ Business Email _____
Company _____ Phone _____
 I would like to be contacted by Cloudera for newsletters, promotions, events and marketing activities. Please read our [privacy and data policies](#).
 Yes, I consent to my information being shared with Cloudera's solution partners to offer related products and services. Please read our [privacy and data policies](#).
 I agree to Cloudera's [terms and conditions](#).
[Reset](#) [Sign me up!](#) [Autofill my information](#)

THE FORRESTER WAVE™
Streaming Analytics
Q3 2019



September 23, 2019

Cloudera named a Strong Performer in The Forrester Wave™: Streaming Analytics, Q3 2019

By Dinesh Chandrasekhar [@appint4all](#)



Cloudera has been named as a Strong Performer in the Forrester Wave for Streaming Analytics, Q3 2019. We are totally excited to make our debut in this Wave at, what we consider to be, such a strong position. We are proud to have been named as one of "The 11 providers that matter most" in streaming analytics. The report states that analytics prowess, scalability, and deployment freedom are key differentiators in the evaluation across 26-criterions.

Recommended for You



Flow Management

First Mile Problem

Lack of Agile Data Collection Tools Has Hindered the Start of the Data Lifecycle Journey



Data Source Diversity and Scale

Platform/Data Ingestion Team

"BU has asked for tooling to ingest data from 80 different sources and observe 1 million events per hour. Our current tools don't support these diverse data sources or extreme scale requirements."



Agility

Data/App Developer

"Onboarding new data sources and updating ingestion services is extremely painful and slow. Too much code to write and maintain to provide enterprise ingestion services (failures, scheduling, re-tries)"



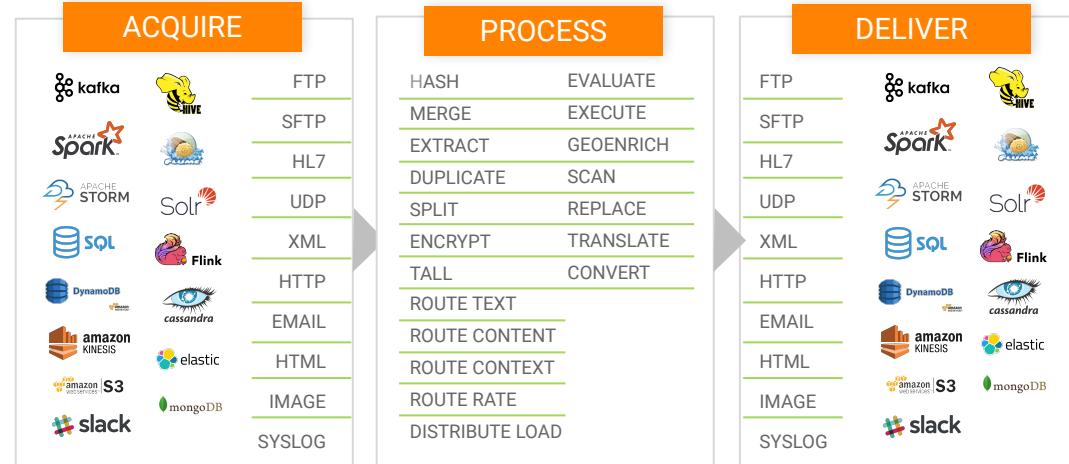
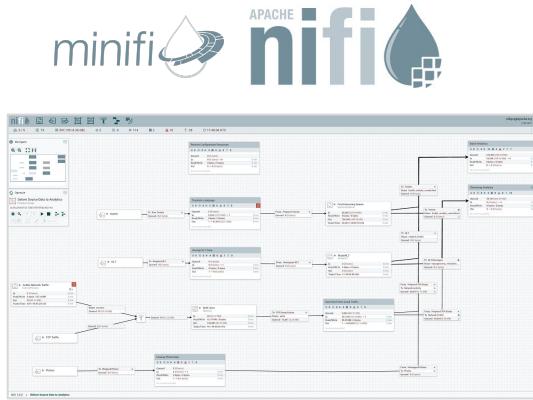
Compliance Requirements

CISO / VP of Platform

"Our homegrown ingestion tool has been around for 20+ years. The ingestion tool doesn't meet new requirements for Sarbanes Oxley, CCPA and GDPR."

Cloudera Flow and Edge Management

Move data from anywhere to anywhere with ease



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

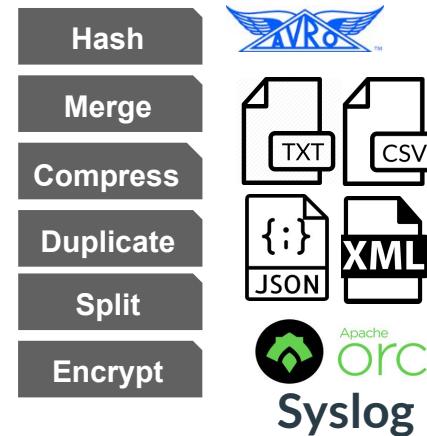
- Guaranteed Delivery
- Full data provenance from acquisition to delivery
- Diverse, Non-Traditional Sources
- Eco-system integration

An overview of NiFi capabilities

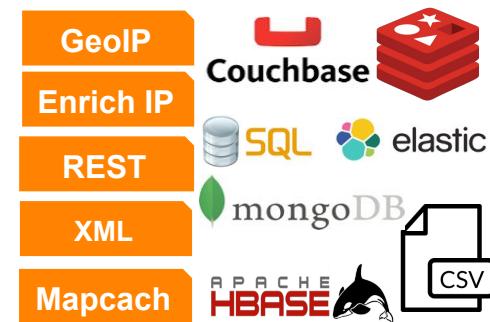
Data Ingest



Data Transformation

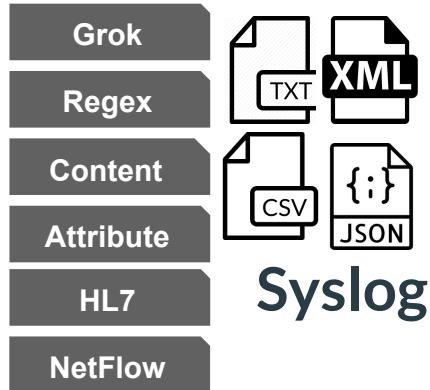


Data Enrichment

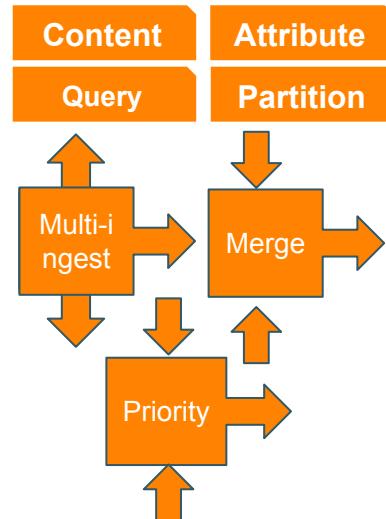


An overview of NiFi capabilities

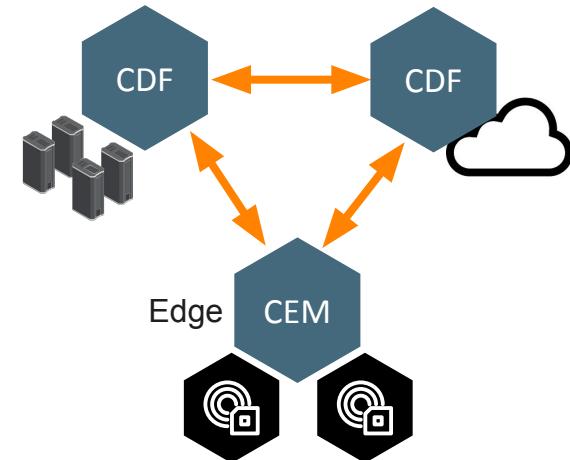
Parsing



Routing

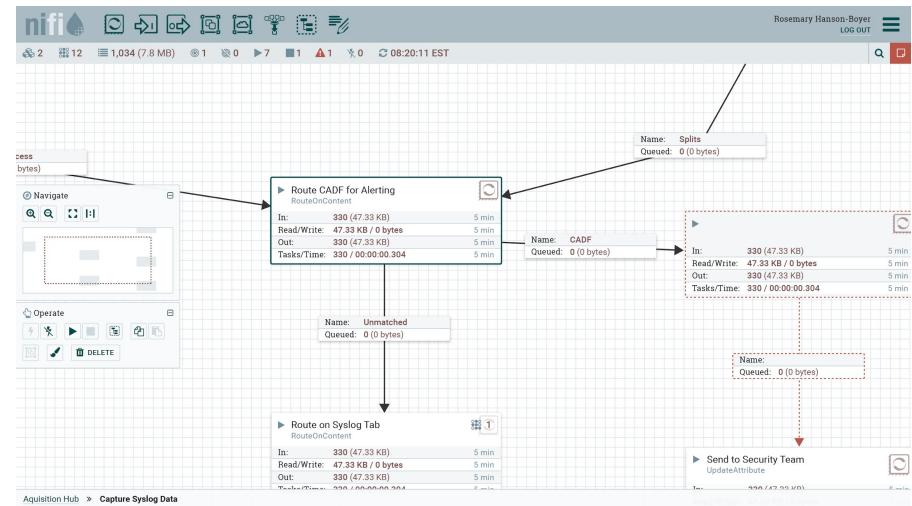


Data Movement



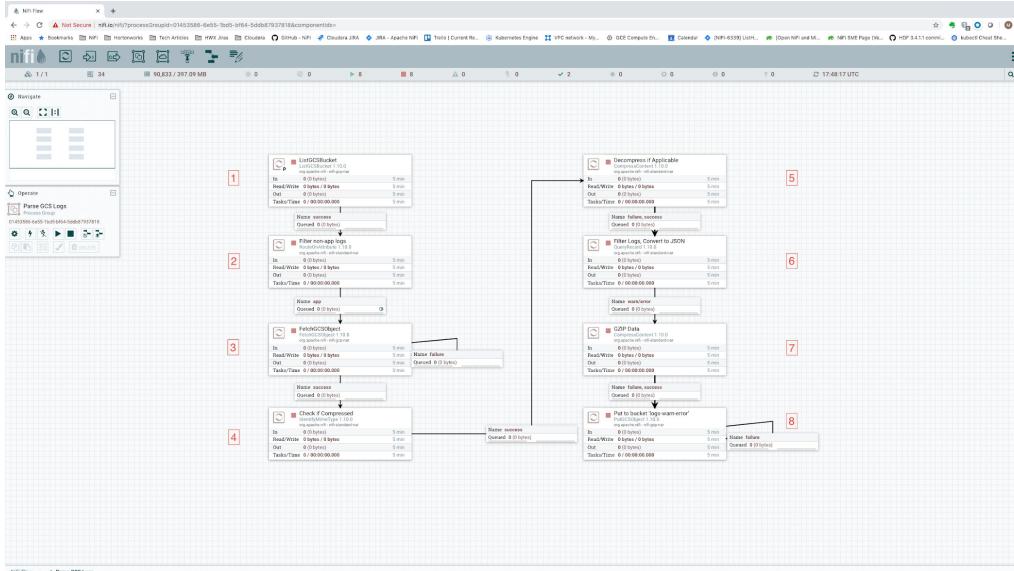
Apache NiFi High Level Capabilities

- Scale horizontal and vertically
 - Scale your data flow to millions event/s
 - Ingest TB to PB of data per day
- Adapt to your flow requirements
 - Back pressure & Dynamic prioritization
 - Loss tolerant vs guaranteed delivery
 - Low latency vs high throughput
- Secure
 - SSL, HTTPS, SFTP, etc.
 - Governance and data provenance
- Extensible
 - Build your own processors and Controller services (providers)
 - Integrate with external systems (Security, Monitoring, Governance, etc)



