CLOUDERA

CLOUDERA STREAM PROCESSING (CSP)

Purnima Kuchikulla
Tim Spann
Abdelkrim Hadjidj
Andre Sousa Dantas De Araujo
Daniel Chaffelson

Welcome to our Edge Management Lab

Who are we?

Cloudera Data in Motion Field Team



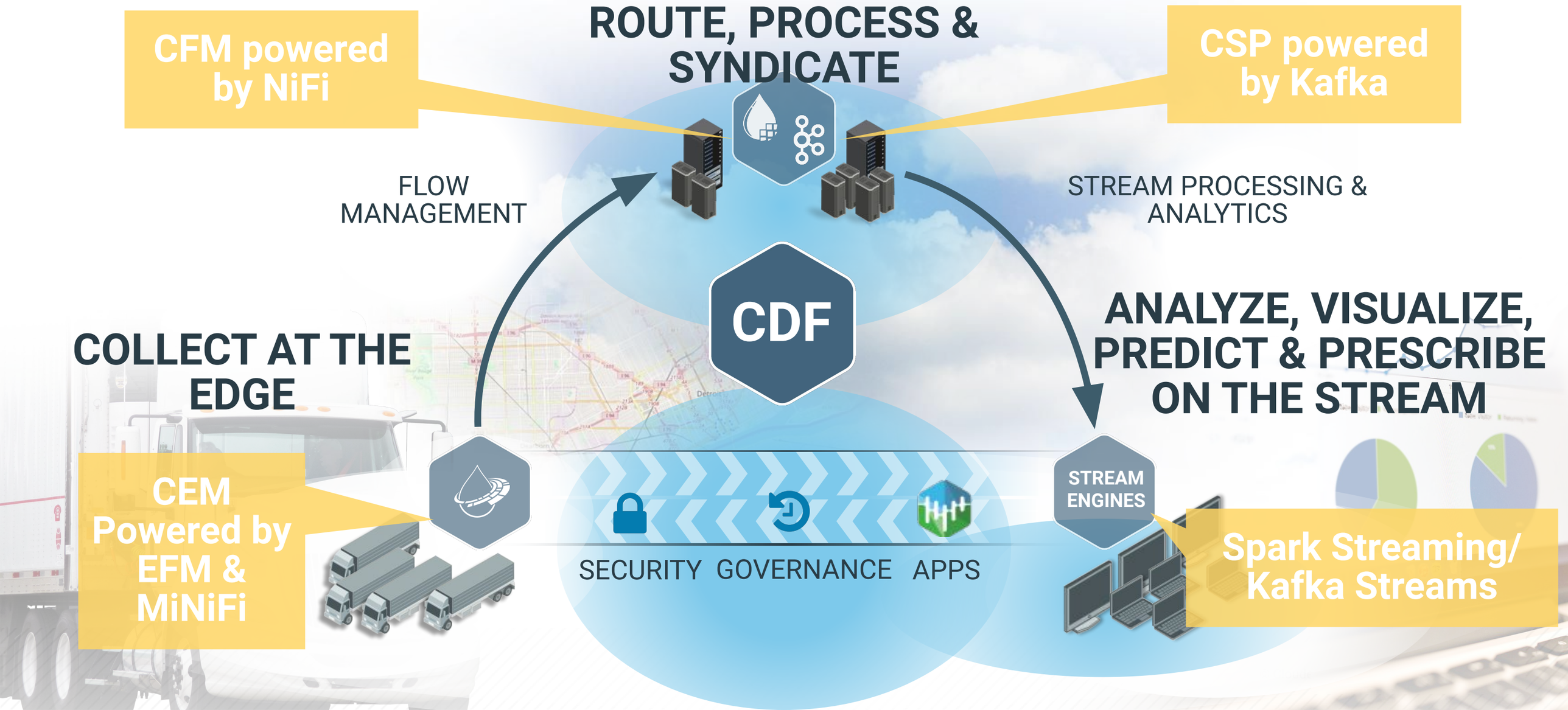


@PaasDev



Kafka

Cloudera DataFlow (CDF)



CLOUDERA DATAFLOW DATA-IN-MOTION PLATFORM

EDGE MANAGEMENT

Edge data collection, routing and monitoring

- MiNiFi
- Edge Flow Manager
- NiFi Registry

FLOW MANAGEMENT

Enterprise data ingestion, transformation and enrichment

- Apache NiFi
- NiFi Registry

STREAM PROCESSING

Real-time stream processing at IoT scale

- Apache Kafka
- Schema Registry

STREAMS MANAGEMENT

- Streams Messaging Manager
- Streams Replication Manager

STREAMING ANALYTICS

Predictive analytics and real-time insights

- Kafka Streams
- Apache Flink
- Spark Streaming

ENTERPRISE SERVICES

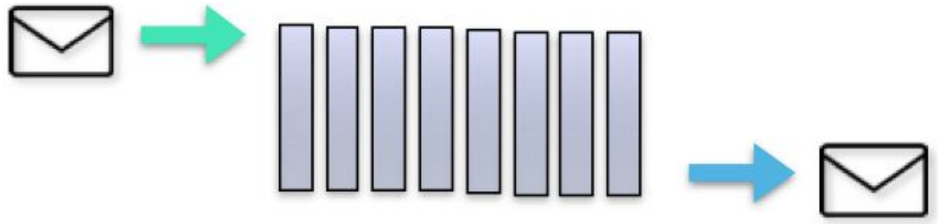
Provisioning, management and monitoring

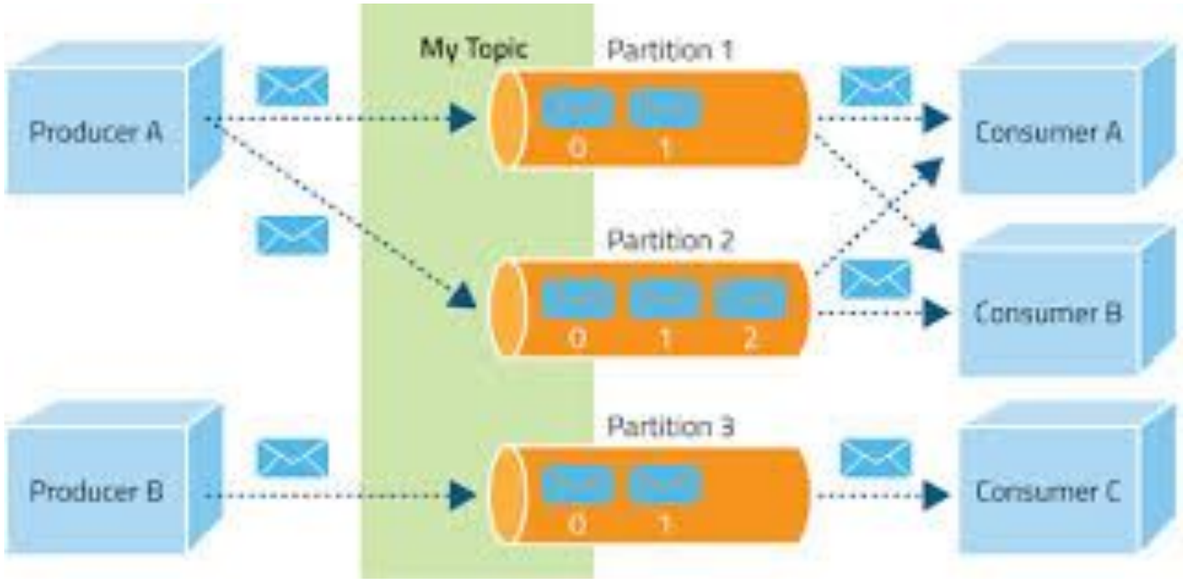
- Unified Security
- Edge-to-Enterprise Governance
- Single Sign-on

What is Kafka

- Who is Kafka?
- What is it
  - A messaging system
  - It can handle lots of messages
  - Messages cannot get lost
  - Message order is preserved
  - Message will get delivered
  - A *producer* sends the message and one or more *consumers* can read the message
- Kafka was built at Linkedin in 2011
- Open sourced as an Apache project

Inside Kafka

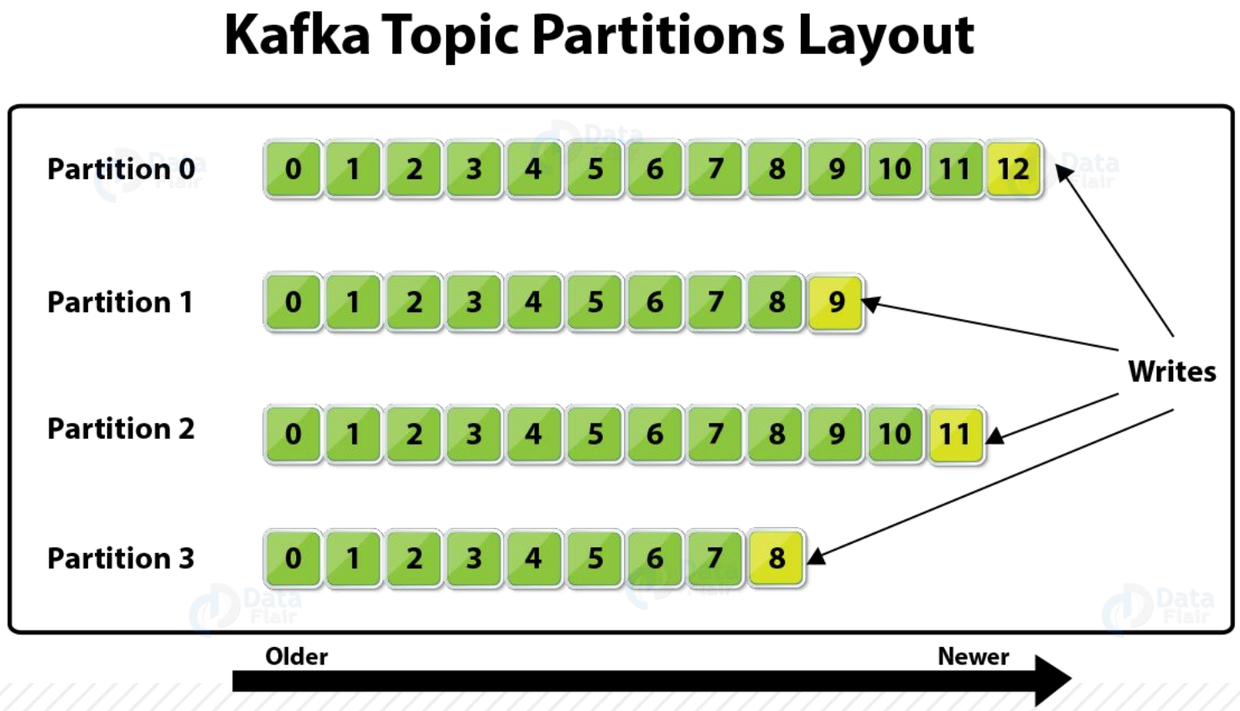




- Kafka topic and topic partition
- Kafka Broker
- Producers
- Consumers

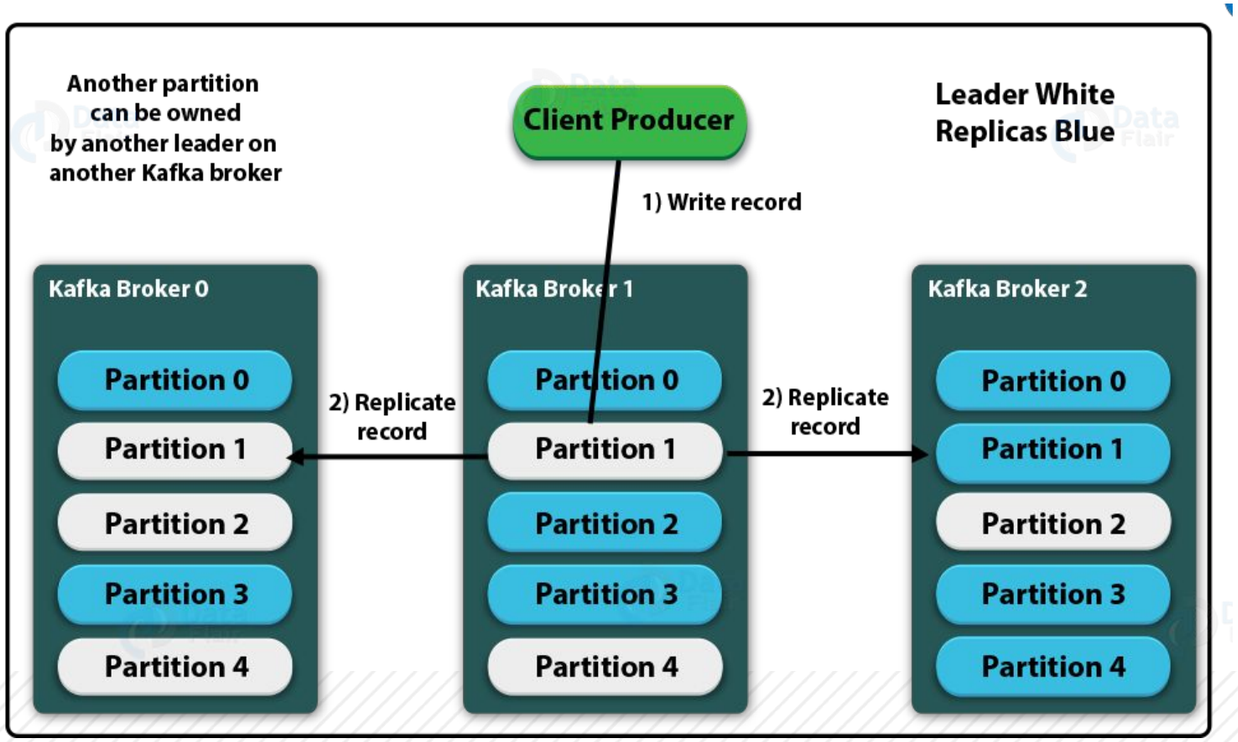
How does Kafka preserve message order

- Partition algorithm is fixed (hash on key)
- Stored as a log sequential write to a file
- Consume in order based on offset