Processing one billion events per second with NiFi

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

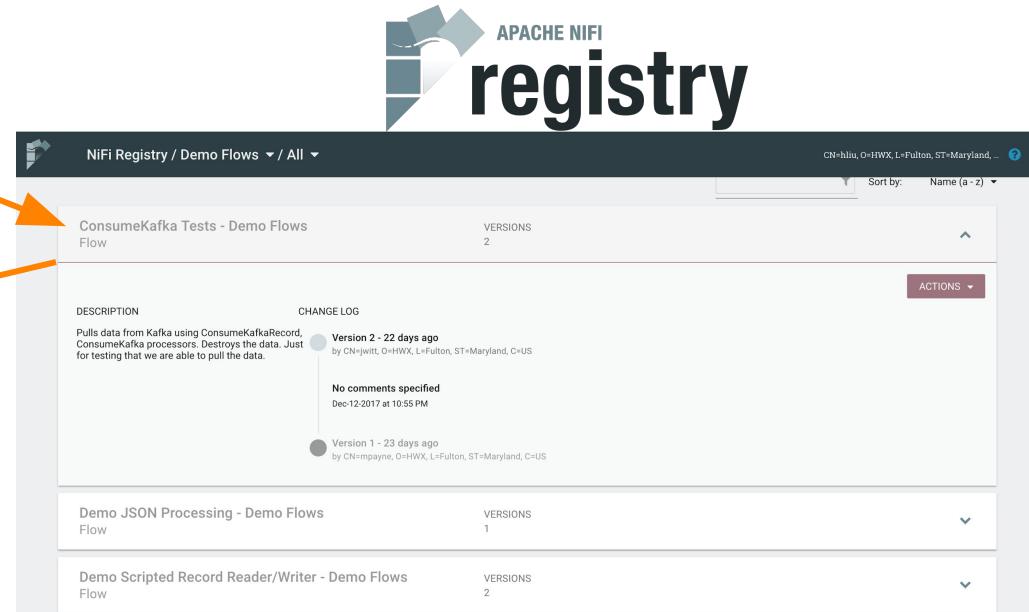


Nodes	Data rate/sec	Events/sec	Data rate/day	Events/day
1	192.5 MB	946,000	16.6 TB	81.7 Billion
5	881 MB	4.97 Million	76 TB	429.4 Billion
25	5.8 GB	26 Million	501 TB	2.25 Trillion
100	22 GB	90 Million	1.9 PB	7.8 Trillion
150	32.6 GB	141.3 Million	2.75 PB	12.2 Trillion

NiFi Flow Registry

ConsumeKafka Tests - Demo flows

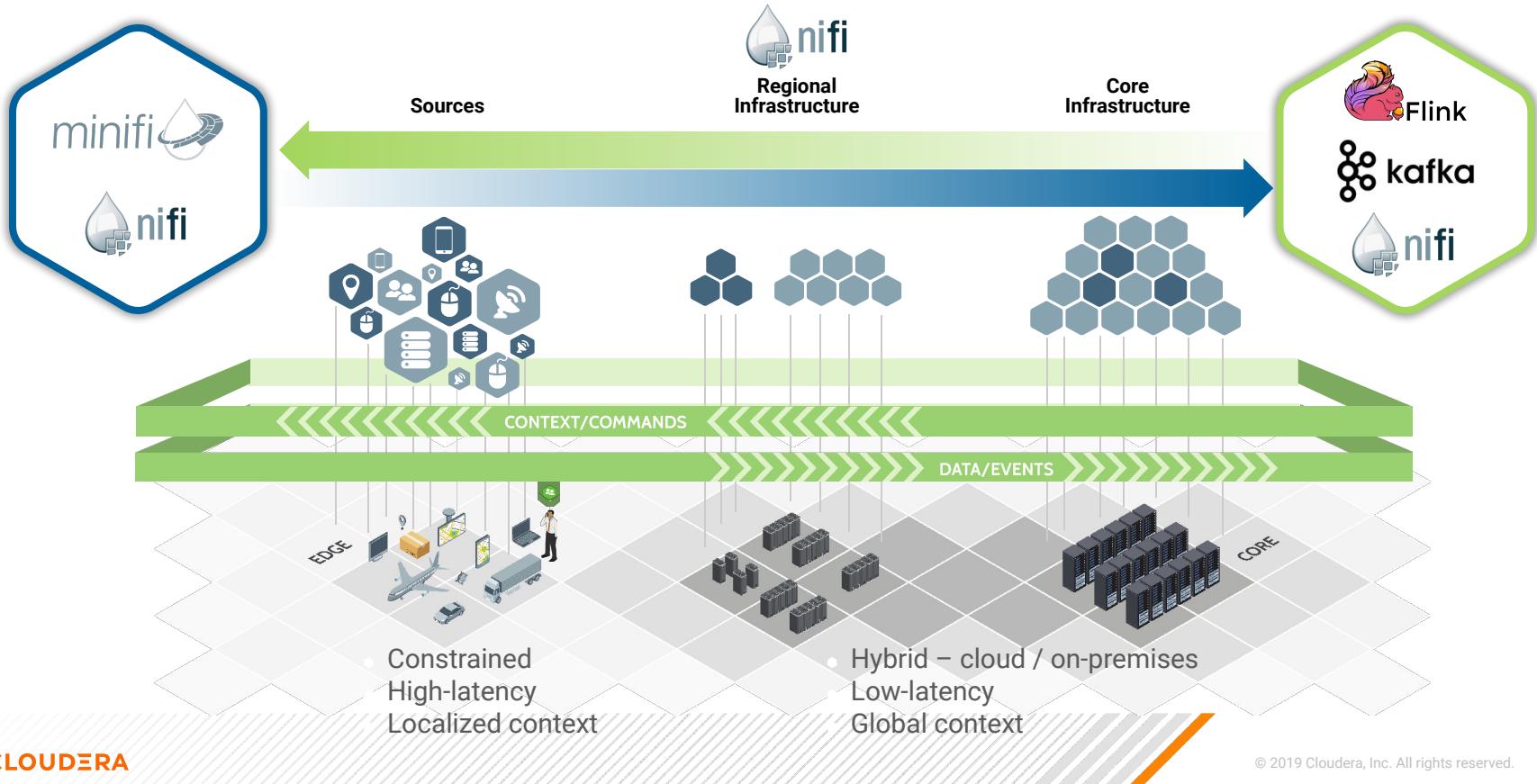
	Configure
Queued	Variables
In	Version
Read/Wri	▶ Start version control
Out	Enter group
✓ 0	5 min
✗ 0	▶ Start
	■ Stop



CLOUDERA

Edge Management

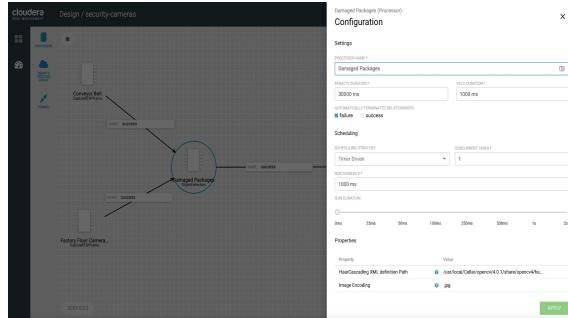
End to End data management



Cloudera EDGE Management

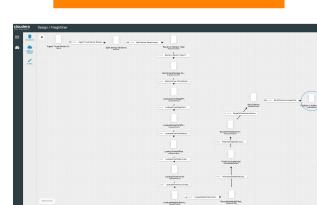
Edge device data collection and processing with easy to use central command and control

Edge Flow Manager

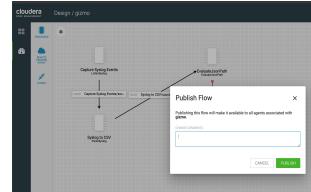


A lightweight edge agent that implements the core features of Apache NiFi, focusing on data collection and processing at the edge

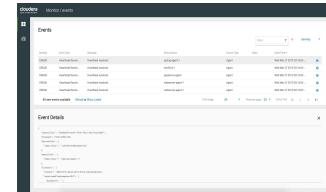
Flow Authorship



Flow Deployment



Flow Monitoring

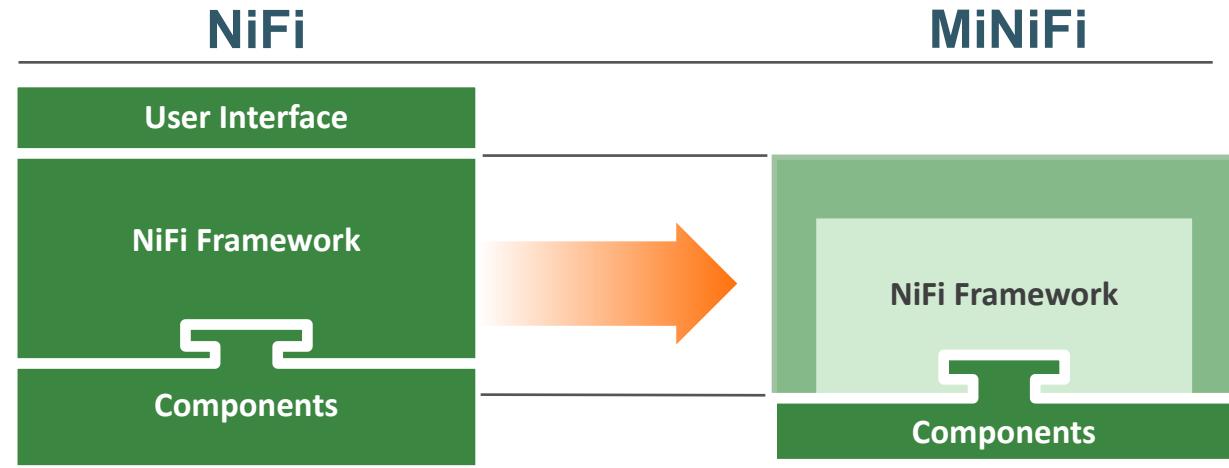


- Small footprint agent with MiNiFi
- Java and C++ agents
- Rich edge processors (edge collection & processing)
- End to end lineage and security

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

MiNiFi agents

- C++ and Java agents
- Security & Data provenance
- Guaranteed delivery
- Data buffering, Backpressure
- Prioritized queuing, Flow QoS
- Loss tolerance
- TensorFlow support of Edge AI
- Supports IoT ecosystem



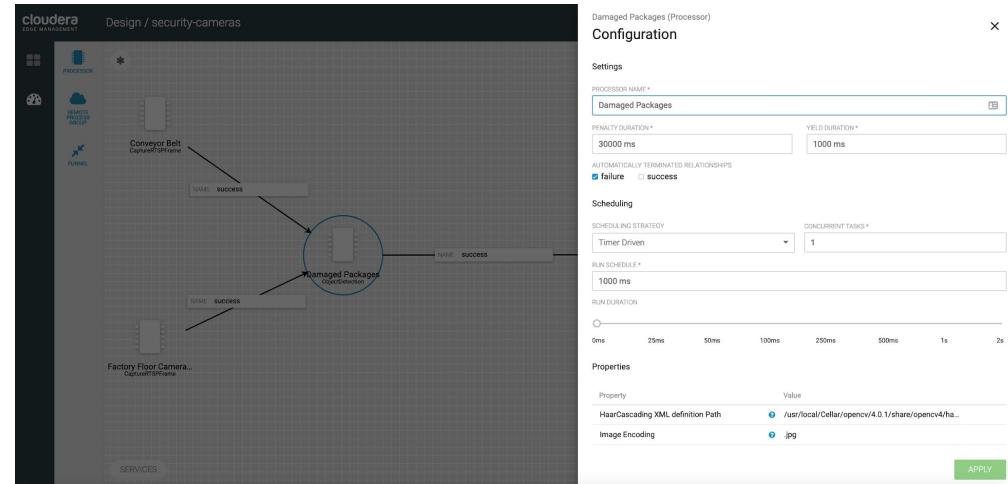
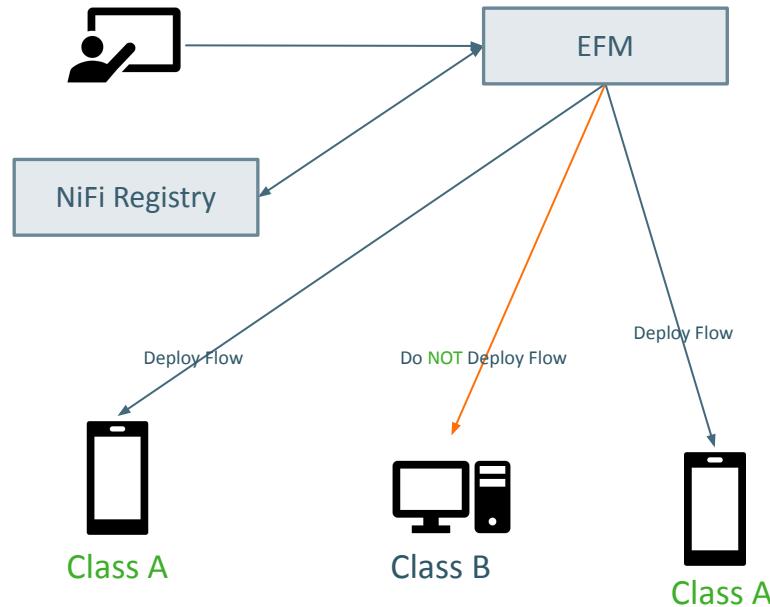
Edge Flow Manager (EFM)