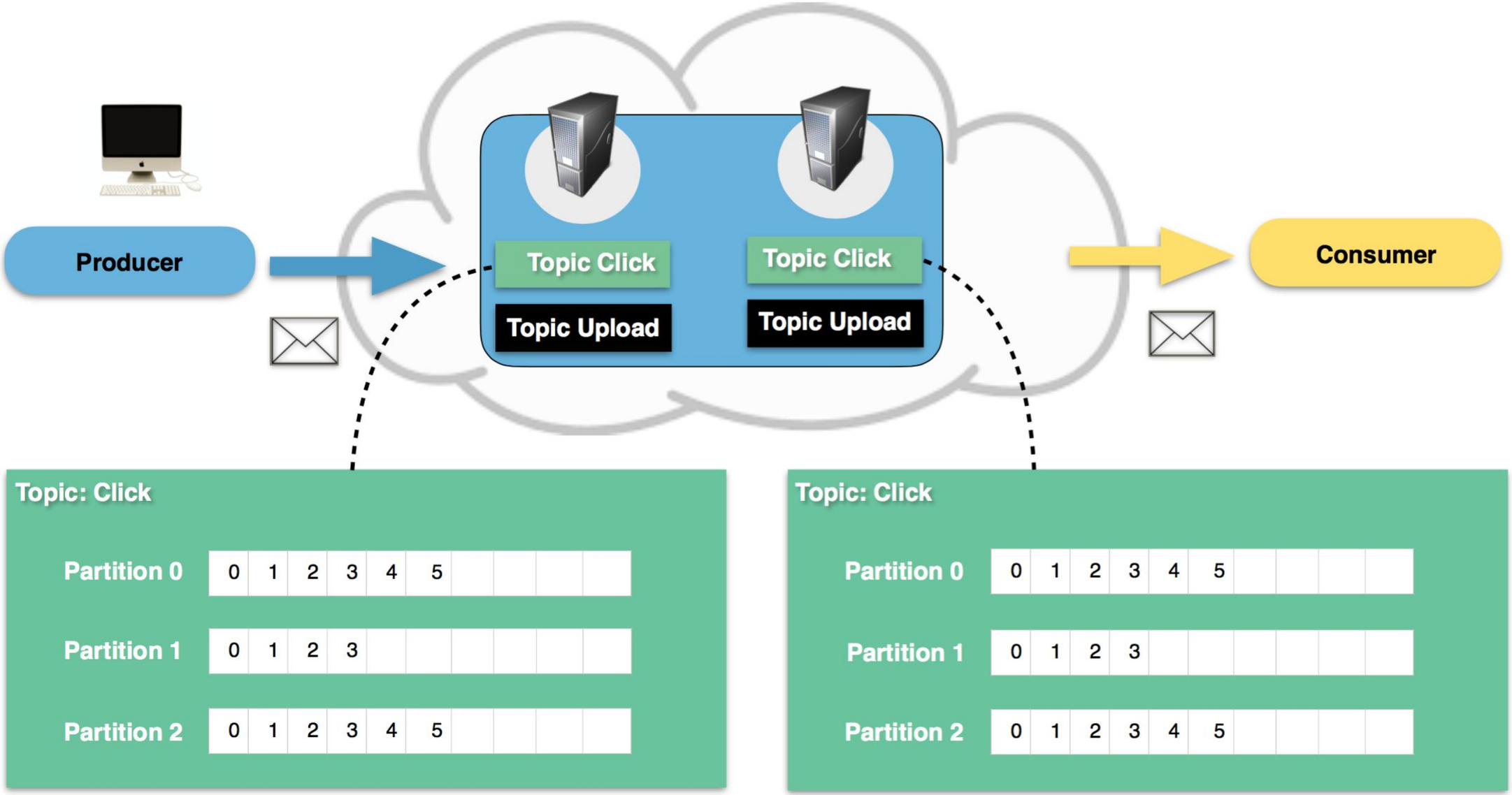


How does Kafka prevent data loss

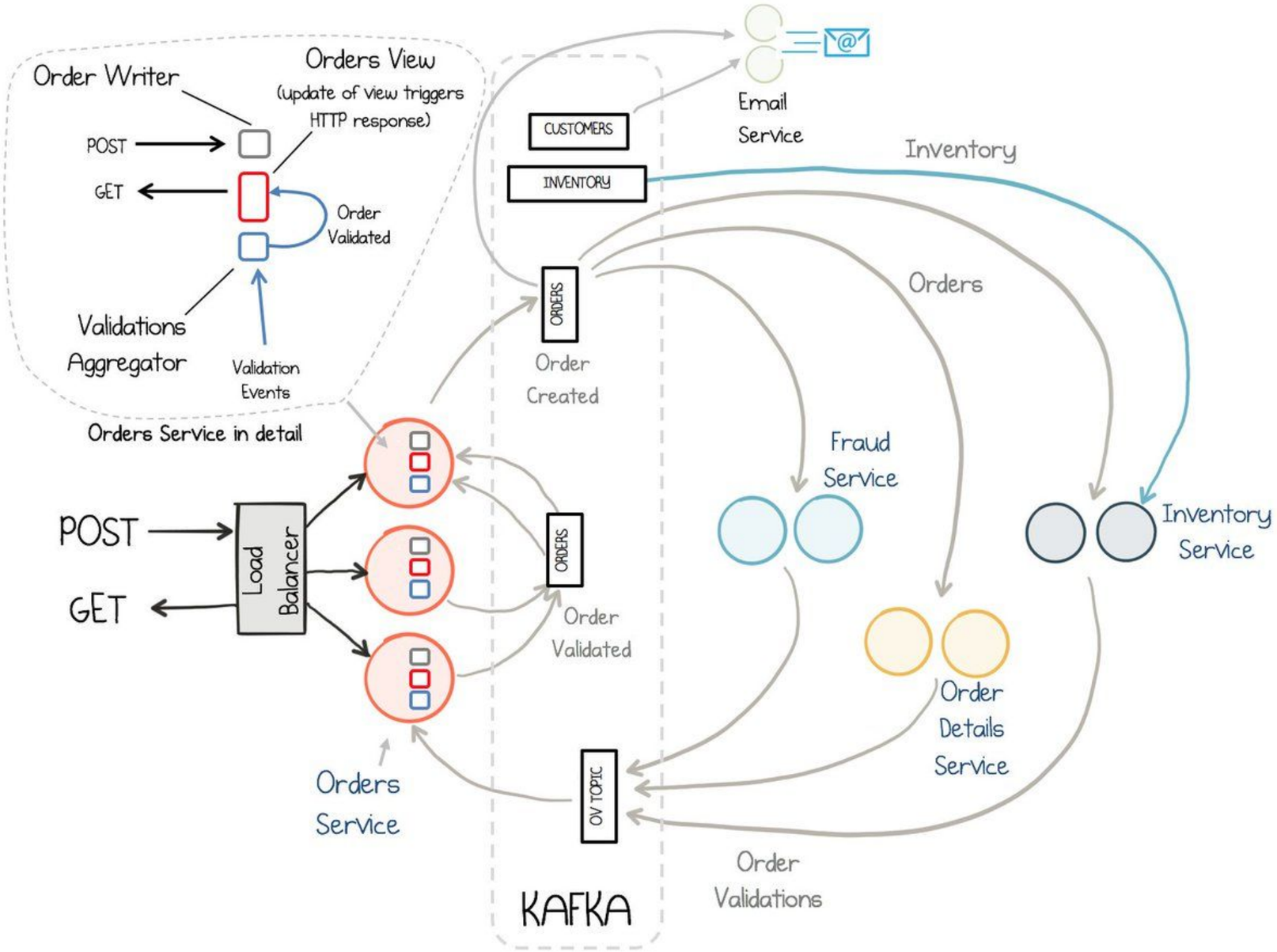
- Replicate, replicate, replicate
- Acknowledge you got the message
- Keep it even after it is consumed



Kafka Use Case: Clickstream Data



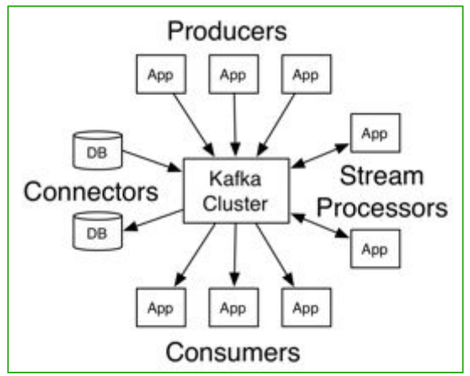
Kafka Use Case: Online ordering system



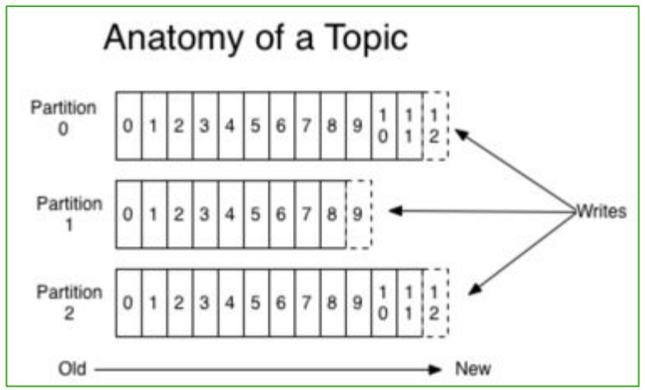
Kafka Basics - Summary

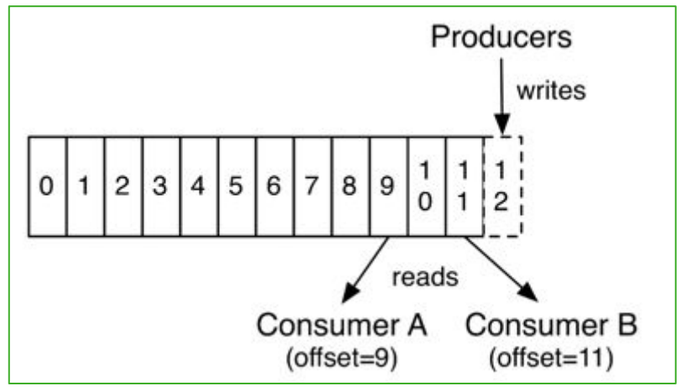
Kafka has 4 core APIs

1. Producer API
2. Consumer API
3. Streams API
4. Connector API

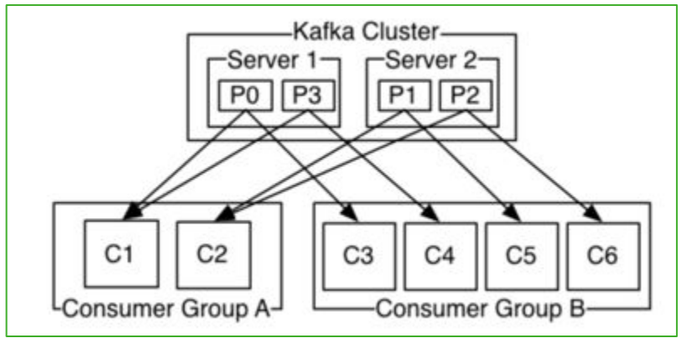


Anatomy of a Kafka Topic





Kafka Consumers



Kafka's Omnipresence Has Led to the Onset of "Kafka Blindness"

- What is **"Kafka Blindness"**?
  - Customers who use Kafka today struggle with monitoring / "seeing"/troubleshooting what is happening in their clusters

- Who is **Affected**?
  - Platform Operation Teams
  - Developers / DevOps Teams
  - Security / Governance Teams

- What are the **Symptoms**?
  - Difficulty seeing who is producing and consuming data
  - Difficulty understanding the flow of data from producers -> topics consumers
  - Difficulty troubleshooting/monitoring.

Streams Messaging Manager (SMM)

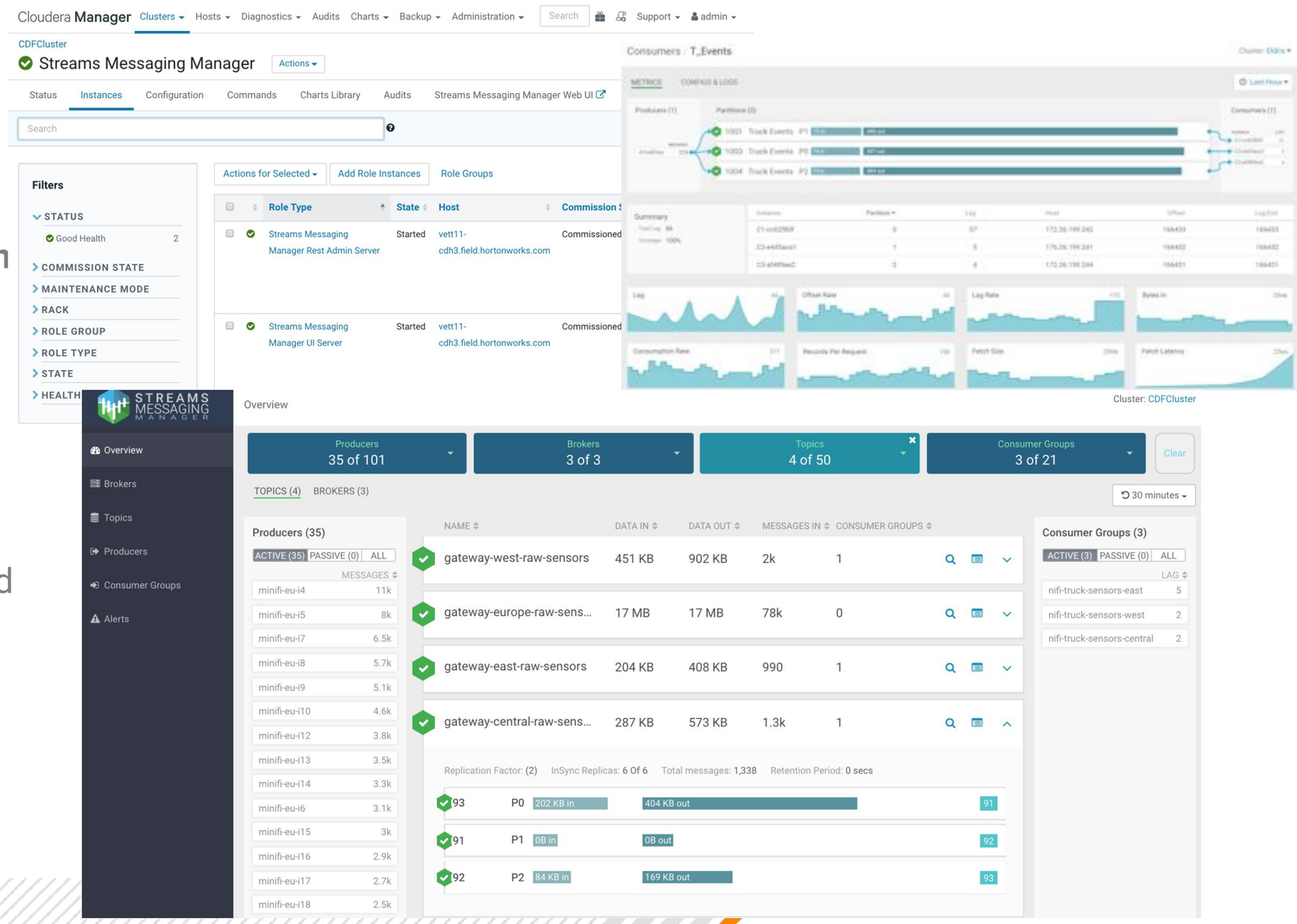
Curing the Kafka Blindness

Problem Statement / Requirements

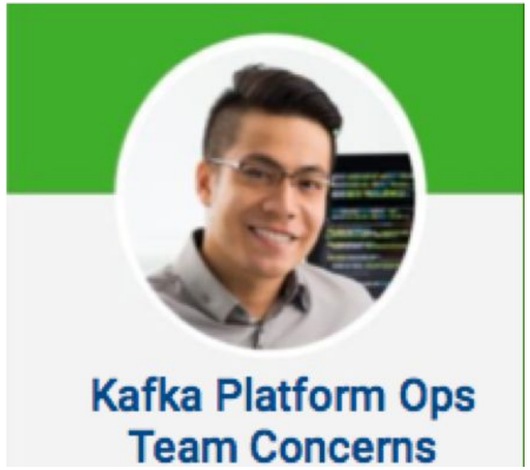
- Customers **struggle with monitoring / "seeing"/ troubleshooting what is happenin in their Kafka clusters**
- They need an enterprise Kafka monitoring solution full integrated with platform services including CM and Sentry

Solution

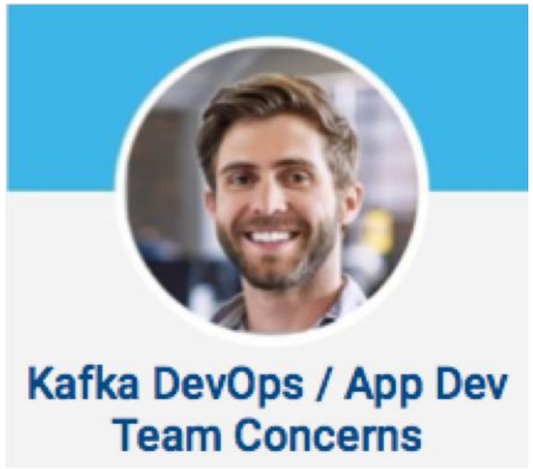
- SMM provides single monitoring Dashboard for Kafka Clusters across 4 entities: Broker, Producer, Topic, Consumer
- SMM integrated with CM Log Search
- Kerberized based authentication and rich access Control via Sentry
- DataPlane Platform NOT required



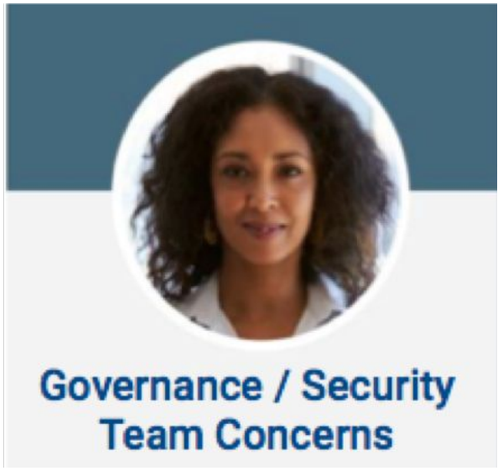
SMM Addresses the Distinct Needs of 3 Personas/Teams



Concerned with monitoring the overall health of the cluster and the infrastructure it runs on

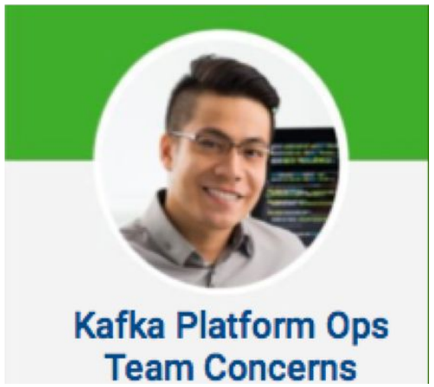


Concerned with monitoring the Kafka entities associated with their apps



Concerned with audit, compliance, access control & chain of custody requirements

SMM for the Platform Ops Team



Do I have any offline topic partitions?

Which consumer group us falling behind the most?

Are any of my brokers down?

Which producers are generating the most data right now?

What is the throughput in/out for a given partition on that broker?

What hosts are my brokers located on?

Are all my replicas in my topic in-synch?

How many total active producers/ consumers is there currently?

Which producers are generating the most data right now?

Are there any skewed partitions for a broker?

How many total active producers and consumers exist now?

Are any of my brokers running hot? Which broker has the highest throughput in and and rates?

How many total topics does my Kafka cluster have?

Which producers are generating the most data right now?

Are any of my brokers running out of disk space?

How of the cluster is being used, how much capacity do I have available per broker / cluster?

Which partitions are located on each broker?

Which of my topics has produced/consumers the most messages over the last N minutes/hours?

SMM for the DevOps/AppDev Teams

Kafka DevOps / App Dev Team Concerns

Find all entities (producer, consumers, topics) associated with my app.

What is the retention rate for my app topics?

What is the replication factor for my app topics?

What brokers holds the partitions for my app topics?

Who are all the producers and consumers connected to my app topics?

What type of events are in my application topics? What does the event look like?

Did a consumer rebalance occur for a given topic?

What is the total number of messages into my topic over the last N minutes/hours?

Are any of my consumers/ consumer-groups that are over-consuming?

Are there consumers in a consumer-group for a given topic slow/falling behind?

How many active consumers instances are in a given consumer group?

Are any of my consumers/consumer-groups that are under-consuming?

What topic(s) are the consumer group consuming messages from?

SMM for the Security and Governance Team

When was a topic created?

What is the schema for a given topic?

What is the linage of a kafka topic?

How has the schema evolved for a given topic?

How does data flow across multiple kafka hops?

Who has edited the ACL policies of a given topic?

Which consumers have consumed from a topic?

What are the ACL policies for a given topic?

When were additional brokers added to the topic?

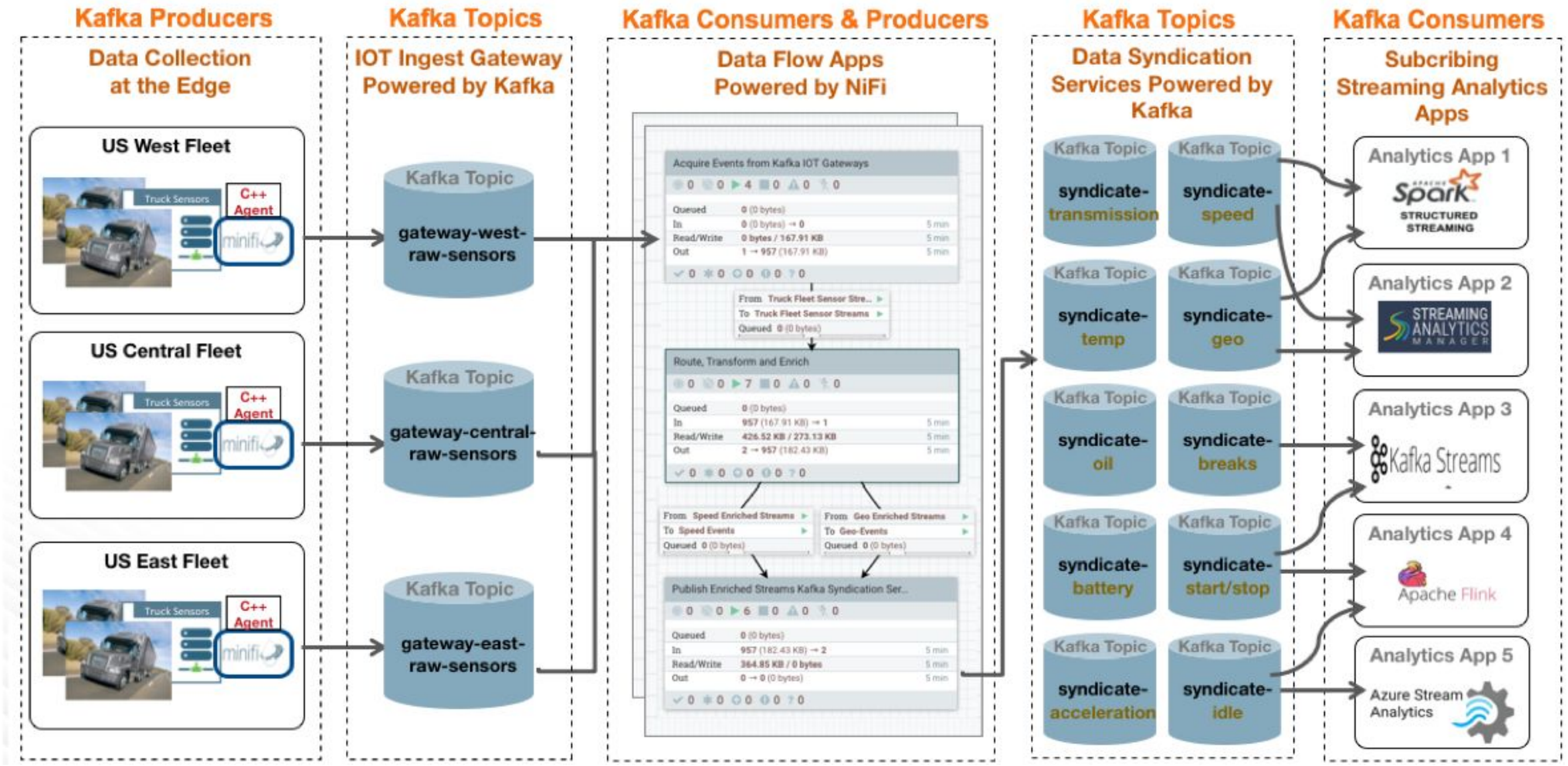
Which users/groups/service accounts have read from a given topic?

Which users/groups/service accounts have sent data to a topic?

When was the topic configuration last modified?

Which producers have sent data to the topic?

Demo Setup: Dev Ops / App Dev Persona – Monitoring the Streaming Truck App



Kafka Challenges

Shared Schema Registry

How do I associate schemas for messages in Kafka topics

Agility and Self-Service

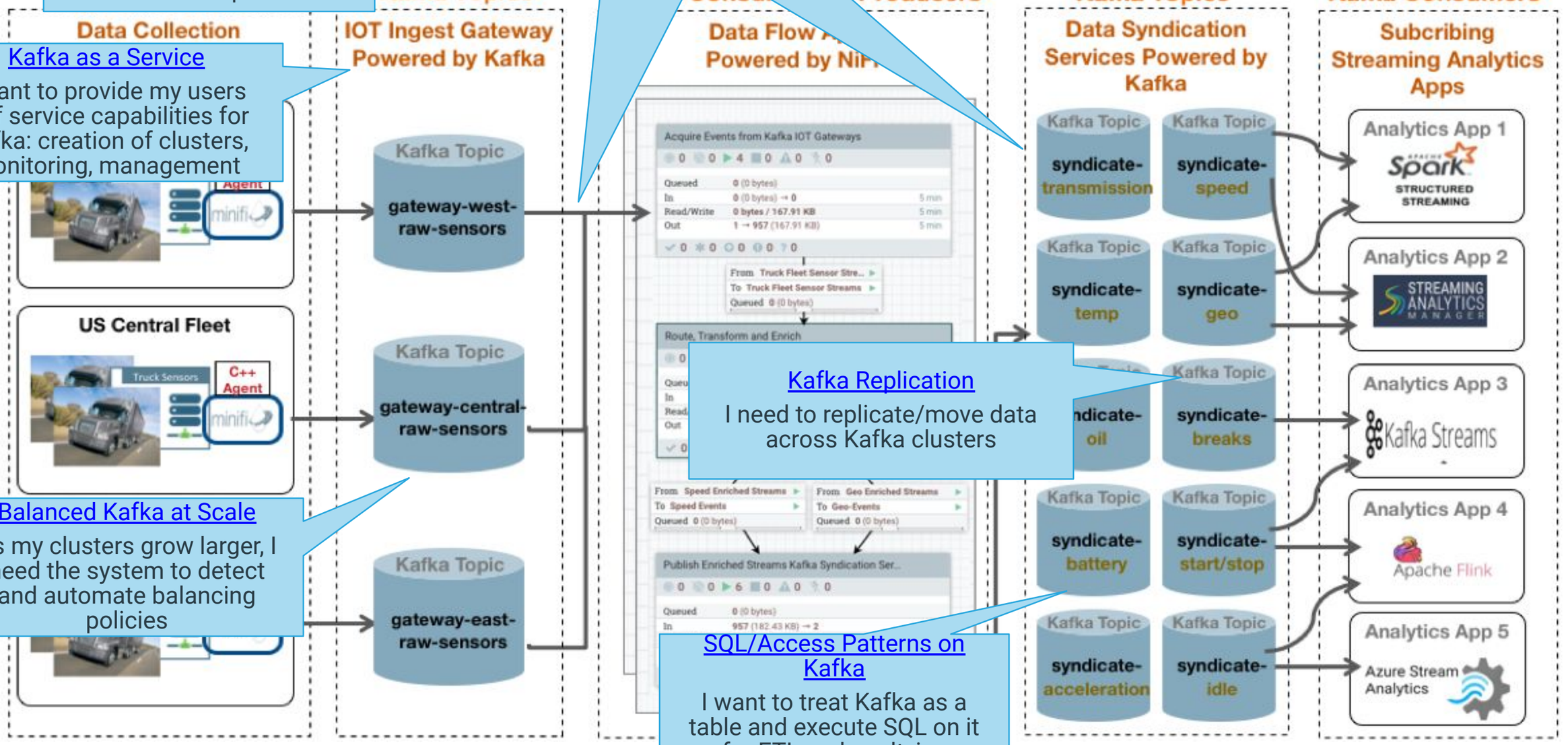
How do I develop, deploy & manage Kafka producers and consumers without code in self service manner

Kafka Blindness

How do I manage/monitor the different kafka clusters, topics, consumers, and producers

Kafka as a Service

I want to provide my users self service capabilities for Kafka: creation of clusters, monitoring, management



Kafka Replication

I need to replicate/move data across Kafka clusters

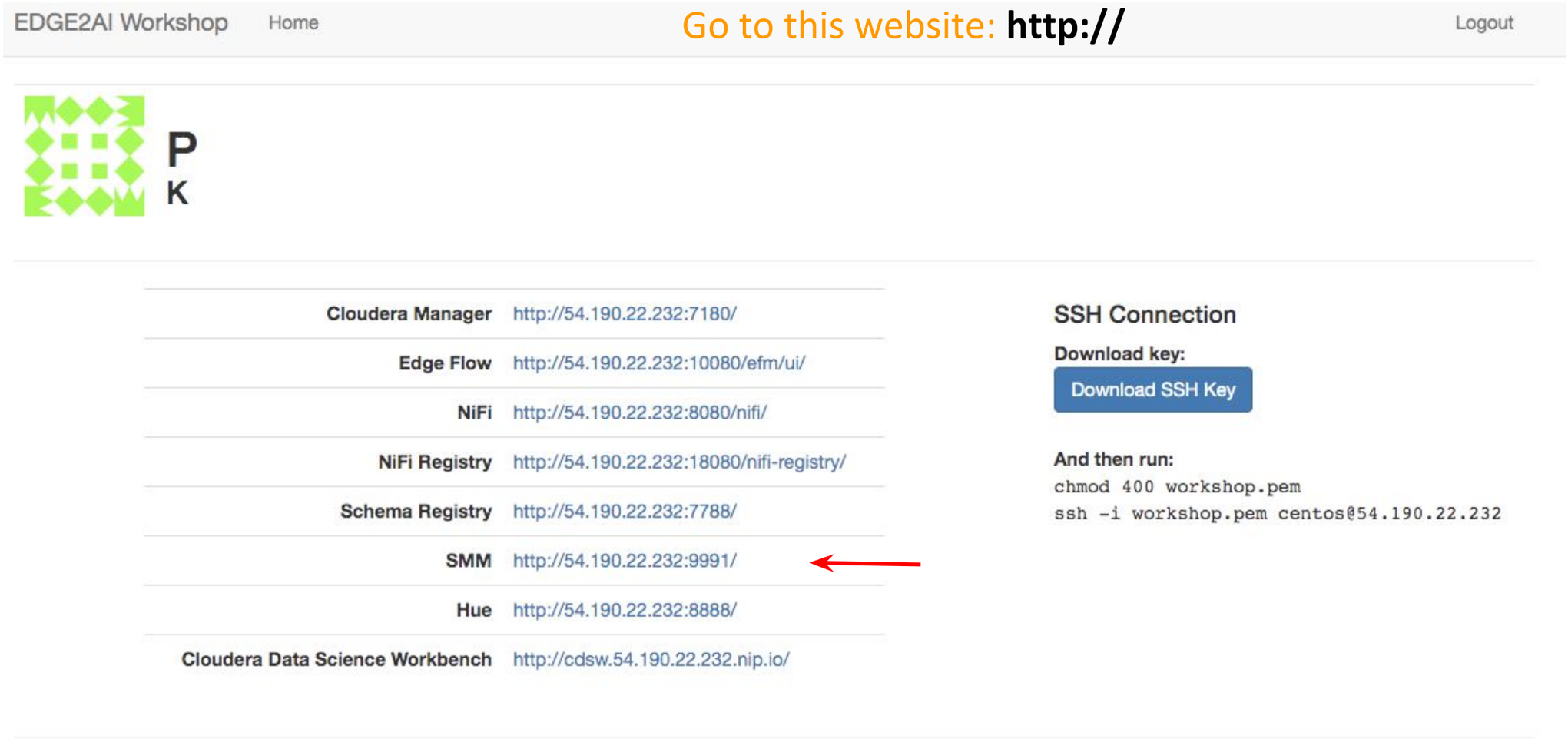
Balanced Kafka at Scale

As my clusters grow larger, I need the system to detect and automate balancing policies