- Manage thousands of deployed agents
- NiFi-like user interface to develop and deploy flow files to the edge
- Application lifecycle management
- Update and deploy ML model files to the edge agents
- Monitor thousands of edge agents
- Integration with NiFi Registry

The top screenshot shows the 'Design / security-cameras' configuration screen. It features a flow diagram with nodes: 'PREPROCESS', 'Conveyor Belt (Camera/Offshore)', 'RAMP', and 'Managed Package Repository'. Arrows labeled 'NAME success' connect the nodes. The right side of the screen contains configuration settings for a processor named 'Damaged Packages', including 'PENALTY DURATION' (3000 ms), 'YIELD DURATION' (1000 ms), and 'AUTOMATICALLY TERMINATED RELATIONSHIPS' (failure). The scheduling strategy is set to 'Timer Driven' with a 'PERIODIC SCHEDULE' of 1000 ms. The bottom screenshot shows the 'Monitor / events' page with a table of events. The table has columns: Severity, Event Type, Message, Event Source, Source Type, Class, and Date/Time. A filter bar at the top allows filtering by time range (All, hr, day, week) and severity (Severity, Info, Warn, Error, Critical). The table shows 40 new events available. An 'Event Details' modal is open, displaying a JSON object with fields like 'identifier', 'created', 'event_type', and 'event_time'.

EFM Command & Control

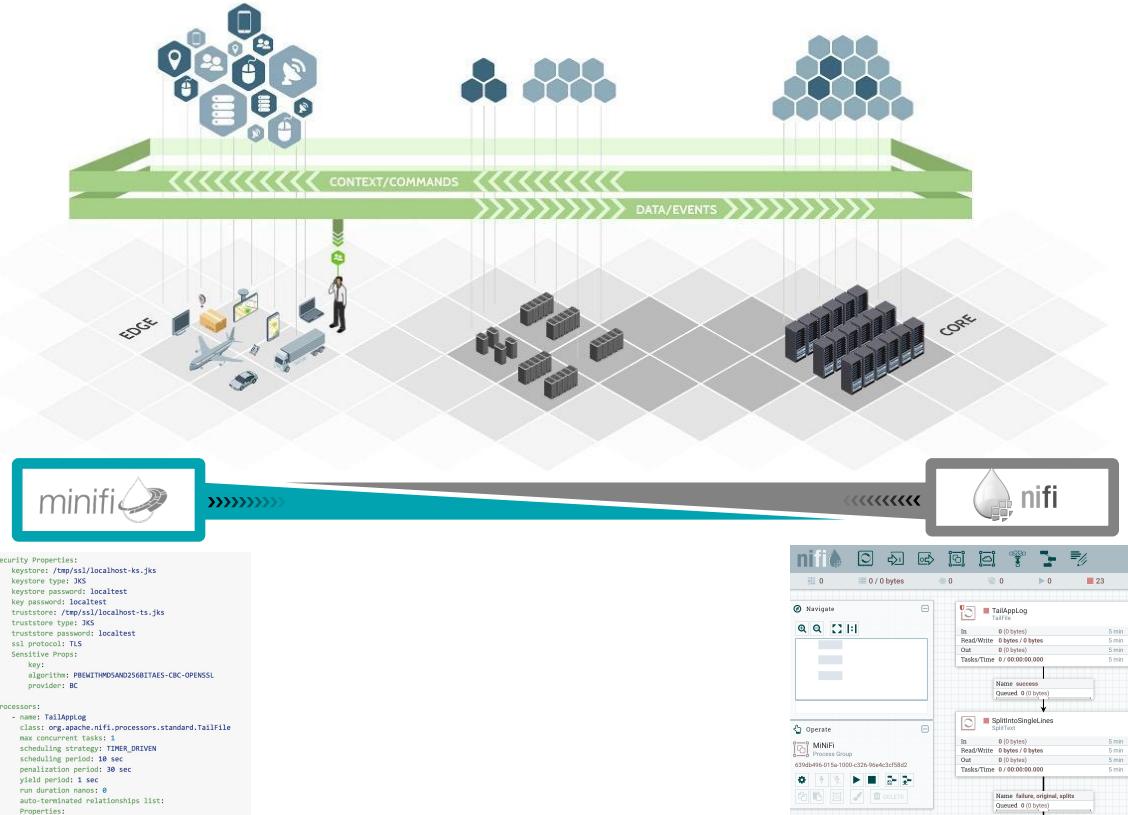
Flow Authorship & deployment



- User designs flow in EFM UI
- Completed flow is saved to NiFi Registry
- Class A flow is pushed to only class A devices

How Does MiNiFi Interact With NiFi?

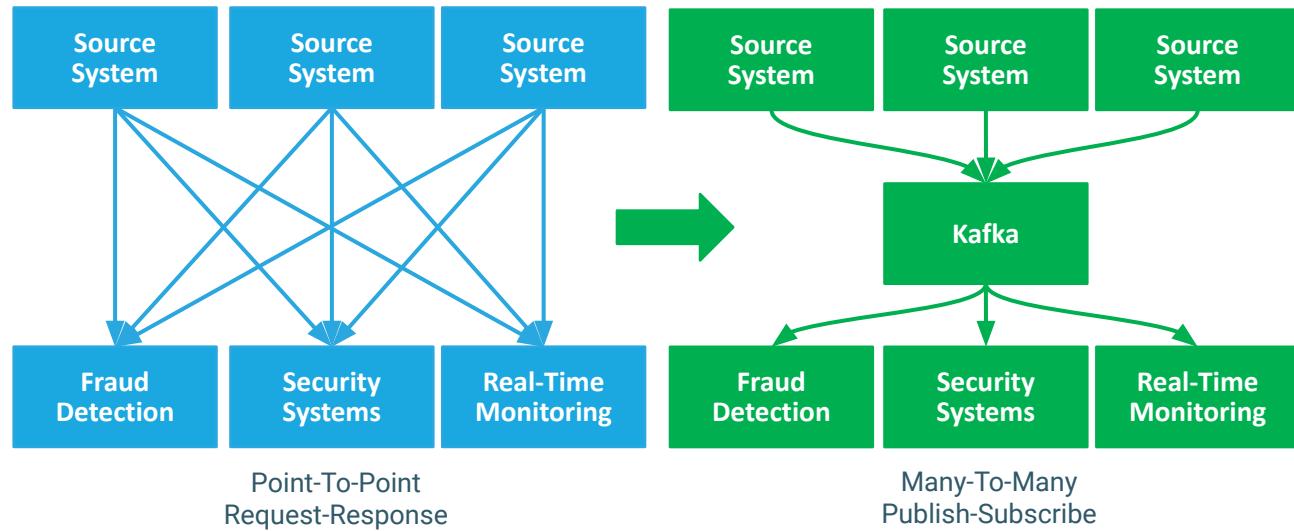
- MiNiFi
 - Receive flows
 - Collect data
 - Send for processing
- NiFi
 - Design flows
 - Aggregate data from many sources
 - Perform routing / analysis / SEP



Stream Messaging

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 Tooling

Kafka Services for Schema Management, Replication and Monitoring

Schema Registry

New Kafka Schema Governance

The screenshot shows the Schema Registry interface with the following details:

- All Schemas** tab selected.
- Search by name** input field.
- Sort by** dropdown set to **Last Updated**.
- Branch**: **MASTER**.
- Branch Description**: "MASTER branch for schema metadata 'syndicate-speed-event-avro'".
- Version Description**: "Enriched Speed Events from trucks in Kafka Topic".
- Schema Definition (JSON)**:

```
1 {  
2     "type": "record"  
3     "name": "syndicate-speed-event-avro"  
4     "fields": [  
5         {"name": "eventTime",  
6             "type": "string"  
7         },  
8         {"name": "eventTimeLong",  
9             "type": "long",  
10            "default": 0  
11        },  
12     ]  
13 }
```

Streams Messaging Manager (SMM)

Kafka Monitoring / Management Service

The screenshot shows the Streams Messaging Manager (SMM) interface with the following sections:

- Overview**: Shows Producers (16 of 45), Brokers (3 of 3), Topics (4 of 45), and Consumer Groups (3 of 5). It includes filters for **TOPICS (16)** and **BROKERS (0)**, and a search bar.
- Producers (16)**: A table listing producers with metrics like DATA IN, DATA OUT, and MESSAGES IN. One entry is highlighted: **minifl-eu1** (21k messages).
- Topics (4)**: A table listing topics with metrics like DATA IN, DATA OUT, and MESSAGES IN. One entry is highlighted: **gateway-west-raw-sensors** (406 KB).
- Consumer Groups (3)**: A table listing consumer groups with metrics like DATA IN, DATA OUT, and MESSAGES IN. One entry is highlighted: **minifl-eu1** (21k messages).
- Replicators**: A section showing replicator status for topics like **gateway-west-raw-sensors** and **gateway-central-raw-sensors**.

Streams Replication Manager (SRM)

New Kafka Replication Engine powered by MirrorMaker2

The screenshot shows the Streams Replication Manager (SRM) interface with the following sections:

- Cluster Replications**: A table showing cluster replications between **CDFClusterSc...** and **CDFClusterA...**. It includes columns for STATUS, SOURCE #, TARGET, TOPICS #, CONSUMER GROUPS #, THROUGHPUT #, REPLICATION LATENCY #, and CHECKPOINT LATENCY #.
- Throughput**: A line chart showing throughput over time (15:00 to 19:45) with values ranging from 1.4 to 2.5 MB/s.
- Replication Latency**: A line chart showing replication latency over time (15:00 to 19:45) with values ranging from 0 to 15 ms.
- Search By Topic Name**: A search bar to filter topics.
- Topics Table**: A table showing topics with columns for PARTITIONS #, CONSUMER GROUPS #, THROUGHPUT #, and REPPLICATION LATENCY #.

Apache Kafka Tooling

Kafka Services for Schema Management, Replication and Monitoring

Cruise Control

Intelligent Kafka Cluster
Rebalancing & Self Healing

The screenshot shows the Cloudera Manager interface for a 'OneNodeCluster'. The left sidebar includes sections for Clusters, Diagnostics, Audits, Charts, Administration, and Private Cloud. The main area is titled 'Cruise Control' under the 'Configuration' tab. It displays a list of 'Default Goals' under the 'Cruise Control Server Default Group'. The goals listed include: 'com.linkedin.kafka.cruisecontrol.analyzer.goals.RackAwareGoal', 'com.linkedin.kafka.cruisecontrol.analyzer.goals.ReplicaCapacityGoal', 'com.linkedin.kafka.cruisecontrol.analyzer.goals.DiskCapacityGoal', 'com.linkedin.kafka.cruisecontrol.analyzer.goals.NetworkInboundCapacityGoal', 'com.linkedin.kafka.cruisecontrol.analyzer.goals.NetworkOutboundCapacityGoal', 'com.linkedin.kafka.cruisecontrol.analyzer.goals.CpuCapacityGoal', 'com.linkedin.kafka.cruisecontrol.analyzer.goals.ReplicaDistributionGoal', 'com.linkedin.kafka.cruisecontrol.analyzer.goals.PotentialNwOutGoal', 'com.linkedin.kafka.cruisecontrol.analyzer.goals.DiskUsageDistributionGoal', 'com.linkedin.kafka.cruisecontrol.analyzer.goals.NetworkInboundUsageDistributionGoal', and 'com.linkedin.kafka.cruisecontrol.analyzer.goals.NetworkOutboundUsageDistributionGoal'. The interface also includes filters for Scope, Category, and Status.