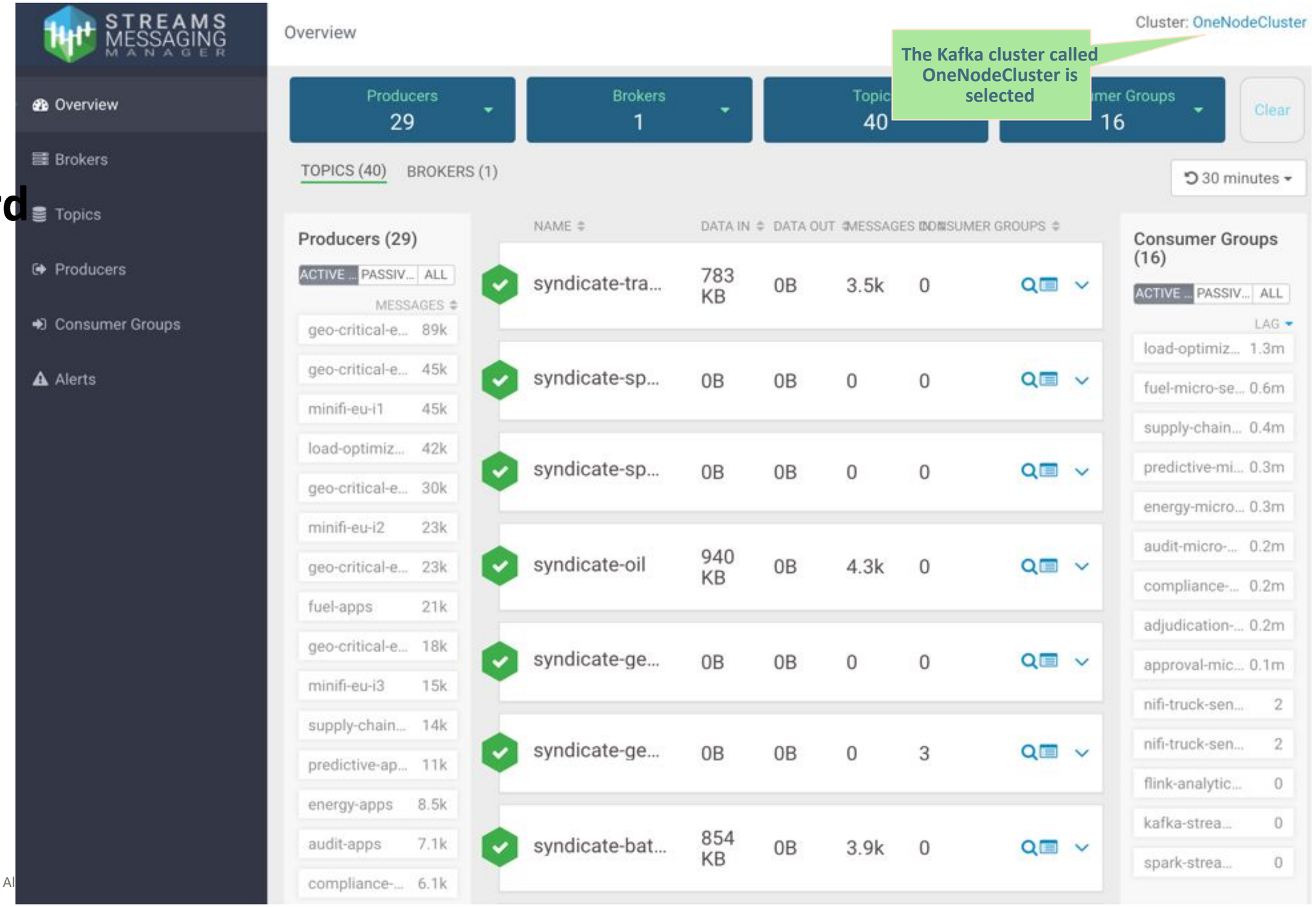
SQL/Access Patterns on Kafka

I want to treat Kafka as a table and execute SQL on it for ETL and analtyics

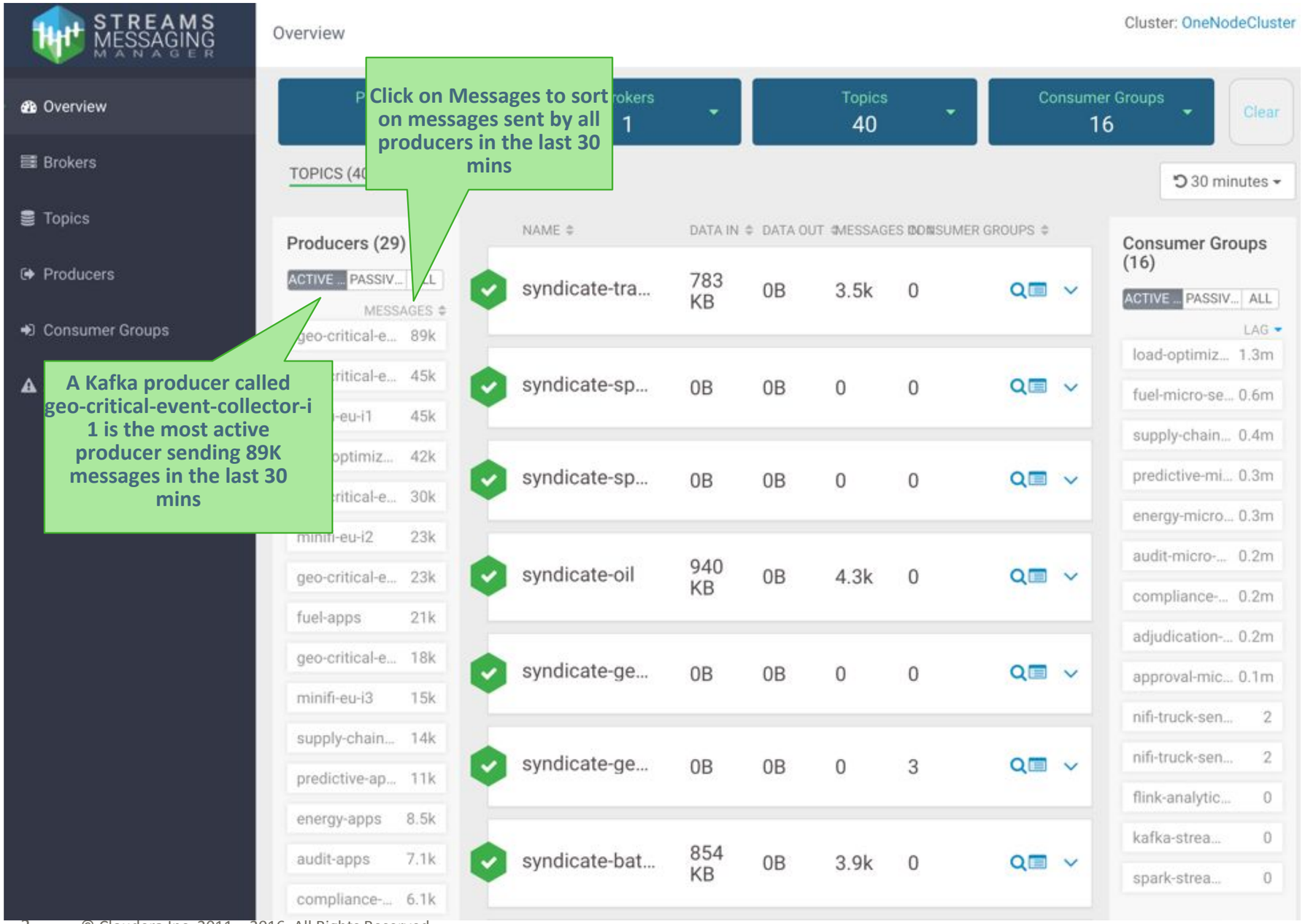
Tutorial - Log In Instructions



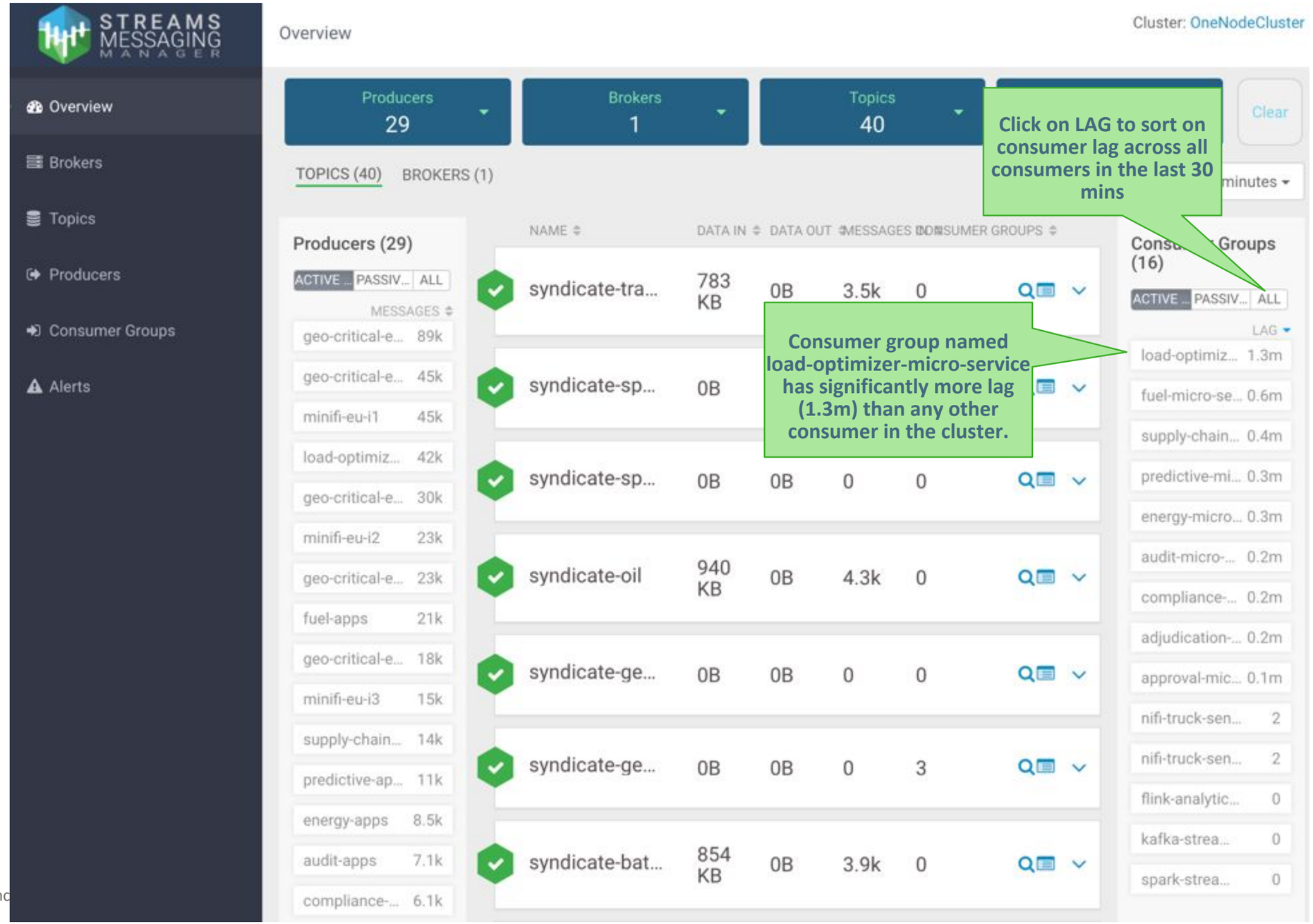
Main Dashboard View



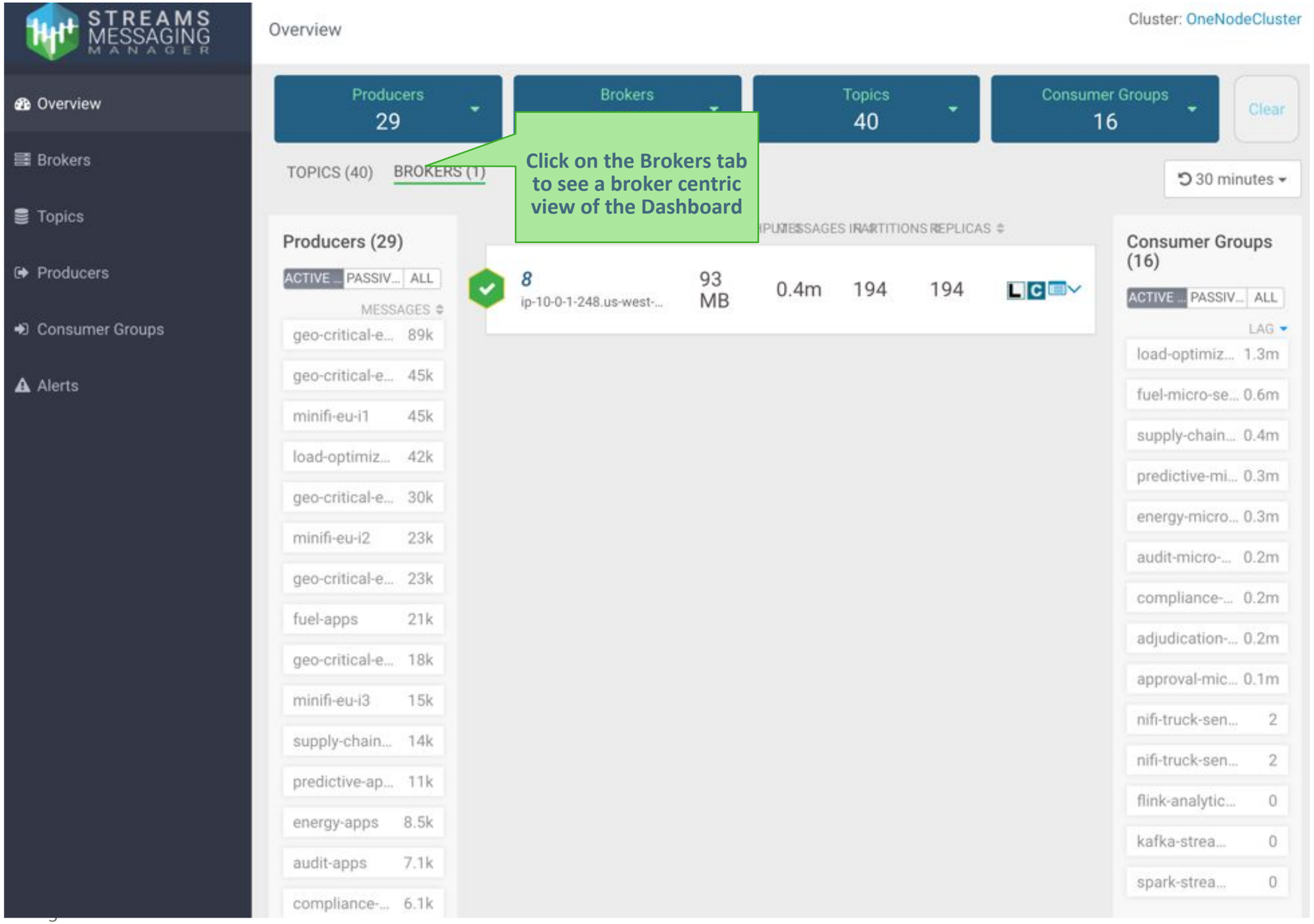
Find the Most Active Producer in my Cluster



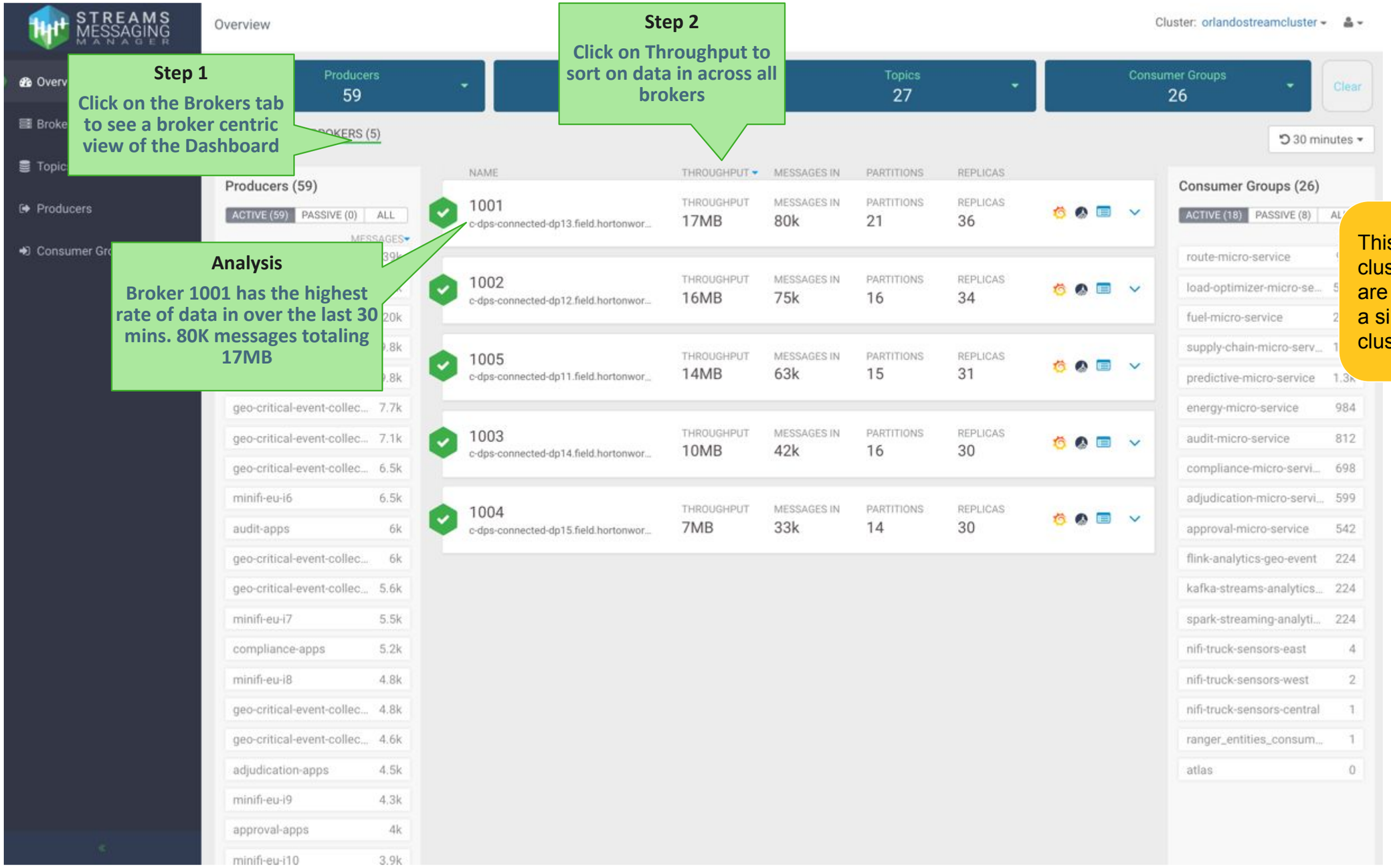
Find the Consumer Who Has Fallen Behind the Most



Broker Centric View – View Details of the Brokers in My Cluster

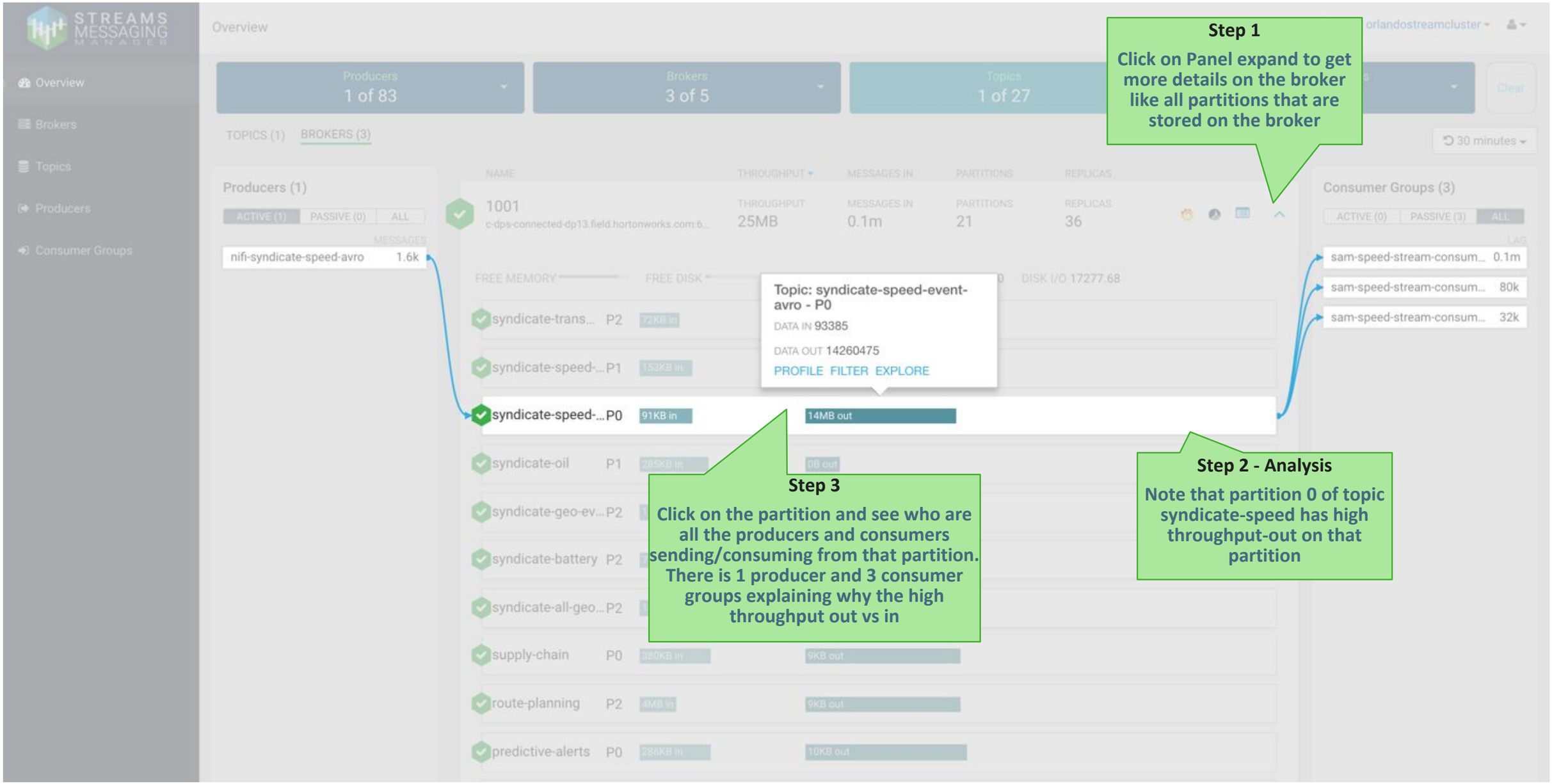


Broker Centric View: Find my Hottest Broker – Broker with Highest Throughput In

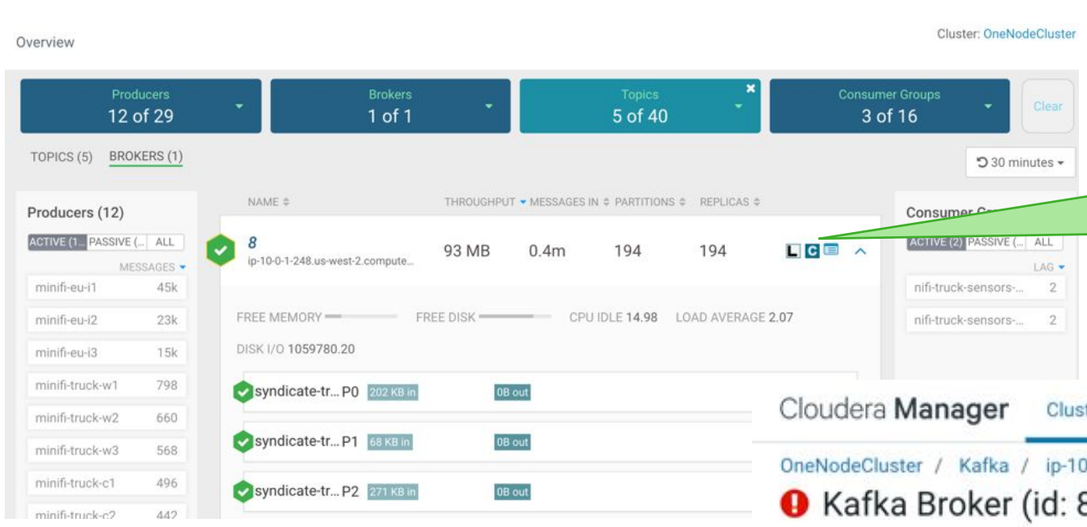


This is a multi-node cluster. What you are working with is a single node cluster.