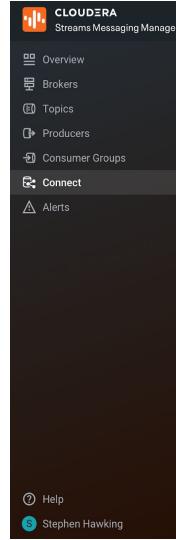
Kafka Connect

Simple Data Movement
for Kafka

The screenshot shows the Cloudera Manager interface for a cluster named 'srctwentyeight'. The main area is titled 'Kafka Connect' under the 'Status' tab. It features a 'Health Tests' section with a button to 'Create Trigger' and a 'Log Directory Free Space' section which notes that the role has no Log Directory configured. Below this is a 'Health History' section listing events such as 'Host Health Good' (Nov 20 5:06:40 PM), 'Host Health Bad' (Nov 20 5:06:35 PM), '3 Became Good' (Nov 20 5:01:14 PM), 'Unexpected Exits Good' (Nov 20 5:00:24 PM), and 'Process Status Good' (Nov 20 4:59:38 PM). To the right are several monitoring charts: 'Health' (percent good health 100), 'Role CPU Usage' (percent usage), and 'Resident Memory' (bytes usage). The bottom right corner shows a small orange bar with the text 'KAFKA_CONNECT (sam-in-srctwentyeight-1.vpc... 1.4G).

Kafka Connect Fully Integrated

- Support for Kafka Connect Runtime.
- Provision & manage Kafka Connect cluster via Cloudera Manager
- Deploy, monitor and management Kafka Connect connectors in SMM
- Supported set of sink and source connectors: HDFS, S3, File based
- Next supported connectors: JMS, MQ, MQTT, JDBC
- Custom/community connectors, can be integrated to SMM



A screenshot of the Cloudera Manager interface. The top navigation bar shows 'connect-default-cluster'. The main area is titled 'Connect Cluster' and displays 'Connector Overview' with statistics: TOTAL CONNECTORS 16, RUNNING CONNECTORS 10, FAILED CONNECTORS 6, DEGRADED CONNECTORS 0, and PAUSED CONNECTORS 0. Below this are three tabs: 'Connectors' (selected), 'Cluster Profile', and 'Alerts'. The 'Connectors' tab shows detailed lists for Source Connectors (10 entries), Topics (8 entries), and Sink Connectors (6 entries), each with columns for Name, Tasks, and status indicators. To the right, there's a detailed view for 'srctwentyeight / Kafka / samin-srctwentyeight-1' under 'Kafka Connect'. This view includes tabs for Status, Configuration, Processes, Commands, Charts Library, Audits, Log Files, Stacks Logs, and Quick Links. Under 'Status', there are sections for 'Health Tests' (with a 'Create Trigger' button) and 'Charts'. The 'Charts' section contains four line graphs: 'Health' (mostly green), 'Role CPU Usage' (mostly green), 'Important Events and Alerts' (mostly green), and 'Resident Memory' (mostly green). The bottom right corner of the interface has a copyright notice: '© 2019 Cloudera, Inc. All rights reserved.'

NiFi vs. Kafka Connect for Data Ingestion

Feature	NiFi	Kafka Connect
Data movement	Move data from any source to any destination. Kafka is one of many the supported systems.	
Richness of connectors	NiFi offers more extensive set of processors to read/write from other data source	
Data Pipeline	Usually data is not ready to be consumed by apps. NiFi can filter, transform, enrich data before even publishing it for consumption or processing.	
Edge collection	MiNiFi extends NiFi capabilities to the edge	
Data pipeline design	UI based design, Flow/Variable Registries	
Universality	Not all data sources are compatible with pub/sub model. NiFi is polyglot (Request/Response, Polling, etc)	
Simple data movement with ecosystem containing only Kafka	NiFi provides processors more most sources and sinks in enterprise and not just Kafka	
Data Ordering guarantees	Ordering of events from source to destination is not guaranteed	Ordering of events in Kafka partition is preserved when writing to downstream systems (HDFS, S3, ADLS)

Schema Registry

- Centralized registry to provide reusable schema
- Version management to define relationship between schemas
- Format validation to enable generic format conversion
- Avoid attaching schema to every piece of data
- Integration with the stack (NiFi, Kafka, SAM)

The screenshot shows the Schema Registry interface for a schema named "raw-truck_events_avro". The schema is defined as follows:

```
1 {
2   "type": "record",
3   "namespace": "hortonworks.hdp.refapp.trucking",
4   "name": "truckgeoevent",
5   "fields": [
6     {
7       "name": "eventTime",
8       "type": "string"
9     },
10    {
11      "name": "eventTimeLong",
12      "type": "long",
13      "default": 0
14    }
15 }
```

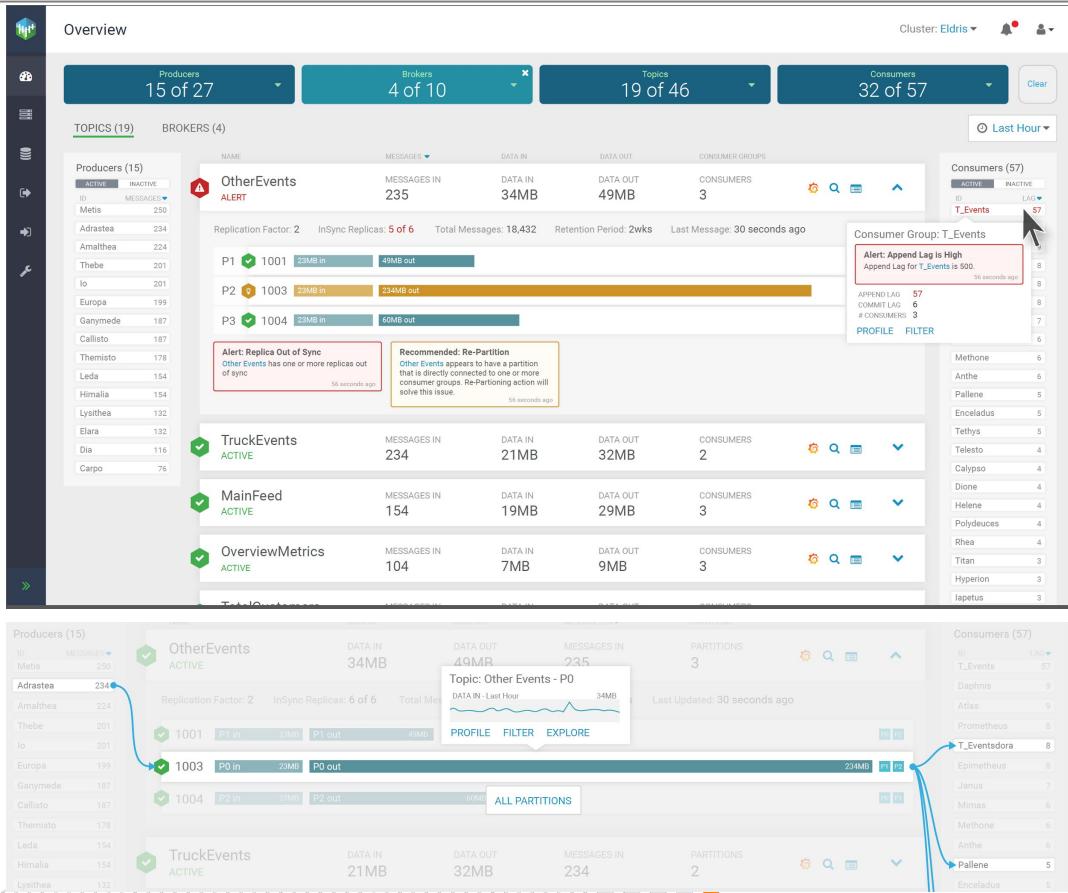
The schema is of type "avro" and is grouped under "truck-sen...". It has two branches: "2" and "0". The "2" branch is the active version (Version 2), while the "0" branch is disabled. The "2" branch was last updated 1m 11s ago and is in review. The "0" branch was last updated 2d 17h 26m 26s ago and is enabled.

The top screenshot shows the "Configure Controller Service" dialog for the "KAFKA TOPIC" field. A dropdown menu is open, showing options like "Use Embedded Avro Schema" and "Use Schema Name Property".

The bottom screenshot shows the "TruckGeoEvent" configuration dialog. It includes fields for "CLUSTER NAME" (set to "victoria"), "ZOOKEEPER CONNECTION URL" (set to "hdfvictoria0.field.hortonworks.com:2181,hdfvictoria1.fi"), "KAFKA TOPIC" (set to "truck_events_avro"), "CONSUMER GROUP ID" (set to "truck_events_avro_860"), and "Output" (a detailed schema definition). A callout box points to the "Output" section, stating: "After the user selects a Kafka topic, the stream analytics tool will fetch the schema for the Kafka Topic from the shared Schema Registry".

Streams Messaging Manager (SMM)

- Single Monitoring Dashboard for all your Kafka Clusters across 4 entities
- Message viewer
- Topic administration: CRUD
- Notification on available metrics
- REST as a First Class Citizen
- Designed for the Enterprise
 - Support for Secure Kafka cluster
 - Rich Access Control Policies (ACLS)



Kafka Replication Use Cases



Disaster Recovery

In an event of a partial or complete datacenter disaster, providing failover/fallback to a secondary cluster in a different region / DC



Centralized Analytics

Aggregate data from multiple Kafka clusters into one location for organization-wide analytics



Geo-Locality

Active-active geo-localized deployments allows users to access a near-by data center to optimize their architecture for low latency and high performance.



Workload Isolation

Creation of different envs for SDLC: Dev, Test, Prod. Clusters for specific use case cases (ETL, ingestion, analytics, etc)



Data Movement / Deployment

Use Kafka to synchronize data between on-prem applications and cloud deployments

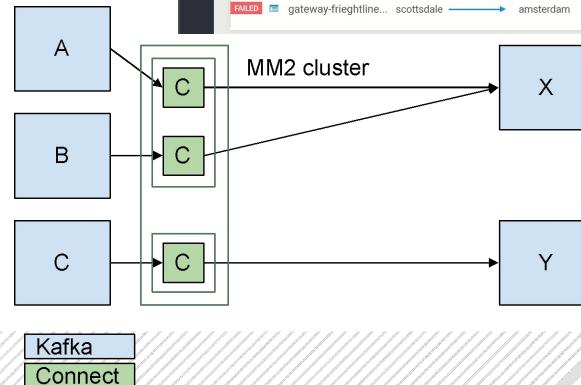
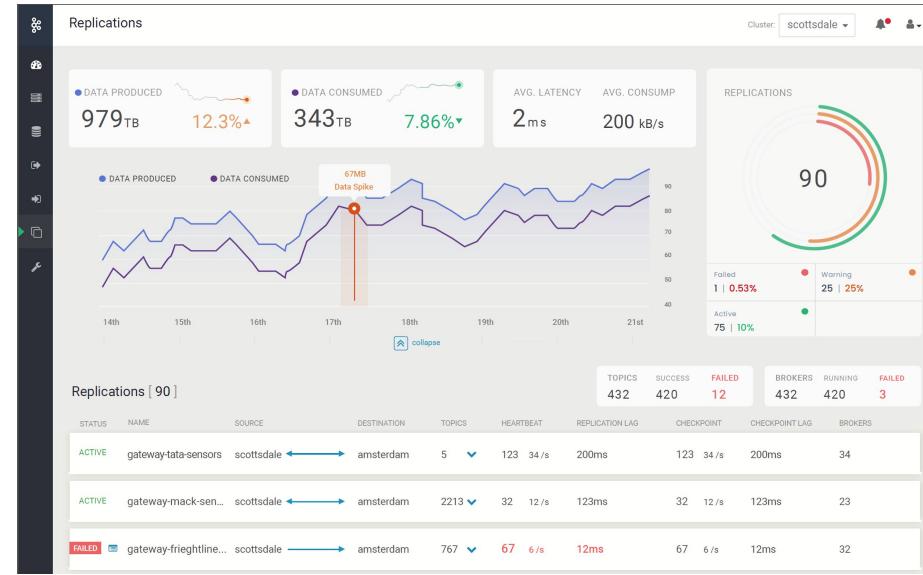


Legal / Compliance

Different data storage and security policies require clusters to be created in region but data still needs to be shared.