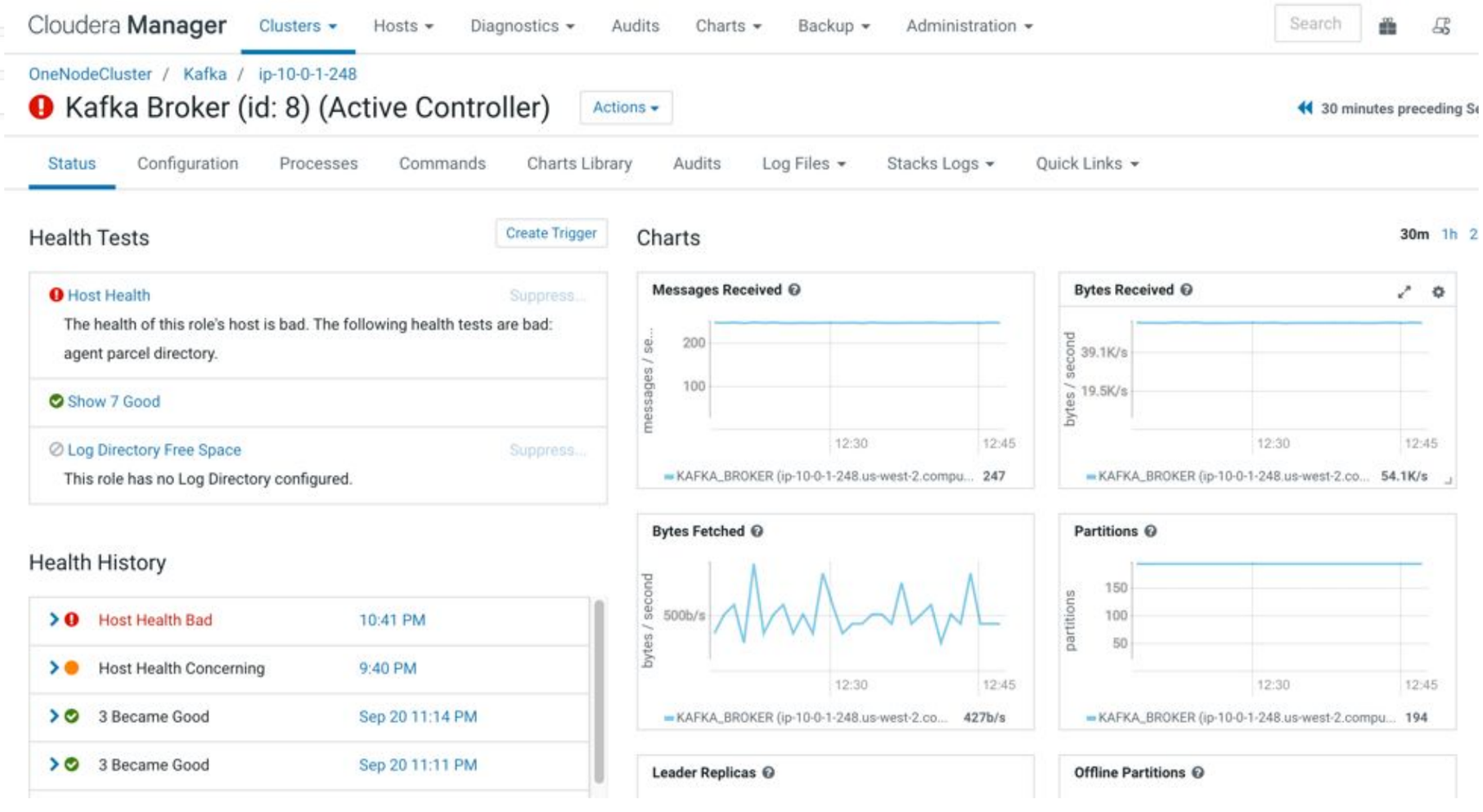
Find Partitions on a given Broker and Understand flow of data flow from Producer to selected Broker Partition to Consumer



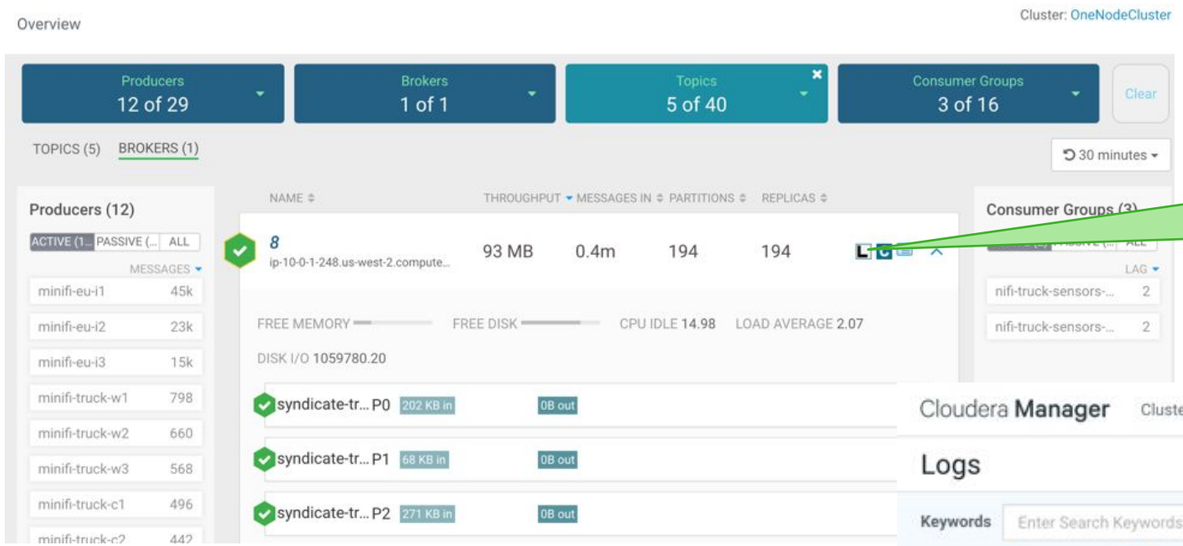
Analyze Detailed Broker Host Metrics – Cloudera Manager Integration



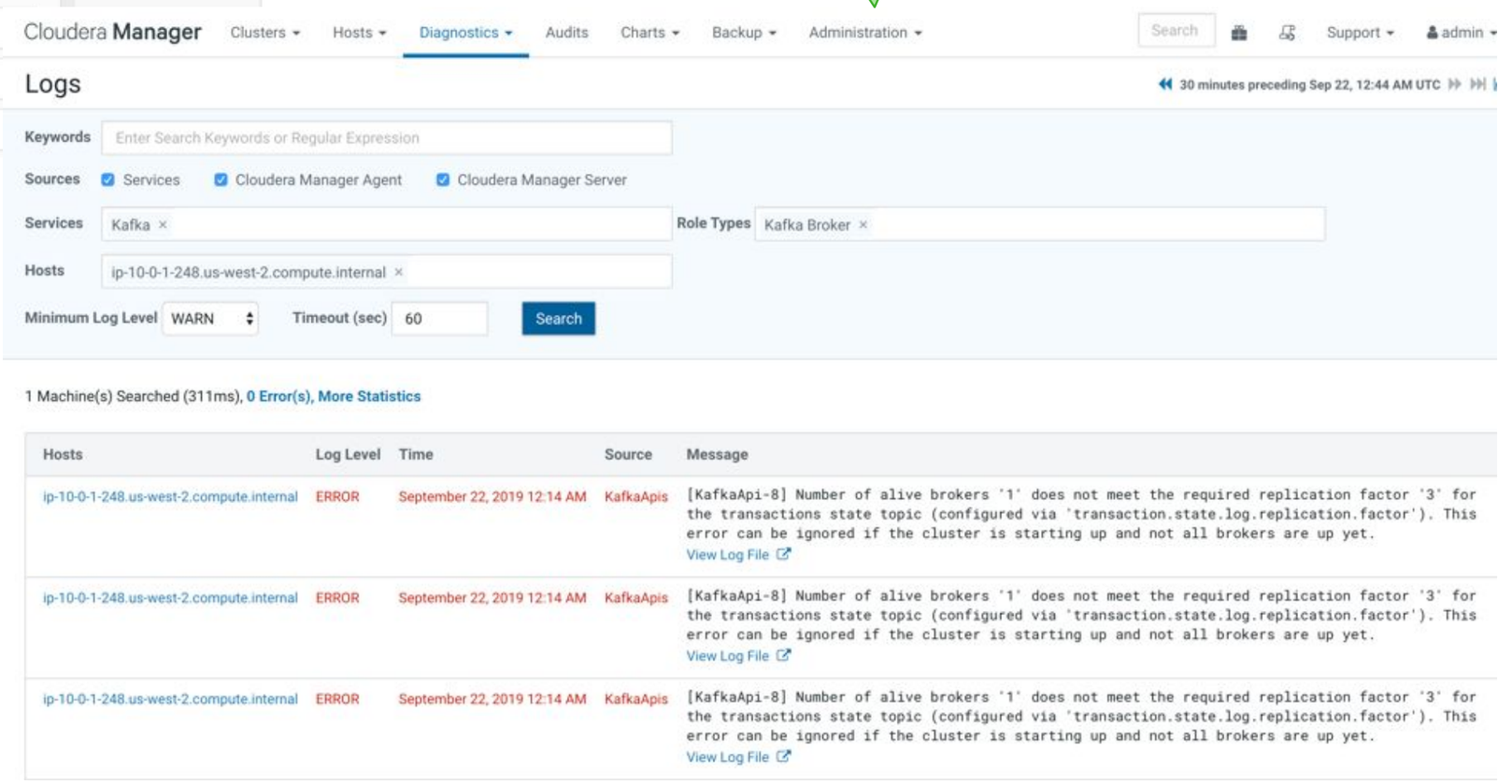
Click on the CM icon on the broker panel and the CM host detail view for that broker is displayed providing host level metrics and a view of other services running on that host



Keyword Search via Log Search

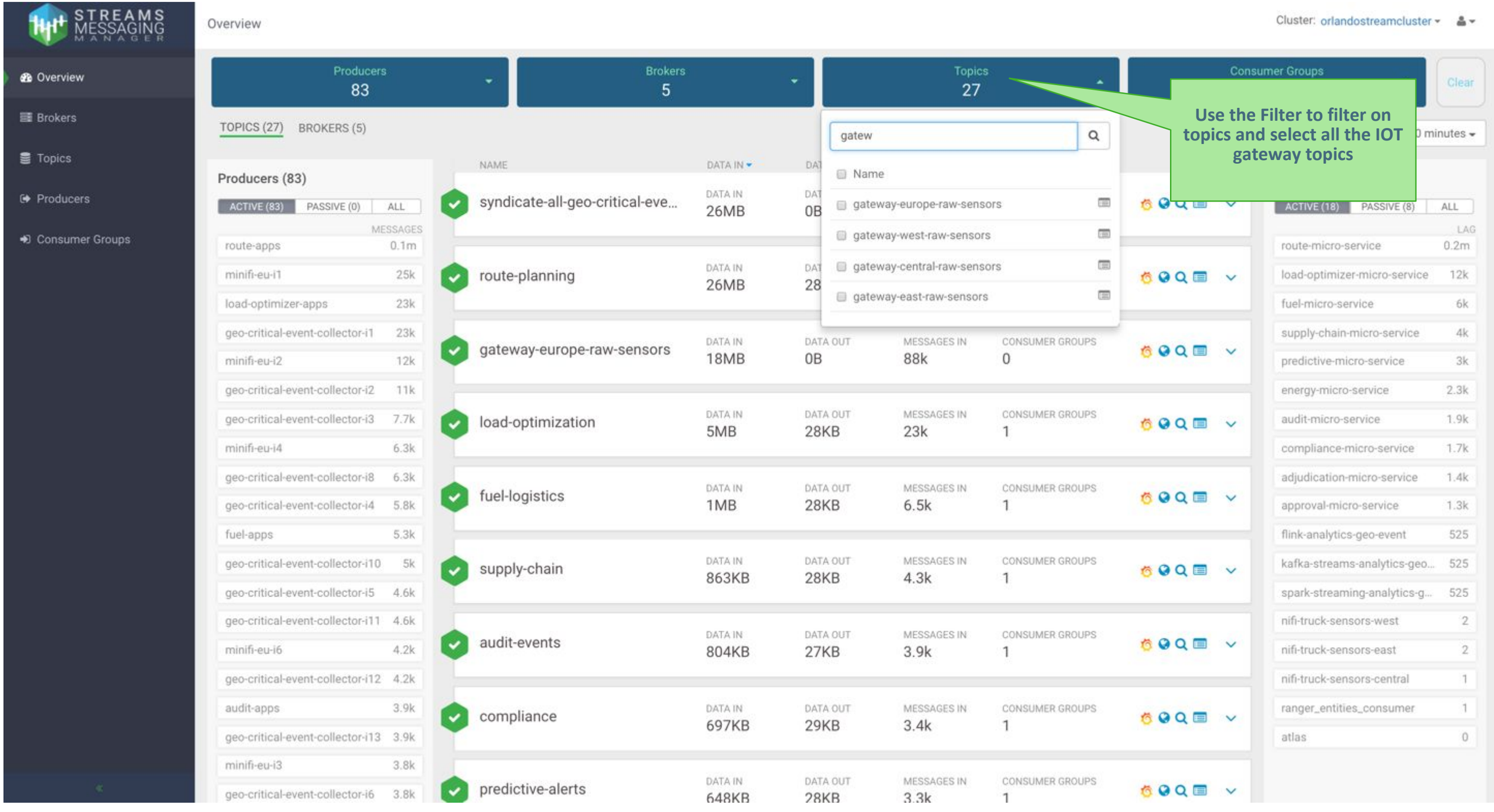


Click on the Log Search icon on the broker panel and the Log Search detail view is displayed. This enables you to search for specific keywords and to filter for specific log levels, components, and time ranges.



SMM DevOps/App Dev Use Cases

Topic Centric Dashboard View: Filter on Topics associated with my Topic

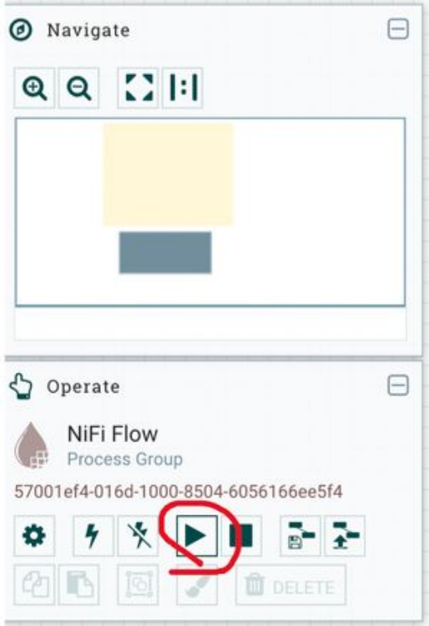


EDGE2AI Workshop Home Users Clusters

pk-strata

Start NiFi Flow

Cloudera Manager	http://54.185.190.193:7180/
Edge Flow	http://54.185.190.193:10080/efm/ui/
NiFi	http://54.185.190.193:8080/nifi/
NiFi Registry	http://54.185.190.193:18080/nifi-registry/
Schema Registry	http://54.185.190.193:7788/
SMM	http://54.185.190.193:9991/
Hue	http://54.185.190.193:8888/
Cloudera Data Science Workbench	http://cdsw.54.185.190.193.nip.io/

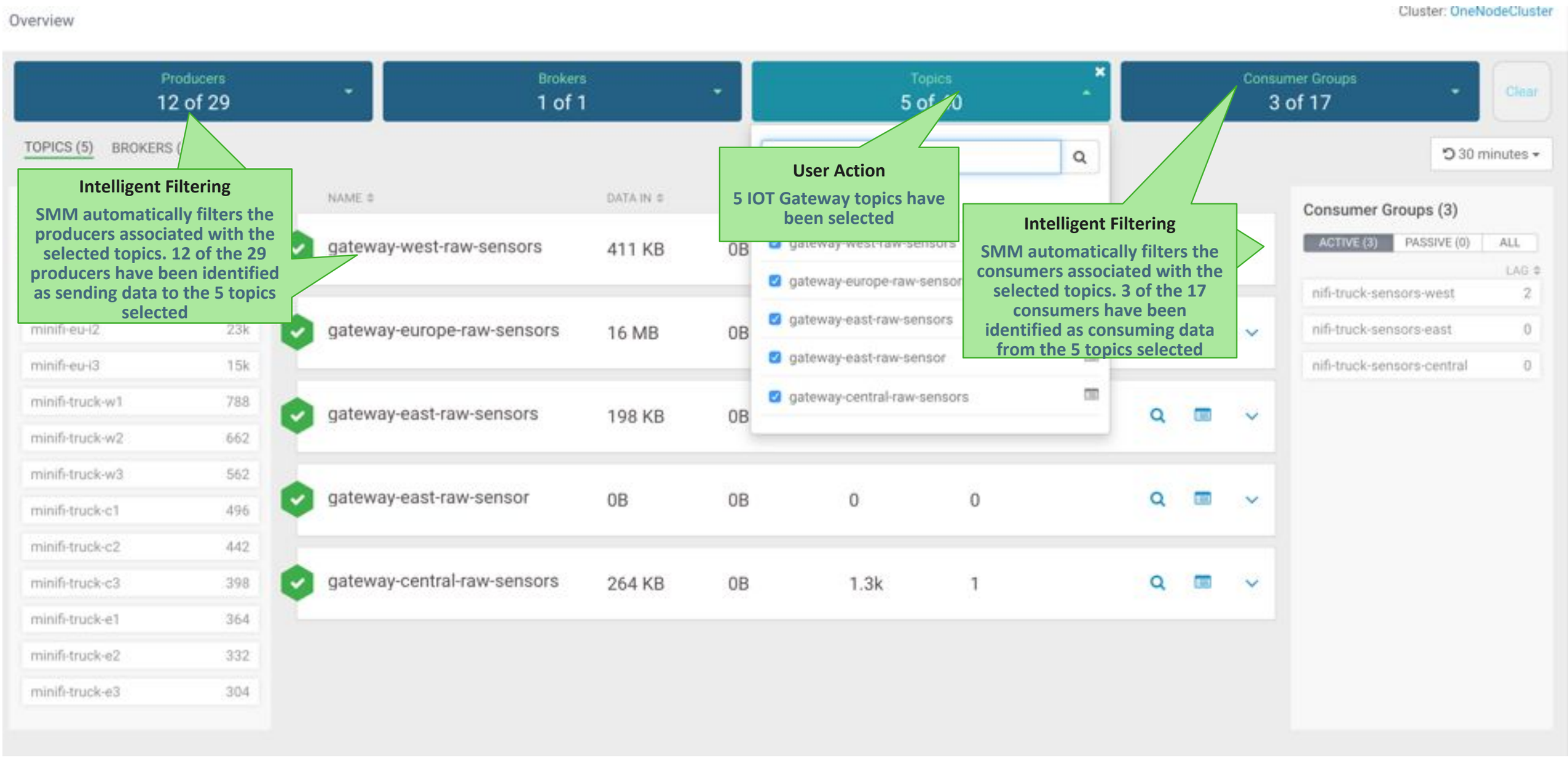


1. Nifi ingest Trucking CSV Events with kafka headers that contains schema name which is a pointer to schema in HWX Schema Registry..