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 & smart client for simplified failover and fallback
- Integrate replication monitoring with SMM



Kafka Cruise Control Service

Managing Kafka Clusters at Large Scale

Problem Statement / Requirements

- Platform teams need first class management services that address hard problems such as frequent hardware/vm failures, cluster expansion/reduction, and load skew among brokers.
- These class of problems require the need to **balance the cluster intelligently with automated anomaly detection and remediation**.

Solution / Benefit

- Support for Cruise Control in CSP, a open source project created by LinkedIn.
- Includes admin operations such add/decommission/demote brokers, rebalance the cluster.
- Generation of rebalancing proposals based on goals with automated anomaly detection and remediation
- Foundation for providing first class Kafka Cloud Workload in Cloudera Data Hub

The screenshot displays the Cloudera Data Hub Kafka Cruise Control Service interface. It includes:

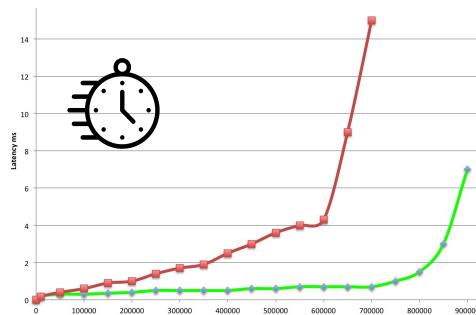
- Kafka Cluster Administration:** A table showing brokers (91, 92, 93) with their #Replicas (81), #Leaders (81), #Out of Sync Replicas (0), and #Offline (0).
- PLE Flags:** Options for Concurrent Leader Movements (radio button selected) and DryRun (checkbox checked).
- Kafka Server Load:** A table showing hostnames (vertt11-c0b7-field, vertt11-c0b8-field, vertt11-c0b9-field) with their broker details and network rates.
- Kafka Broker Load:** A table showing brokers (91, 92, 93) with their broker details and network rates.
- Clear Response:** A code block showing a JSON response with fields like "summary", "recentLeaderMovements", "excludedBrokersForLeadership", "queuedReplicaMovements", "intraBrokerReplicaMovements", "recentLeaderMovements", "dataInNetwork", "monitoredPartitionsPercentage", "extendedTopics", "unleaderedReplicas", and "extendedBrokersForReplicaMove".

Stream Processing and Analytics

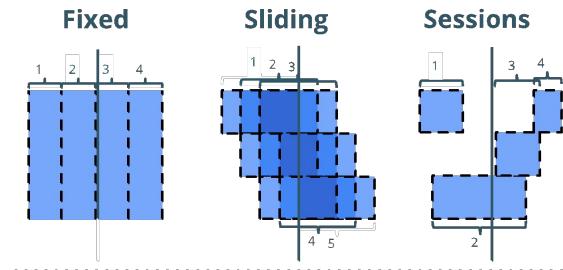
Choosing a streaming engine

Things to consider

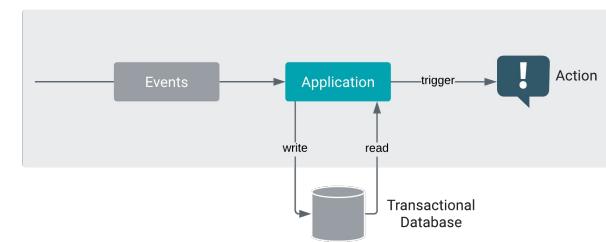
Latency



Windowing



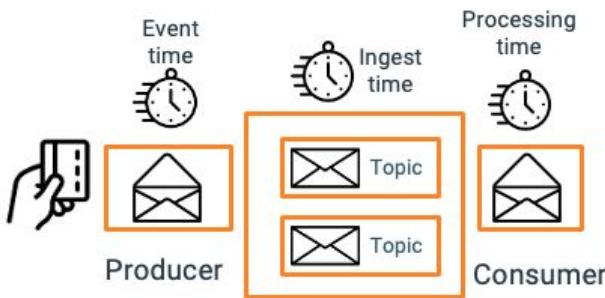
State Management



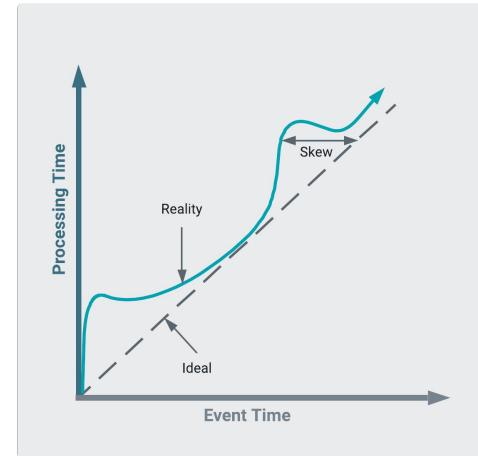
Choosing a streaming engine

Things to consider

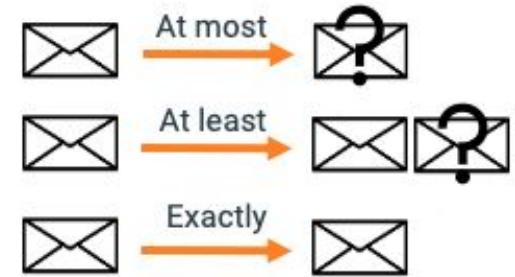
Event time



Watermarking



Processing Semantics



Choose The Right Stream - Download the Whitepaper

CLOUDERA

WHITE PAPER

Choose the Right Stream Processing Engine for Your Data Needs



bit.ly/choose-the-right-stream



Streaming Engines in a nutshell

Already using **Spark**?

Need highest throughput?

Want unified batch/stream?

Don't need low latency?

Don't need advanced time/state features?



Like **Kafka**?

Want low latency?

Want just Kafka?

Only need microservices?

Don't need advanced Time/State features OOTB?



Like **Flink**?

Want flexibility?

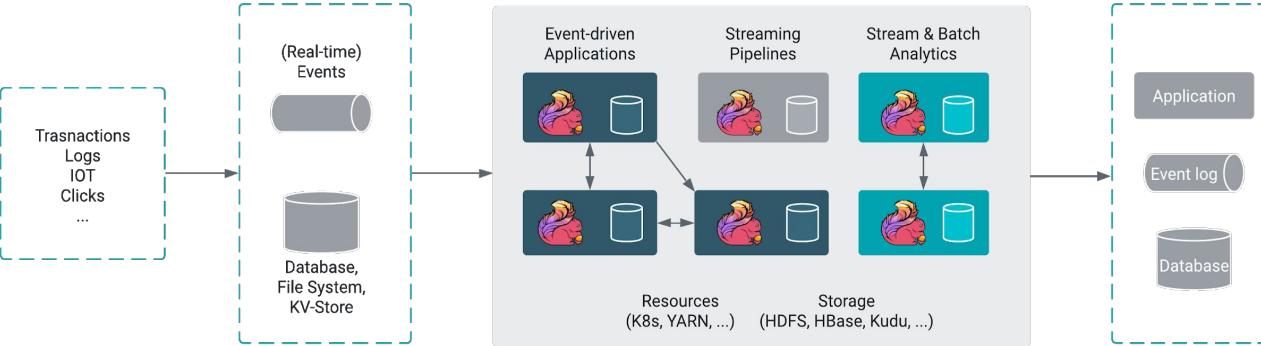
Want low latency?

Want leading-edge technology?



Apache Flink

- A distributed processing engine for stateful computations
- Manages 10s TBs of state
- In-memory processing at scale
- Flexible and expressive APIs
- Guaranteed correctness & Exactly-once state consistency
- Event-time semantics
- Flexible deployment & large ecosystem (K8S, YARN, S3, HDFS..)
- Fully integrated with CM



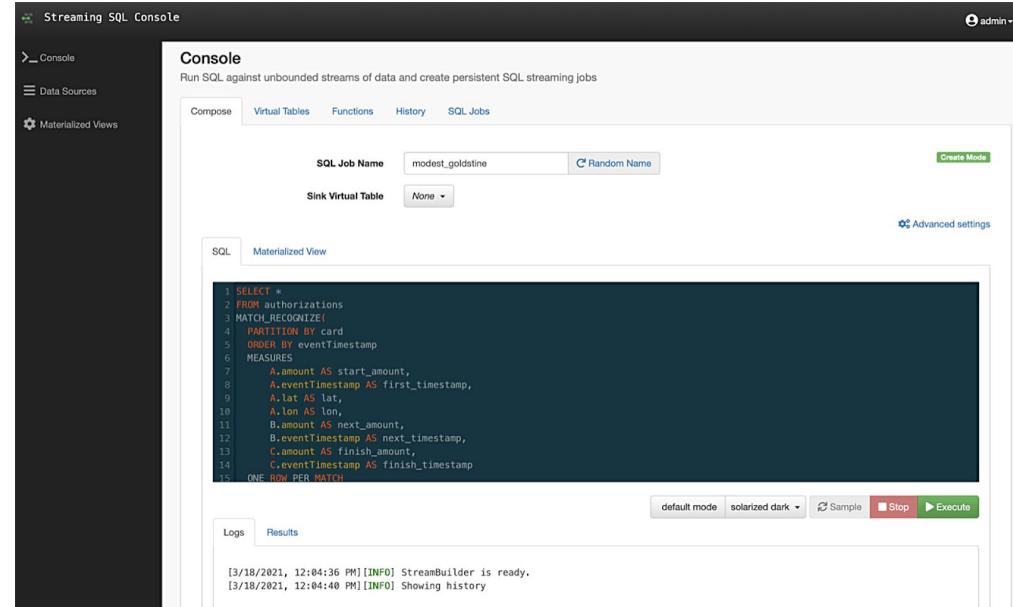
The screenshot shows the Cloudera Manager interface. On the left, there's a navigation bar with "Clusters", "Hosts", "Diagnostics", "Audits", "Charts", "Backup", and "Administration". Below it, a "Status" section displays the status of various services: CDH 6.3.0 (Parcels), 8 Hosts, Flink (2 instances), HDFS-2 (3 instances), Kafka-2, Schema Registry, Streams Metrics, Streams Replication, YARN (1 instance), and ZooKeeper-2. A "Health Tests" section shows "Create Trigger" and "Charts" options. The main area shows a "Trucking Streaming Analytics Flink App" running with ID: 72c0982cf7fbcb027e2872926070558, Start Time: 2019-09-17 09:53:51, Duration: 1d 4h 52m. The "Overview" tab is selected, showing a complex streaming job graph with nodes like "Source: Kafka TruckGeoDrive", "Parallelism: 6", "Stream Join using Internal Join", "Parallelism: 6", and "Window(TumblingEventTimeWindows(180000), EventTimeTrigger)". The graph also includes "Source: Kafka SpeedGeoDrive", "Parallelism: 6", and "Parallelism: 6". A "Metrics" table at the bottom provides detailed metrics for the job components.