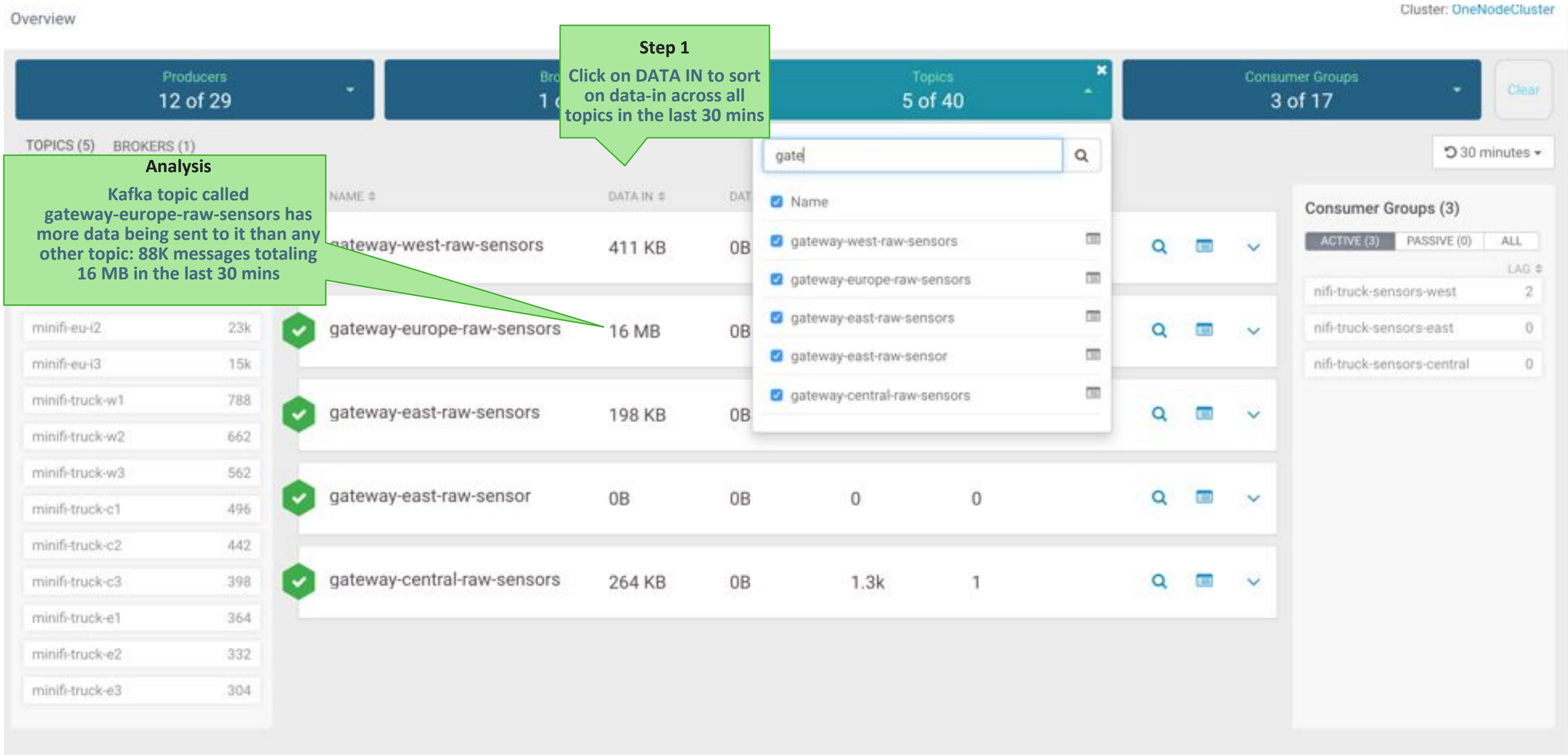
2. Nifi extracts the schema name from kafka header and fetches schema from the HWX Schema Registry to perform record based processing including filtering, routing and enrichment

3. These CSV records are then enriched and converted into Avro Records and send to a Kafka Topic.
When publishing to Kafka Topic, Nifi will look up the schema associated with the kafka topic in HWX SR and encode the Avro binary with HWX schema encoding so that SAM can work with the data.

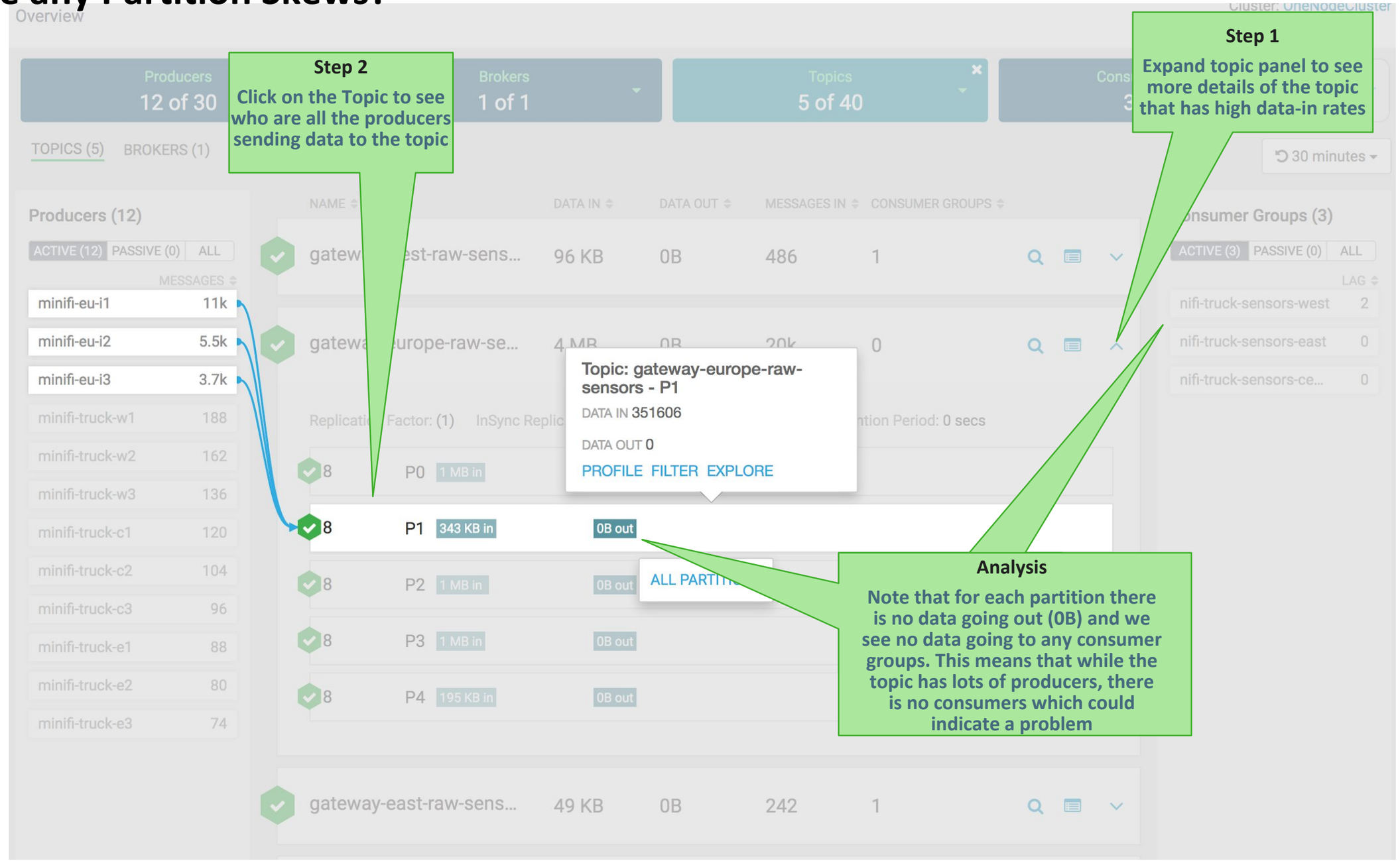
Intelligent Filtering – Selected Topics causes Producers / Consumer to be Intelligently Filtered



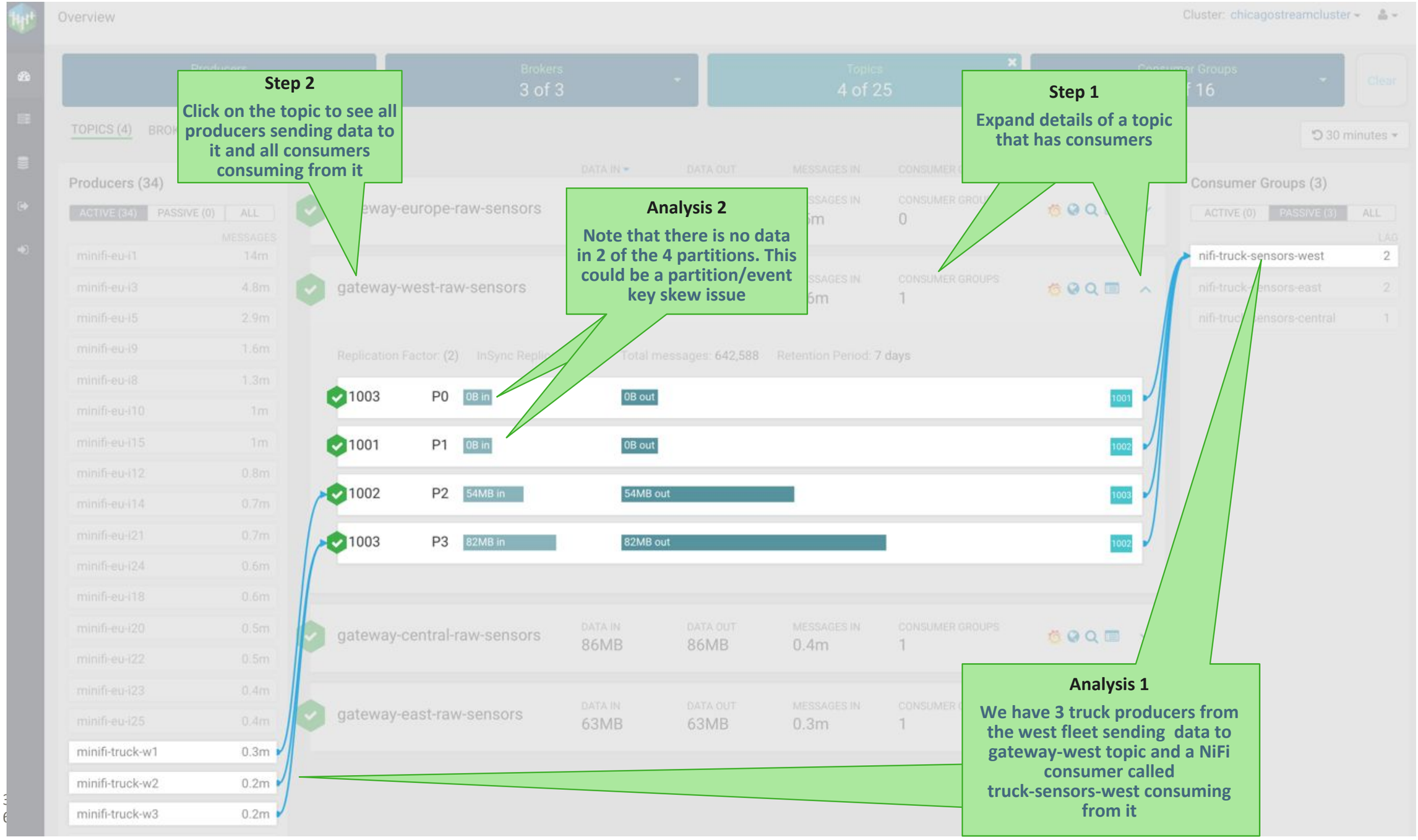
Find the Hottest Topic – Topic With Highest Throughput-In



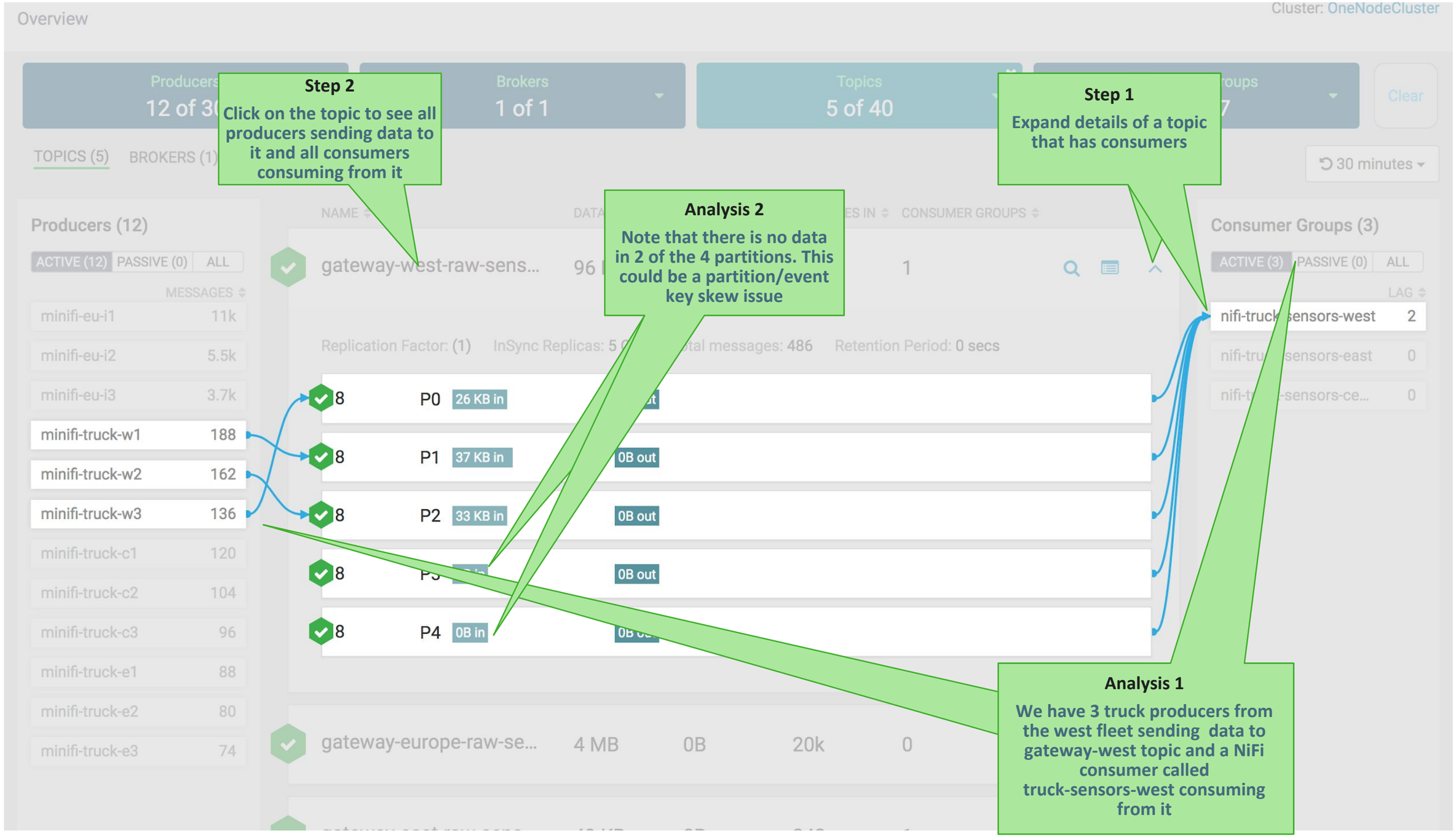
How are the Partitions Laid out for the Topic? Who are the Producers and Consumers? Are there any Partition Skews?



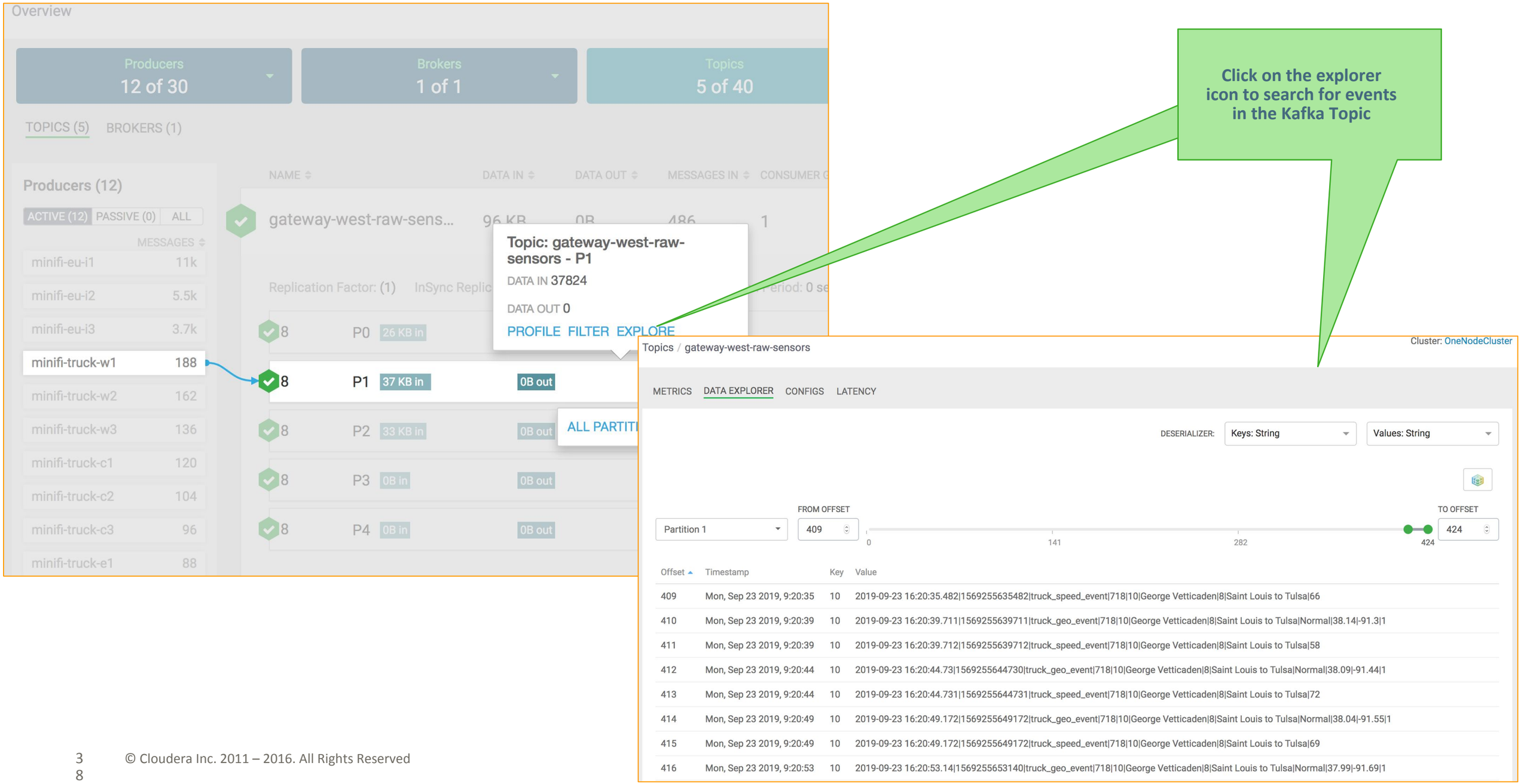
How does Data Flow between Producers to Topics to Consumers?