Name	Status	Records Received	Bytes Sent	Tasks
Source: Kafka SpeedGeoStream	RUNNING	0 B	0	3.16 MB
Source: Kafka TruckGeoStream	RUNNING	0 B	0	4.09 MB
Stream Join using Internal Join -> Timestamps/Watermarks	RUNNING	7.48 MB	9,980	4.19 MB
Window(TumblingEventTimeWindows(180000), EventTimeTrigger)	RUNNING	4.58 MB	4,994	0 B

Flink SQL: Complex event processing made easy

Agile Streaming App Development using SQL

- SQL Console for writing and scheduling SQL Streaming jobs
- Unified APIs for streaming data and data at rest
- Run the same query on batch and streaming data
- ANSI SQL: No stream-specific syntax or semantics!
- Many common stream analytics use cases supported



The screenshot shows the Flink Streaming SQL Console interface. On the left is a sidebar with 'Console', 'Data Sources', and 'Materialized Views'. The main area has tabs for 'Compose', 'Virtual Tables', 'Functions', 'History', and 'SQL Jobs'. Under 'Compose', there's a 'SQL Job Name' input set to 'modest_goldstine' with a 'Random Name' button, and a 'Sink Virtual Table' dropdown set to 'None'. Below these are tabs for 'SQL' and 'Materialized View', with the 'SQL' tab selected. A code editor displays the following Flink SQL query:

```
1 SELECT *
2 FROM authorizations
3 MATCH_RECOGNIZE(
4   PARTITION BY card
5   ORDER BY eventTimestamp
6   MEASURES
7     A.amount AS start_amount,
8     A.eventTimestamp AS first_timestamp,
9     A.lat AS lat,
10    A.lon AS lon,
11    B.amount AS next_amount,
12    B.eventTimestamp AS next_timestamp,
13    C.amount AS finish_amount,
14    C.eventTimestamp AS finish_timestamp
15   ONE ROW PER MATCH
```

At the bottom, there are 'Logs' and 'Results' tabs, with 'Logs' selected. The logs pane shows two entries:

```
[3/18/2021, 12:04:36 PM][INFO] StreamBuilder is ready.
[3/18/2021, 12:04:40 PM][INFO] Showing history
```

Below the logs are buttons for 'default mode', 'solarized dark', 'Sample', 'Stop', and 'Execute'.

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 for a cluster named 'CDFClusterAmsterdam'. The 'SQLStreamBuilder' tab is active. In the main pane, a SQL query is shown:

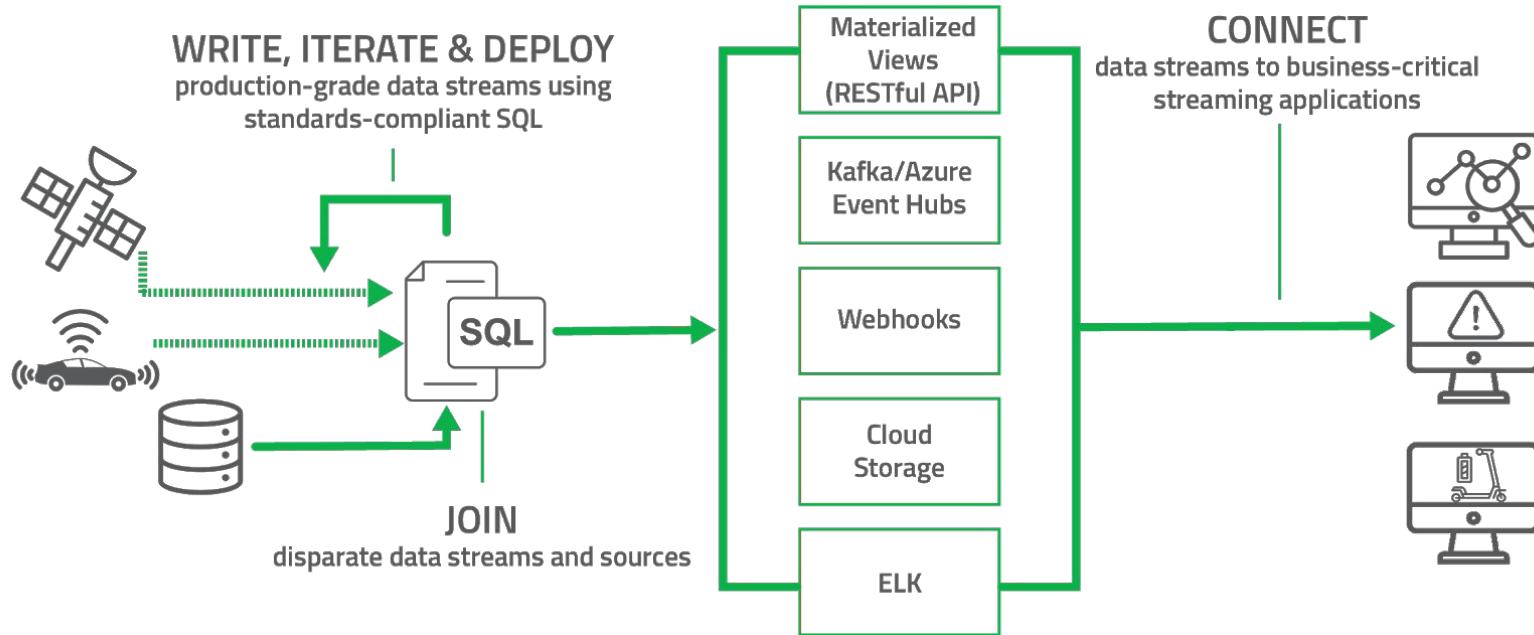
```
1 | SELECT TUMBLE_END(gEO_events.eventTimestamp, INTERVAL '3' MINUTE) as windowEnd,
2 |     geo_events.driverId,geo_events.driverName,gEO_events.route,
3 |     avg(speed_events.speed) as driverAvgSpeed
4 | FROM
5 |     geo_events,
6 |     speed_events
7 | WHERE
8 |     geo_events.driverId = speed_events.driverId 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,
```

The Log pane at the bottom displays the following log entries:

```
[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.
```

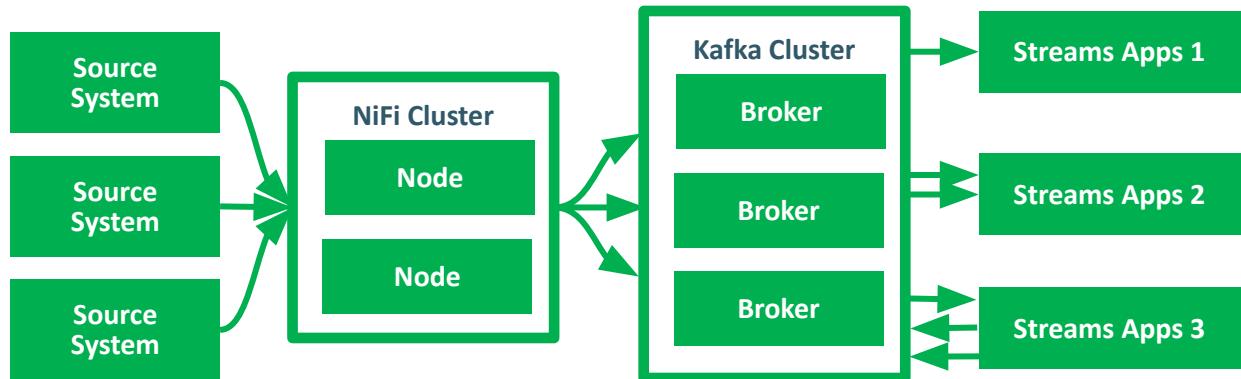
Streaming SQL

Democratizing access to streams of data via structured query language



Kafka Streams

- A client library for applications and microservices on top of Kafka
- Exists in Java and Scala
- Elastic, highly scalable, fault-tolerant
- Supports At-Least-Once and Exactly-once semantics inside Kafka
- Deploy to containers, VMs, bare metal, cloud



Shared Data Experience

Cloudera Manager support, Virtual Cluster Support

Support for Kafka and NiFi Compute Clusters

Home

Status All Health Issues 0 Configuration Actions All Recent Commands

Clusters

- Datalake Cloudera Runtime 7.0.3 (Parcels)
- FlinkStreamProcessingCluster Compute Cluster, Cloudera Runtime 7.0.3 (Parcels)
- KafkaMessagingCluster Compute Cluster, Cloudera Runtime 7.0.3 (Parcels) **Selected**
- NiFiFlowCluster Compute Cluster, Cloudera Runtime 7.0.3 (Parcels)

Customize

Cloudera Management Service

CloudBeams CloudBeams Management ...

This screenshot shows the Cloudera Manager Home page. The KafkaMessagingCluster is highlighted with an orange border. The page includes sections for Status, Clusters, and Cloudera Management Service.

KafkaMessagingCluster Actions

Status Health Issues Configuration

Status

Data Context: SharedSecurityAndGovernanceContext

- Ranger
- Atlas
- HDFS

Charts

Cluster CPU

Compute Cluster, Cloudera Runtime 7.0.3 (Parcels)

Cluster Disk IO

Cluster Network IO

NiFiFlowCluster Actions

Status Health Issues Configuration

Status

Data Context: SharedSecurityAndGovernanceContext

- Ranger
- Atlas
- HDFS

Charts

Cluster CPU

Compute Cluster, Cloudera Runtime 7.0.3 (Parcels)

Cluster Disk IO

Cluster Network IO

Shared Security Context from DataLake consisting of Ranger and Atlas

This screenshot shows three separate Cloudera Manager cluster status pages. Each page has a section for 'Data Context: SharedSecurityAndGovernanceContext' which lists Ranger, Atlas, and HDFS. The clusters shown are KafkaMessagingCluster, NiFiFlowCluster, and another cluster (partially visible). Each cluster has its own set of charts for CPU, Disk IO, and Network IO. An orange box highlights the 'Data Context' section across all three pages, indicating shared security context.

Authorization & audit with Apache Ranger

- Single pane of glass security: NiFi, Kafka, SMM, SR, etc
- Fine-grained access control
- Role & Attribute Based ACL
- Integration with LDAP/AD
- Centralized audit

The screenshot displays the Apache Ranger web interface across three main sections: Service Manager, Audit, and Access.

Service Manager: Shows three service registries: SCHEMA-REGISTRY (cm_schema_registry), KAFKA (cm_kafka), and NIFI (cm_nifi). Each registry has a + icon for adding resources and a delete icon.

Audit: Displays a table of audit logs for the user sales_user1. The logs show various events (Event Time) such as 07/25/2018 05:02:33 PM, 07/25/2018 05:02:32 PM, etc., involving the service hdinsightsecurekafka_kafka and resource salesevents topic. The access type is consistently consume, and the result is Allowed. The last updated time is 07/25/2018 05:03:03 PM.

Access: A policy creation form for a new policy named "kafka_exclude".

- Policy Type:** Access
- Policy ID:** 23
- Policy Name:** kafka_exclude (enabled)
- Topic:** kafka1_topic (exclude)
- Audit Logging:** YES
- Description:** (empty)

Below the form is a section titled "Allow Conditions:" which includes "Select Group" and "Select User" fields, and a "Policy Conditions" table with columns for Add Conditions, Publish, Consume, Configure, Describe, Create, Delete, and Kerberos Admin. The "Add Conditions" row shows "Kafka Admin".

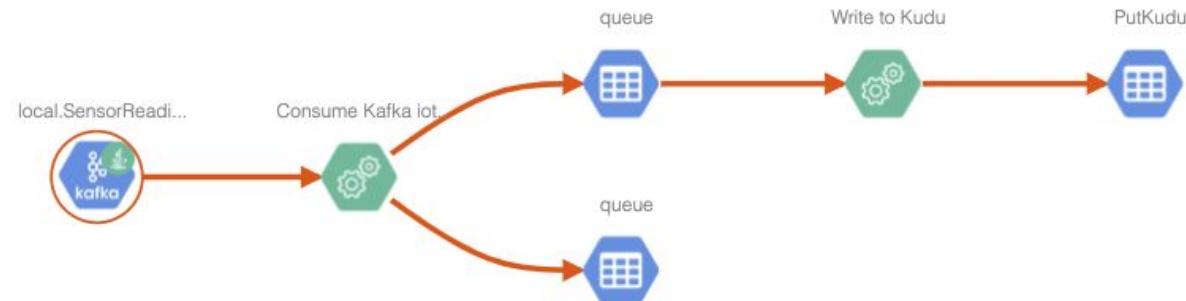
Governance & Lineage with Apache Atlas

- End-To-End lineage
- Integration with NiFi, Spark and Kafka for automatic lineage capture
- Business taxonomy
- Integration with Ranger for Attribute Based Access
- Multi-facets search engine

The screenshot shows the Apache Atlas web interface. At the top, there are tabs for SEARCH, CLASSIFICATION (which is selected), and GLOSSARY. Below the tabs, there is a 'Flat' button (which is selected) and a 'Tree' button, along with a '+' icon and a 'Search Classification' input field containing 'PII'. On the right side, there is a sidebar titled 'PII' with the sub-section 'Personally Identifiable Information'. Underneath, it says 'Attributes:' with a '+' icon. The main content area shows a table with one record:

Name	Owner	Description	Type	Classifications	Term
trip_zone_lookup	hadoop		hive_table	PII	+ X

At the bottom of the table, there are checkboxes for 'Exclude sub-classifications' and 'Show historical entities'.



CDP DataHub Cloud Services

How Does CDP DataHub Bring Value to End Users?



Platform Team



"Sorry. It will take me 2 weeks to provision a fully secured and governed Kafka and NiFi Cluster?"



Sure. I'll get a cluster created for you in the next 30 mins. Do you want on Azure or AWS?



Data/App Developer

Hm..you can spin up clusters for me easily. But what is the point if i can't get data in & out and move it around easily?

"Wow. I love this code-less approach of NiFi to get data in and out of Kafka, DW, DE Cluster, or any cloud source"

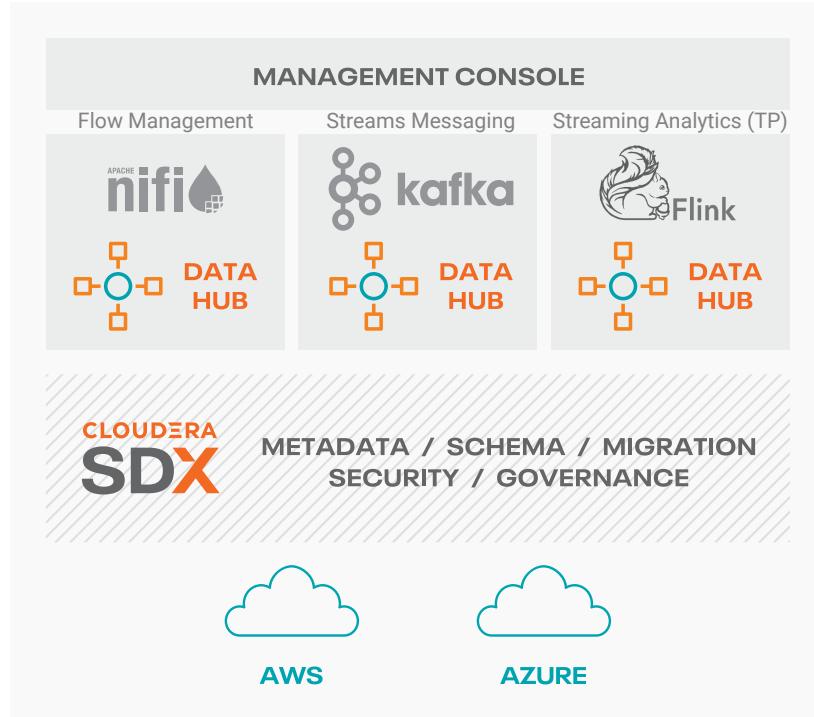


Data Steward

I have no way to track where data is coming from and where its going and security policies are spread everywhere

"Lineage and Provenance of my Data Flows. Security policies for NiFi, Kafka, Hive, Spark and Flink are in a single place with Audit"

CDF on CDP Public Cloud Data Hub



CDP Management Console runs as a web service hosted and managed by Cloudera

Data Hub clusters running Kafka, NiFi and Flink hosted in the customer's cloud environment, but managed by the CDP Management Console

Shared Data Experience (SDX) technologies form a secure and governed data lake backed by object storage (S3, ADLS, GCS)

CDP services are optimized for the elastic compute & 'always-on' storage services provided by the customer's chosen cloud provider

Flow Management Cluster Definition (CDP 7.2.8)

Apache NiFi in CDP Data Hub



Latest NiFi innovation

Apache NiFi 1.11.4

Apache NiFi Registry 0.5



Build flows immediately

Pre-created Ranger policies
allowing immediate access to NiFi

Support for Kerberos
Principal/Password eliminating
need for managing keytabs

Pre-configured with TLS settings
through
RestrictedSSLContextProvider



CDP Platform integration

Managed FreeIPA server to fully **secure**
clusters by default

- Kerberos authentication
- SSL/TLS wire encryption

Integrated with **Apache Ranger** for
managing **granular authorization**
policies on NiFi resources

Protected against cloud infrastructure
failures - **VM reprovisioning**

Pre-configured Atlas lineage reporting

Flow Management in Data Hub

Apache NiFi 1.11 Highlights

Parameter Support across environments

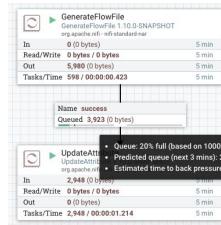
Allows parameterization of all processor properties

Support for sensitive parameters

Allows easy flow promotion through NiFi registry from Dev to Prod

Predictive Monitoring

Queue Length and time to Backpressure are now predicted



<https://www.youtube.com/watch?v=Tt8TSIHu7PE>

For more information, check out: <https://medium.com/@abdelkrim.hadjidj>, <https://www.datainmotion.dev>

General Improvements

Support for "public" (accessible to remote site to site clients) ports for any processor

Support for flow versioning across NiFi versions

New Retry FlowFile processor for improved error handling

Encrypted content and flow file repositories (preview)

Streams Messaging Cluster Definition (CDP 7.2.8)

Apache Kafka in CDP Data Hub



Latest Kafka innovation

Apache Kafka 2.5

[LDAP Callback Handler](#) enabling
SASL/PLAIN authentication



Powerful Kafka Tools

[Cloudera Streams Messaging Manager \(SMM\)](#) for Kafka monitoring and operations

[Cloudera Schema Registry](#) to centrally manage your schemas for Kafka and NiFi



CDP Platform integration

Managed FreeIPA server to fully [secure clusters by default](#)

- Kerberos authentication
- SSL/TLS wire encryption

Integrated with [Apache Ranger](#) for managing [granular authorization policies](#) on topics

Protected against cloud infrastructure failures - [VM reprovisioning](#)

Streaming Analytics Cluster Definition (CDP 7.2.8)

Apache Flink in CDP Data Hub



Latest Flink innovation

Apache Flink 1.10



Cloud optimized

Store application data like
[checkpoints / savepoints](#) on cloud
storage

Scale worker nodes [up and down](#)

Heavy / Light Duty [templates](#) for
different workloads



CDP Platform integration

Managed FreeIPA server to fully [secure](#)
clusters by default

- Kerberos authentication
- SSL/TLS wire encryption

Data access secured by [Apache Ranger](#)
for managing [granular authorization](#)
[policies](#) data sources

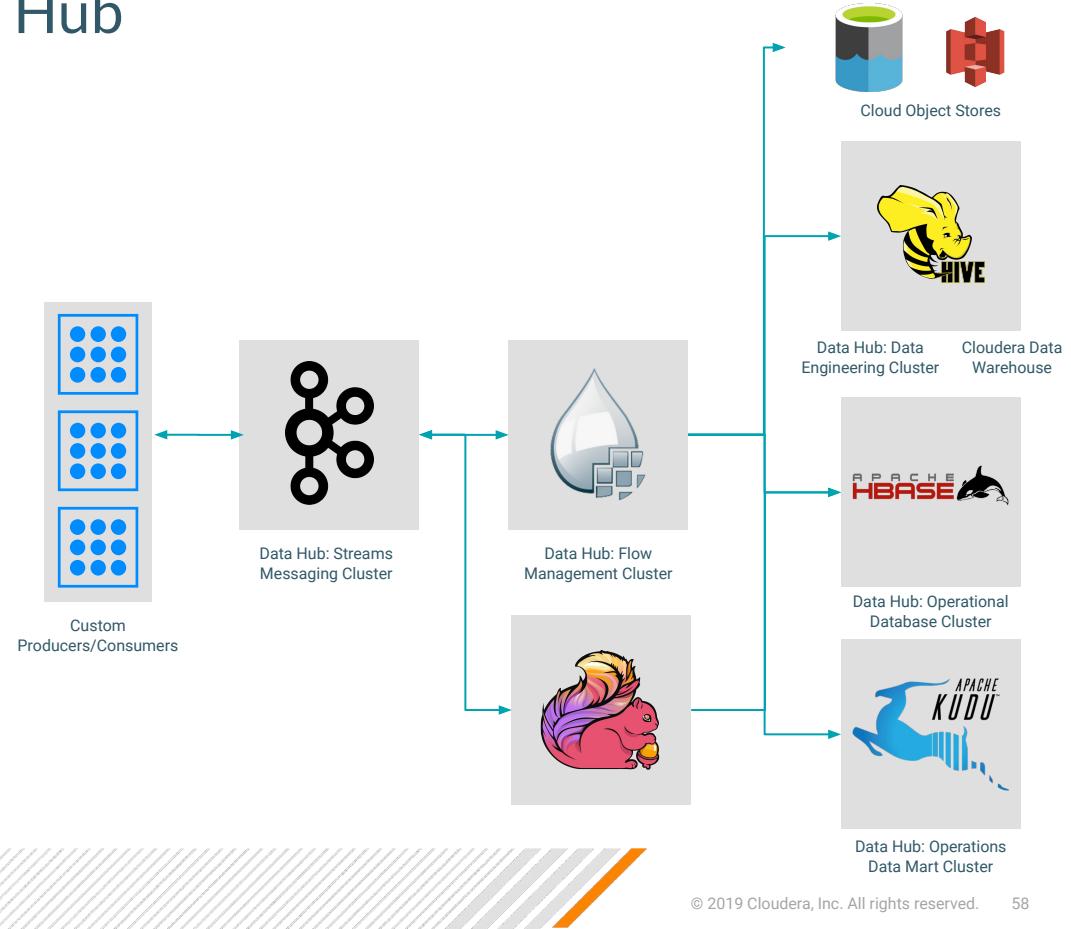
Protected against cloud infrastructure
failures - [VM reprovisioning](#)

[Pre-configured Atlas lineage integration](#)