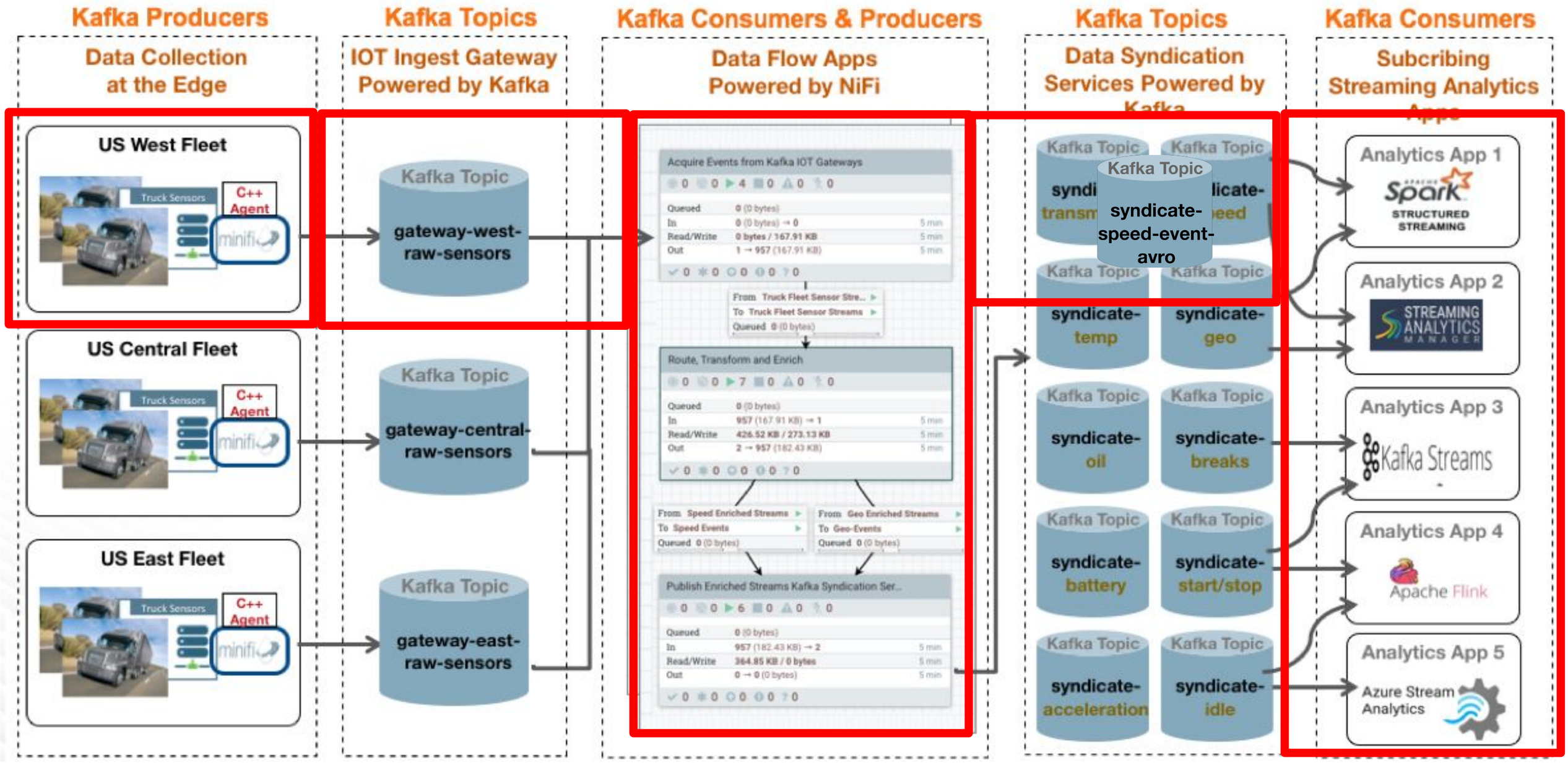
How does Data Flow between Producers to Topics to Consumers?



Explore/Search Messages in the Kafka Topic



Recap: What Did We Just see? Tracking the flow of data across multiple Kafka Hops with SMM & Atlas Integration-Powerful



SMM 1.2 New Features

New Features

Topic Lifecycle Management

- Create
- Update
- Delete

Alerting

- Alert Notifier
- Alert Policy

Schema Registry Integration

- Data Governance
  - Provide reusable schema (centralized registry)
  - Define relationship between schemas (version management)
  - Enable generic format conversion, and generic routing (schema validation)
- Operational Efficiency
  - To avoid attaching schema to every piece of data (centralized registry
  - Consumers and producers can evolve at different rates (version management)
  - Data quality (schema validation)

LESS OF THIS

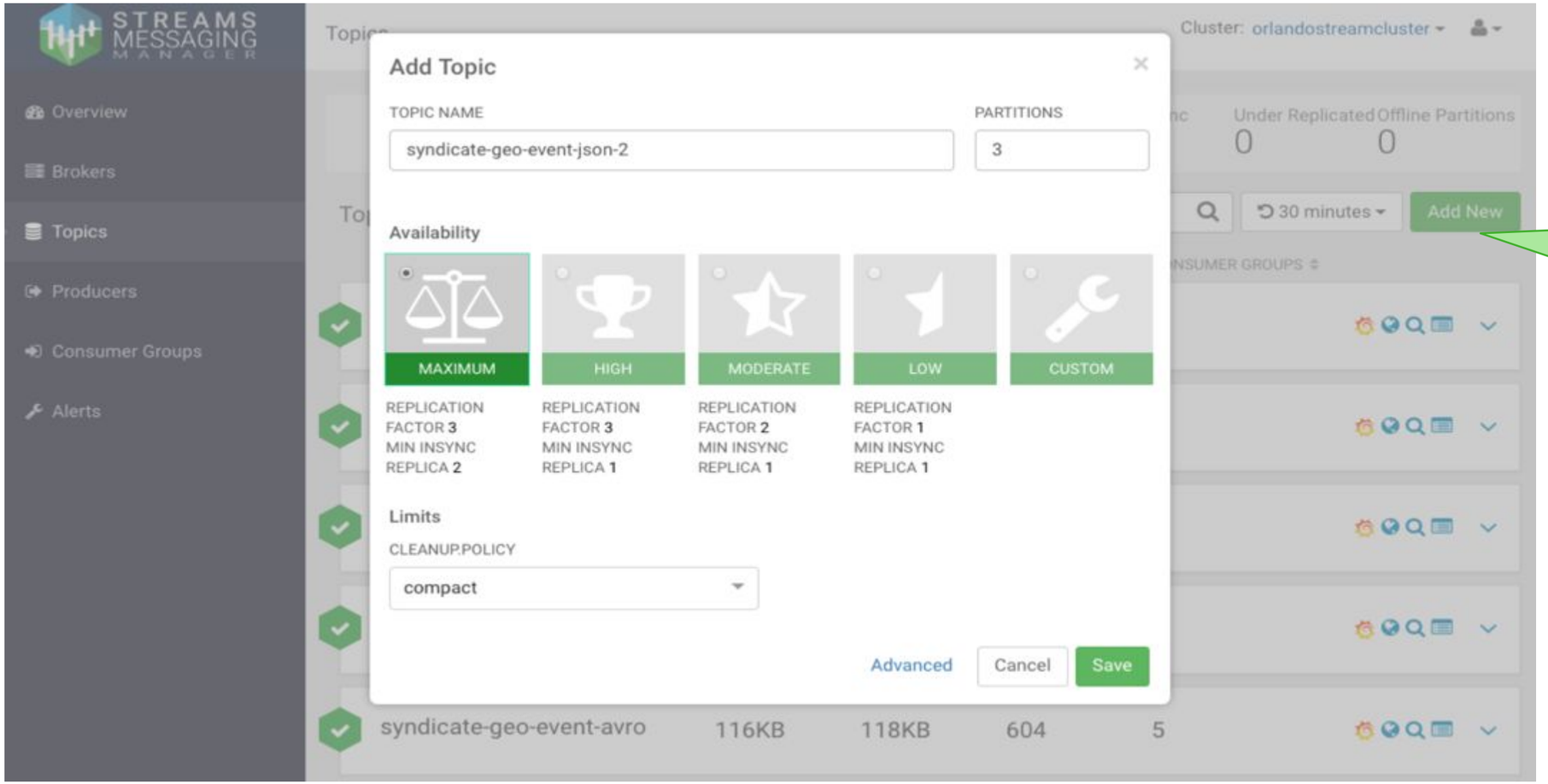
```
[cloudbreak@ip-10-0-1-199 ~]$ kafka-topics.sh --list --zookeeper ip-10-0-1-115.eu-west-1.compute.internal:2181
ATLAS_ENTITIES
ATLAS_HOOK
__consumer_offsets
__smm_alert_notifications
__transaction_state
adjudication
alerts-speeding-drivers
approval
audit-events
compliance
energy-mgmt
fleet-supply-chain
fuel-logistics
gateway-central-raw-sensors
gateway-east-raw-sensor
gateway-east-raw-sensors
gateway-europe-raw-sensors
gateway-west-raw-sensors
gdeleon-test
load-optimization
myFirstTopic
nifi-kafka-demo
predictive-alerts
route-planning
supply-chain
syndicate-all-geo-critical-events
syndicate-battery
syndicate-geo-event-avro
syndicate-geo-event-json
syndicate-oil
syndicate-speed-event-avro
syndicate-speed-event-json
syndicate-transmission
[cloudbreak@ip-10-0-1-199 ~]$ pwd
```

```
Last login: Tue Mar 19 00:42:54 2019 from 195.53.52.170
cat /etc/motd
                    .-..-_
                  /`     '.
              __ /`       '.
           .-'  `/   (   a \
          /      (    \,_   \
         /|       '---` |\ =|
        ` \    /__.-/  /  | |
           |  / / \ \  \   \_\
           |__|_|  |_|__\
================================================
  *                          :

[cloudbreak@ip-10-0-1-199 ~]$ export PATH=$PATH:/usr/hdp/current/kafka-broker/bin/
[cloudbreak@ip-10-0-1-199 ~]$ kafka-configs.sh --zookeeper ip-10-0-1-136.eu-west-1.compute.internal:2181
escribe --entity-name testKnoxSetUp
Configs for topic 'testKnoxSetUp' are cleanup.policy=compact,delete
```

```
[cloudbreak@ip-10-0-1-199 ~]$ kafka-topics.sh --zookeeper ip-10-0-1-136.eu-west-1.compute.internal:2181
tion-factor 1 --topic topic-command-line
Created topic "topic-command-line".
[cloudbreak@ip-10-0-1-199 ~]$
```