CDF Use Cases in CDP Data Hub

Cloud Native Streaming / CDP Ingest

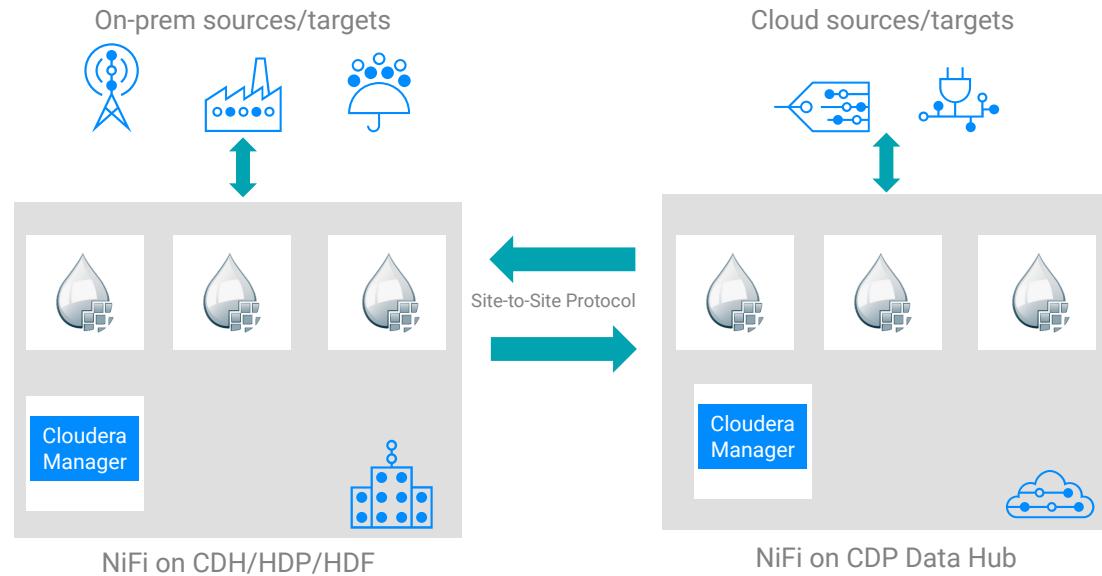
- Enables cloud native streaming applications
- Use NiFi or custom producers/consumers to interact with Kafka
- Use NiFi to power CDP Ingest
 - Hive
 - HBase
 - Kudu
 - Cloud object stores
- Use Flink for Streaming Analytics
 - Stateful and scalable applications
 - Complex Event Processing
 - SQL Streaming



Kafka and NiFi Use Cases in CDP Data Hub

Hybrid NiFi deployments / Cloud migrations

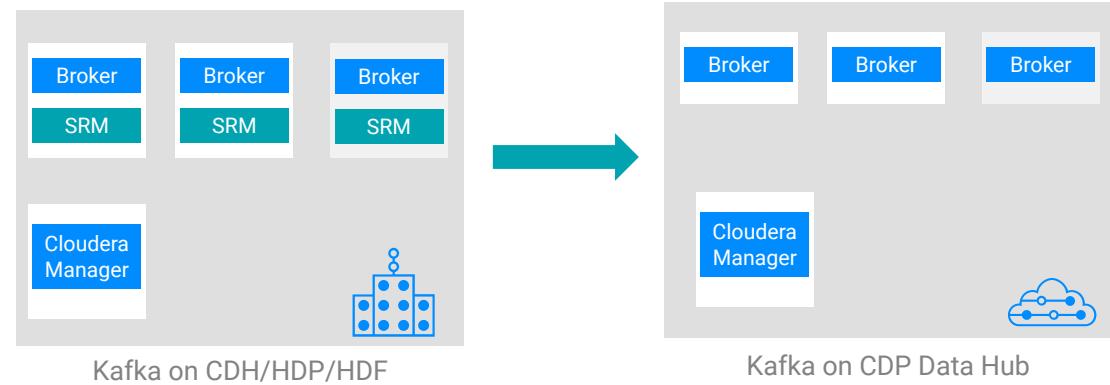
- Set up DataFlows that span on-prem and public cloud environments
- Merge on-prem sources with cloud sources
- Migrate on-prem data sets and make them available for cloud applications



Kafka and NiFi Use Cases in CDP Data Hub

Hybrid Kafka deployments / Cloud migrations

- Set up Kafka in Data Hub as DR cluster
- Migrate existing on-prem Kafka clusters to the cloud
- Leverage Cloudera Streams Replication Manager on on-prem cluster



Cloud Native Streaming Pipelines

Simplifying the User Experience



Cloudera DataFlow Service

Phased Approach for DataFlow Service Delivery

Phase 1 - Flows as a Service

- Shift users' mindsets from thinking about NiFi clusters to Flows
- Focus on Flow Deployment and Monitoring
- Define and assign SLAs to your flows
- Automatic infrastructure sizing and scaling based on workload characteristics and SLAs
- Central monitoring console for all your flows across environments

Phase 2 - Topic as a Service (Kafka) & Stream Apps (e.g: Flink)

- Shift users' mindsets from thinking about Kafka clusters to Topics and end to end data pipelines
- Users provide inputs on the profile, usage characteristics & schema of the topic. DataFlow Service creates the topic on either an existing or new Kafka cluster using K8S operator & scales the clusters to meet the.
- Provide common SDLC experience While Using the Right Compute Engine (e.g: Flink, NiFi) for the Use Case

Flows as a Service

Cluster-Less Data Flow deployment and management

Shifting Users from Thinking about NiFi clusters to Data Flows

Key Capabilities

- Central *DataFlow Catalog* to discover and reuse existing Flow Definitions
- Provide *Flow Parameters*, *Configure Auto-Scaling*, *Define KPIs* in Deployment Wizard
- Monitor and Manage Flow Deployments in a central dashboard

The screenshot displays the Cloudera Data Flow web interface. On the left, a sidebar navigation includes 'Dashboard', 'Catalog' (selected), and 'Environments'. A vertical deployment wizard on the left side shows steps: Overview (done), Parameters (done), Sizing and Scaling (in progress, indicated by a blue dot), KPIs (not started), and Review (not started).

Flow Catalog: A table listing flow definitions. The first entry is 'Kafka-to-Kafka-Route-Filter-Enrich' with version 2 updated 20 days ago. The second entry is 'Machine Data to ML App' with version 1 updated 5 days ago.

Sizing and Scaling: A section titled 'Select the NiFi node size and the number of nodes provisioned for your flow.' It includes 'NIFi Node Sizing' with options for Extra Small, Small (selected), Medium, and Large. Below it is 'Number of NiFi Nodes' with 'Auto Scaling' enabled, showing a slider from 1 to 64 with 'Min. Nodes' at 3 and 'Max. Nodes' at 40.

Dashboard: A real-time monitoring view. It shows two active flows: 'EU Factory to Warehouse' (Status: Good Health) with data rates of 6 MB/s received and 6 MB/s sent over a 30-minute window; and 'Trucking IoT-App-KafkaTo-Kafka-3' (Status: Good Health) with 0 B/s received and 0 B/s sent over a 30-minute window. The dashboard also features a 'Received/Sent Stream Graph' and various filtering and reporting controls.

Topics as a Service

Cluster-Less Topic Creation and Code-less Producers/Consumers

Shifting Users from Thinking about Kafka Clusters to Topics

Key Capabilities

- Users provide **topic profile info including usage characteristics, schema and SLAs**
- DataFlow Service **creates the topic on either an existing or new Kafka cluster using K8S operator & scales the clusters to meet the SLA/KPIS configured.**
- Kafka topic actions include creating **data movement or streaming analytics apps**

The screenshot displays the Cloudera DataFlow service interface across three main panels:

- Create New Topic:** A flow-based wizard with five steps: Topic Overview, Topic Schema, Sizing and Scaling, KPIs, and Review. The "Sizing and Scaling" step is active, showing initial sizing options (X Small, Small, Medium) and auto-scaling configurations (Enabled, Max Nodes: 12, Up to 12 Nodes, Dedicated Cluster, Multi-Zone Availability).
- Topic Overview:** Shows details for the topic "kiosk_events_raw_sfo" in "AWS West". It includes metrics (Schema: KioskEvent), deployment information (Environment: sit-awareness-kiosks-usWest-prod, Node Count: 6, Epoch: 02-24-2020 10:45 AM PST), and actions like SQL Query, Create Flow, and View Kubernetes Cluster.
- Create New Flow Definition:** A modal window for defining data movement flows between various sources and sinks. Quick flows shown include Kafka to S3, Kafka to Hive, Kafka to ADLS, S3 to Kafka, Kafka to Kafka, and ADLS to Kafka. A "Custom" option is also available.

Streaming Apps as a Service

Agile Streaming Analytics App Development using SQL

Shifting Users from Thinking about Flink Clusters to Streaming Apps

Key Capabilities

- Support for **Flink StreamingSQL**
- Select a Kafka Topic and **execute streaming SQL that runs as a continuous query**
- Configure streaming SQL to **write streaming results into Kafka Endpoint**
- **SQL Editor** in Data Flow Service to write, test and deploy streaming SQL jobs

The screenshot shows the Cloudera Data Flow Service interface. At the top, there's a navigation bar with tabs for Metrics, Schema, Consumers, and Producers. A dropdown menu labeled 'Actions' is open, showing options like 'SQL Query', 'Create Flow', and 'View Kubernetes Cluster'. The 'SQL Query' option is highlighted with a yellow box.

The main area displays 'Deployment Information' for a topic named 'kiosk_events_raw_sfo' in 'AWS West'. It shows the schema 'KioskEvent', environment 'sit-awareness-kiosks-usWest-prod', node count (6), and deployment by George Vetticaden. Below this, there's a section for 'Partition Bytes In' showing current usage at 48MB/s and a consumer group section showing current consumption of 24K.

A central modal window titled 'Create New Streaming Application' is open. It has a sidebar with steps: Overview (selected), Configuration, Sizing and Scaling, and SQL, Test and Review. The SQL, Test and Review step is currently active, showing an 'SQL Editor' with the following code:

```

1 SELECT
2   TUMBLE_END(eventTime, INTERVAL '10' MINUTES) as windowEnd,
3   kioskHost,
4   location,
5   COUNT(*) AS numErrors,
6   FIRST_VALUE(event) AS sample
7   FROM
8   kiosk_events_raw_sfo
9   WHERE
10    event LIKE '%ERROR%'

```

Below the editor is a 'SQL Results' table with four columns: windowEnd, KioskHost, Location, and numErrors. The table contains four rows of data. To the right of the table are buttons for 'Stop Test' and 'Deploy' with a cloud icon.

On the right side of the interface, there are two panels: 'Schema: Kiosk Event' which shows the record schema for 'KioskEvent', and 'Application Summary' which provides an overview of the application, configuration, and stream sink details.

Conclusions

Cloudera DataFlow Data-in-motion Platform



What can I use today?

	CDH / HDP / HDF	CDP Private Cloud Base	CDP Public Cloud
NiFi / NiFi Registry	Yes	Yes	Yes
MiNiFi / EFM	Yes	Yes	
Kafka / KStreams	Yes	Yes	Yes
Schema Registry	Yes	Yes	Yes
SMM	Yes	Yes	Yes
SRM	Yes	Yes	Yes
KConnect		Yes	
Flink		Yes	Yes
SQL Stream Builder		Yes (7.1.6+)	
Ranger / Atlas	HDP/HDF only	Yes	Yes

Major organizations leverage CDF

Few examples from Dataworks Summit



Flow &
Streaming use
cases

<https://dataworkssummit.com/san-jose-2018/session/using-nifi-to-simplify-data-flow-streaming-use-cases-mastercard/>



Connected
Plants &
Industrial IoT

<https://dataworkssummit.com/berlin-2018/session/apache-nifi-best-practices-and-lessons-learned/>



Distributed
financial service
with CDF

<https://dataworkssummit.com/berlin-2018/session/synchronicity-of-a-distributed-financial-system/>



Real-time Freight
Visibility

<https://dataworkssummit.com/san-jose-2018/session/real-time-freight-visibility-how-tmw-systems-uses-hdf-to-create-sub-second-transportation-visibility/>

Major organizations leverage CDF

Few examples from Dataworks Summit



Internet of Fleet Management Things

<https://dataworkssummit.com/washington-dc-2019/session/iot-internet-of-fleet-management-things/>



RBC
Royal Bank

Event Standardization Service

<https://dataworkssummit.com/san-jose-2018/session/using-spark-streaming-and-nifi-for-the-next-generation-of-etl-in-the-enterprise/>

ExxonMobil

aetna®

IoT and Time Series data

<https://dataworkssummit.com/san-jose-2018/session/exxonmobil-s-journey-to-unleash-time-series-data-with-open-source-technology/>

High Speed Cybersecurity

<https://dataworkssummit.com/washington-dc-2019/session/building-the-high-speed-cyber-security-data-pipeline-using-apache-nifi/>

TH^{}ON^{} Y^{}U^{}