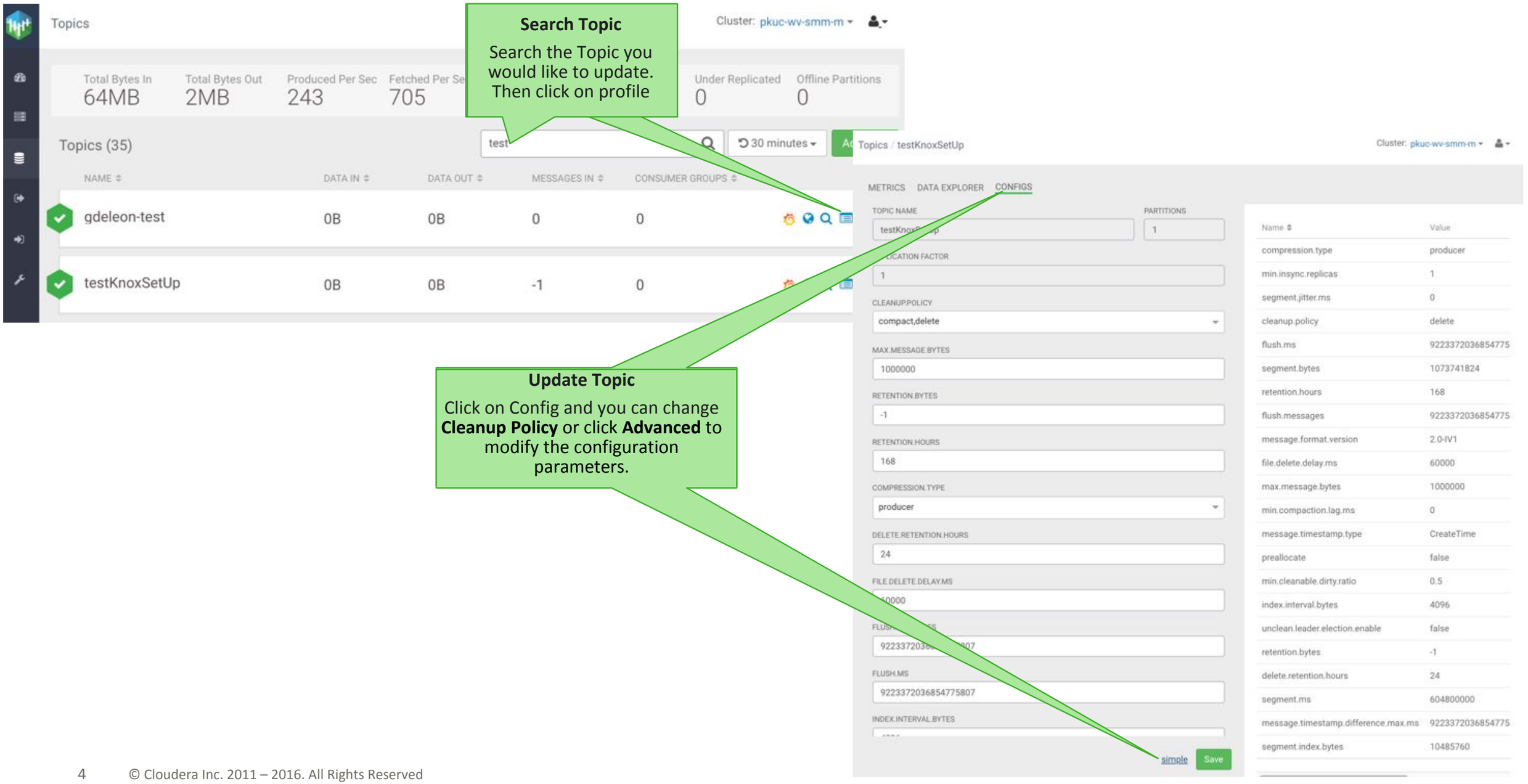
Topic Lifecycle Management



Add Topic

User friendly UI to create new topics. Simple and Advance features are available

Topic Update

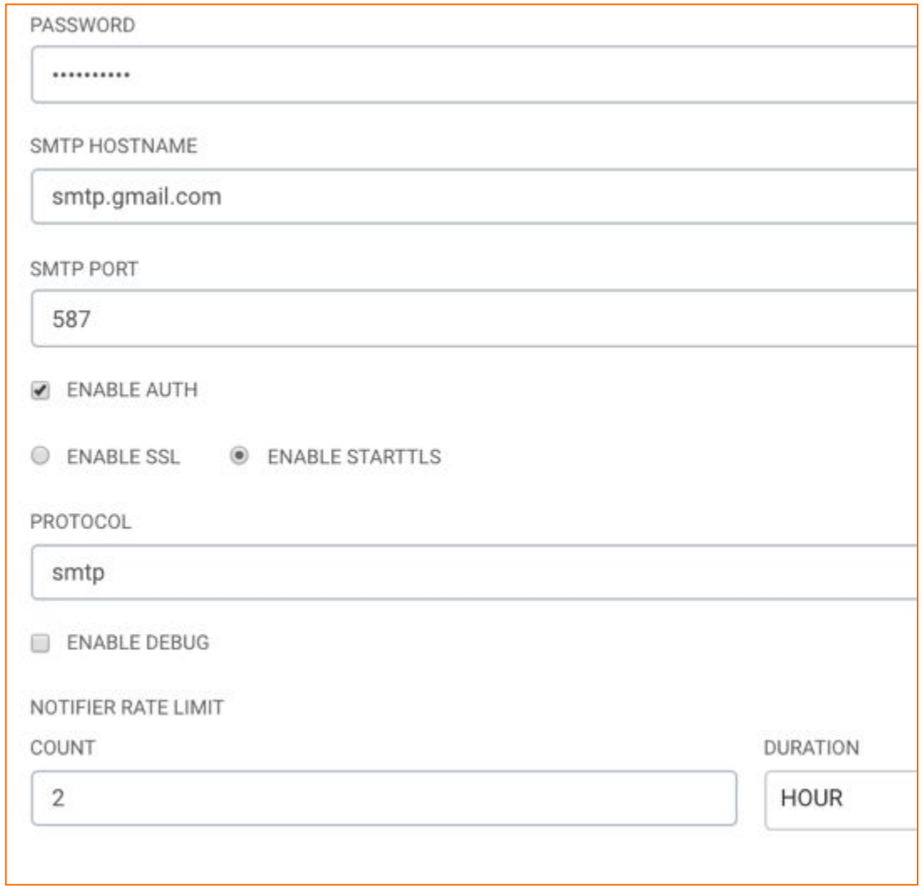


Alerting – Create Notifier

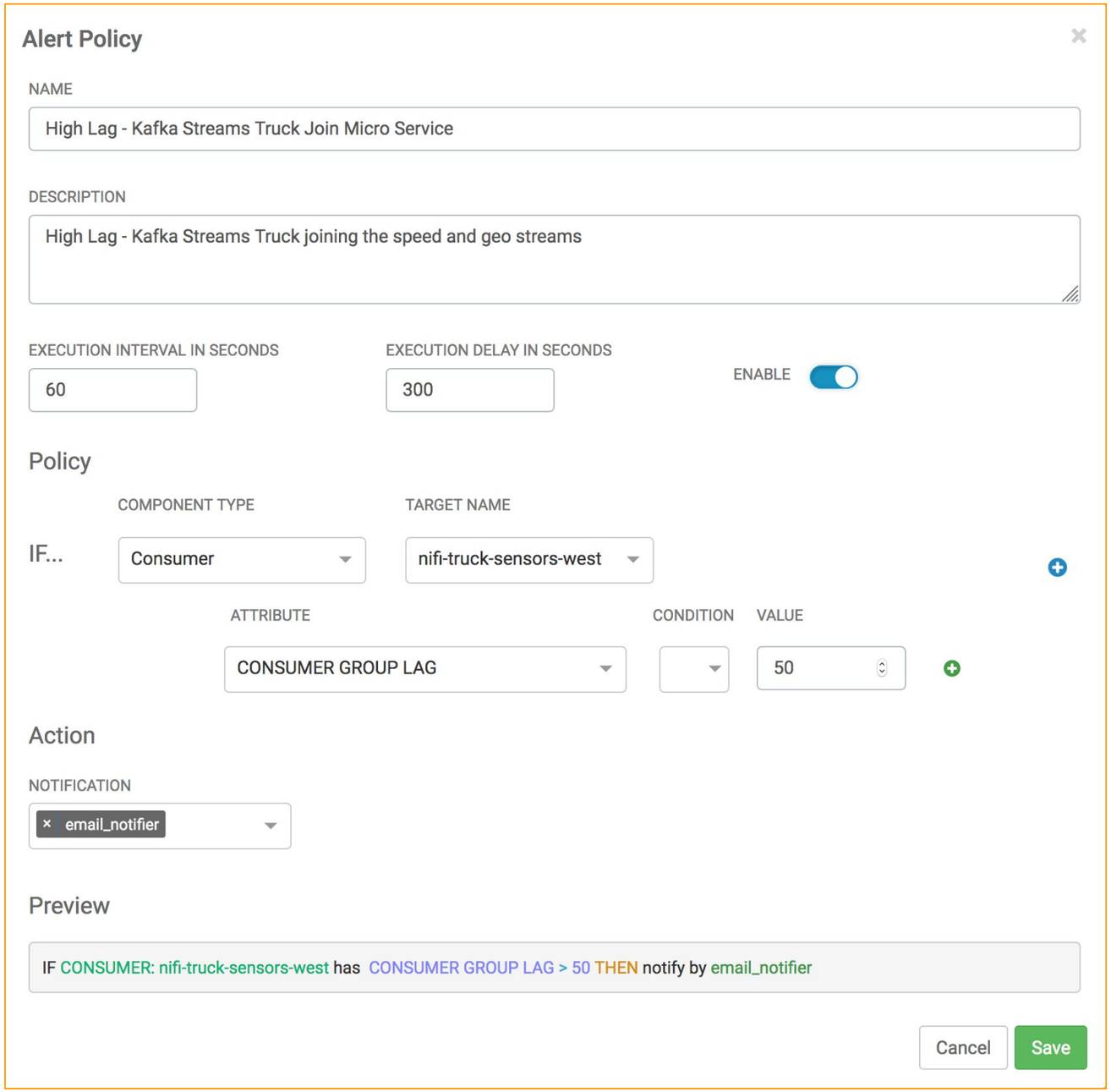
1st Key Construct of Alerts

Alert Notifier

1. Email
2. http end point
3. Kafka topic



Alerting – Create Alert

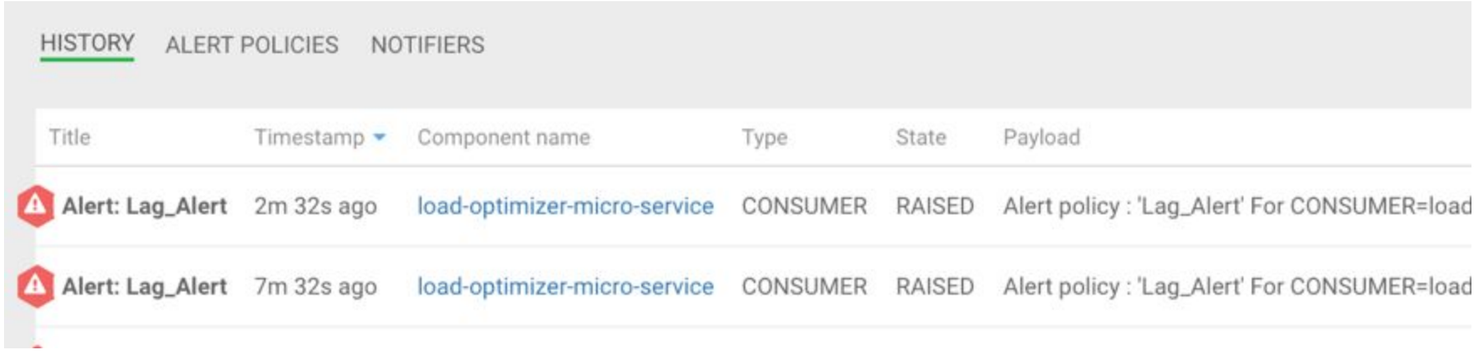


2nd Key Construct of Alerts

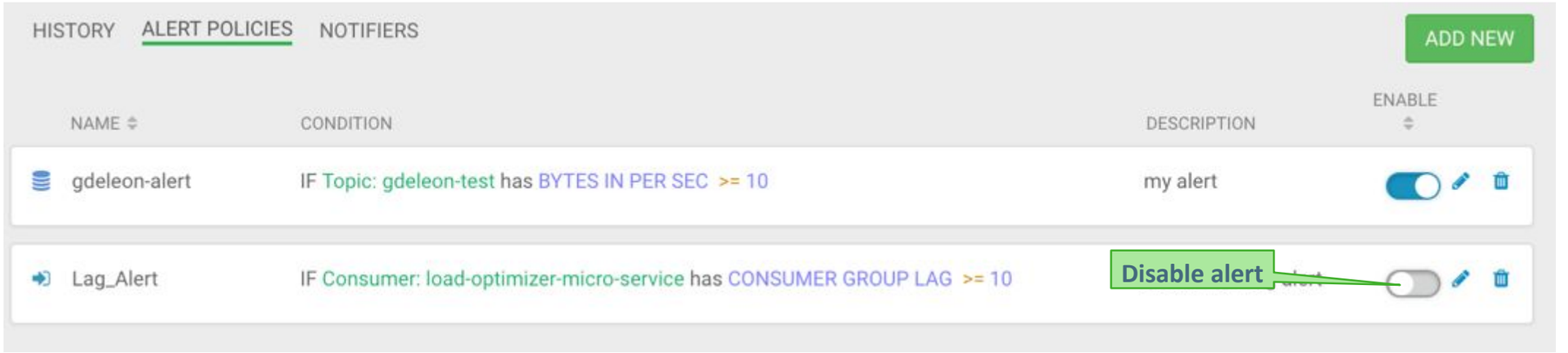
Alert Policy

1. Defined for 6 key entities
   (cluster, broker, topic, producer, consumer, latency, cluster replication)
2. Metrics defined on entities
3. Complex alerts
4. Includes notifier when triggered

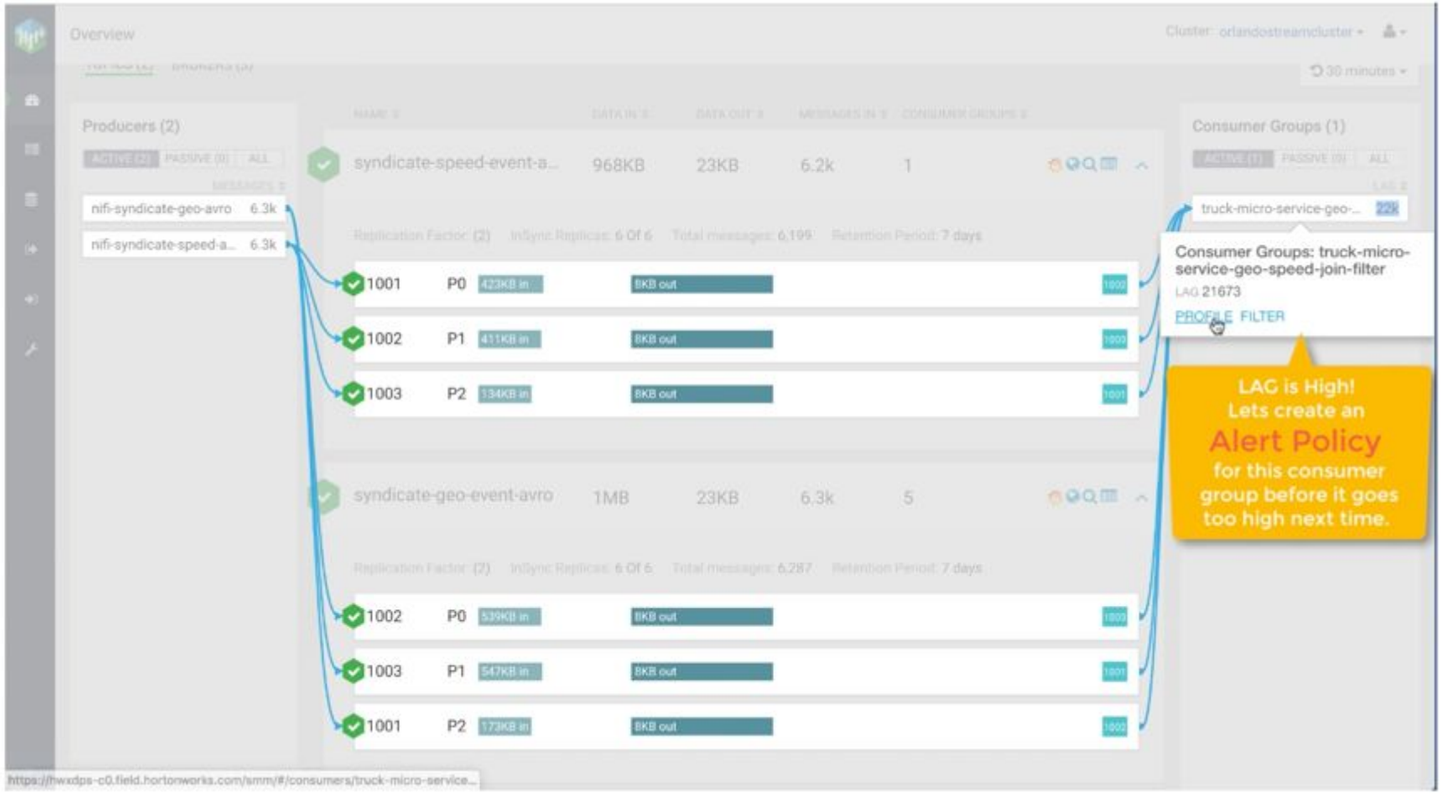
Alert History



Disable Alert



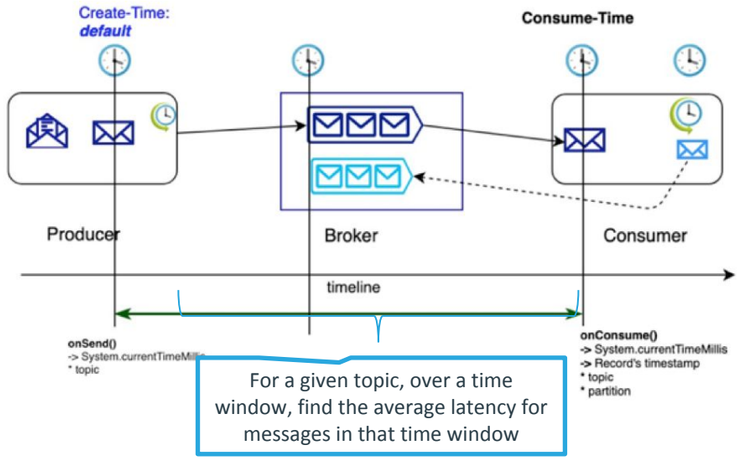
Example: Alerting on Micro-Service Consumer Group with High Lag



New DevOps Monitoring Capability with End to End Latency View

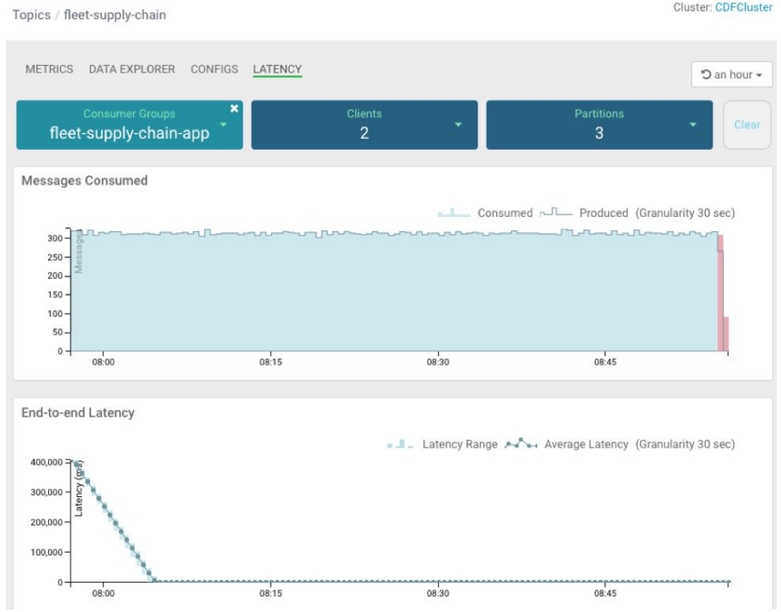
Problem Statement / Requirements

- Kafka DevOps teams need the ability to see an end to end latency view of messages produced and consumed across producers to topics/brokers to consumers
- Common DevOps questions include:
  - For the last five minutes, what is the average latency of messages consumed after being produced.
  - For the last hour, which topics have under consumption?

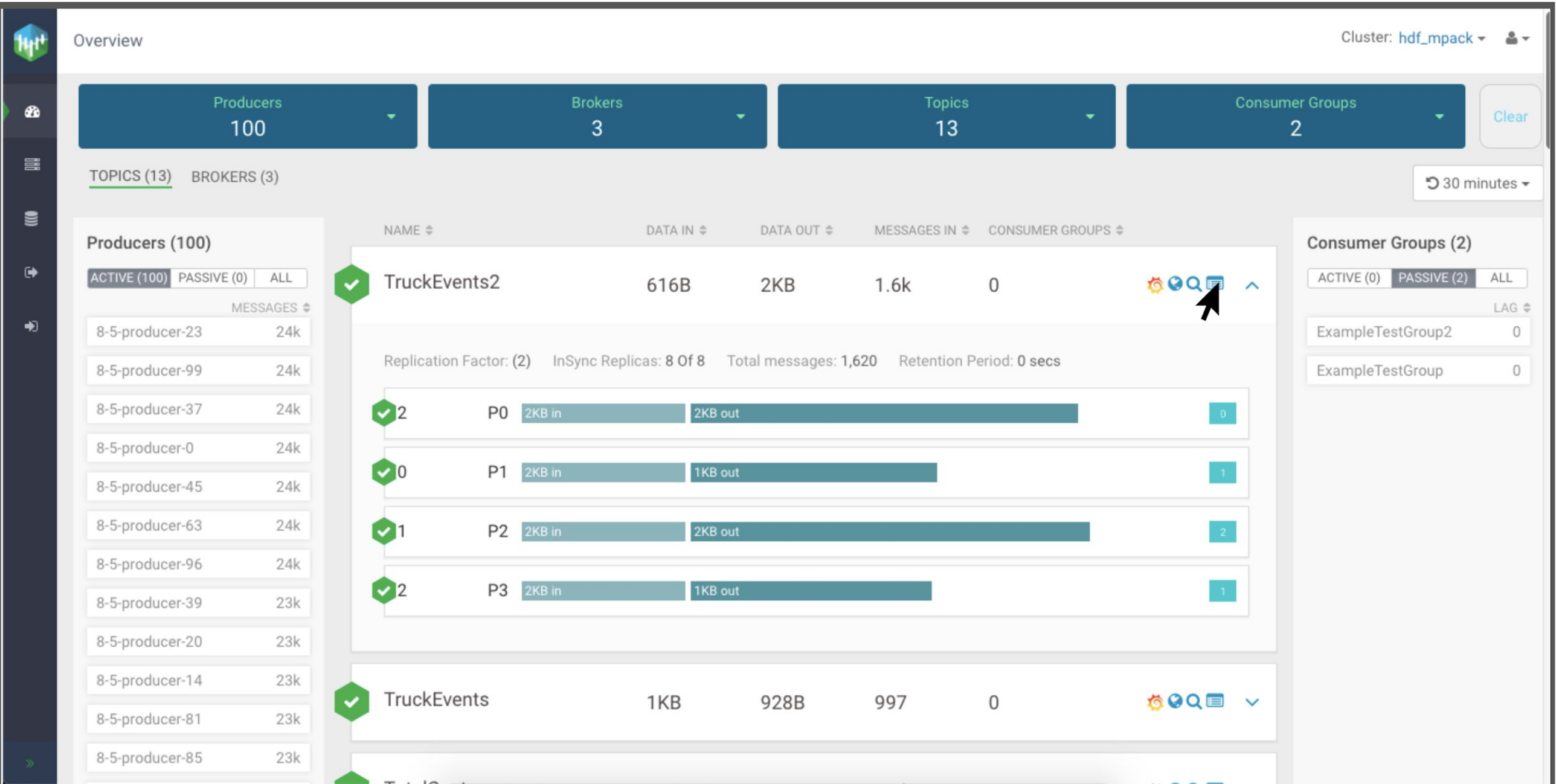


Solution / Benefit

- SMM 2.0 provides new monitoring view for end to end Latency
- End to End Latency view is powered by new embedded Kafka streams application that calculates end to end latencies and new interceptors that can be used to instrument producer and consumer clients.



Select a Topic of Interest



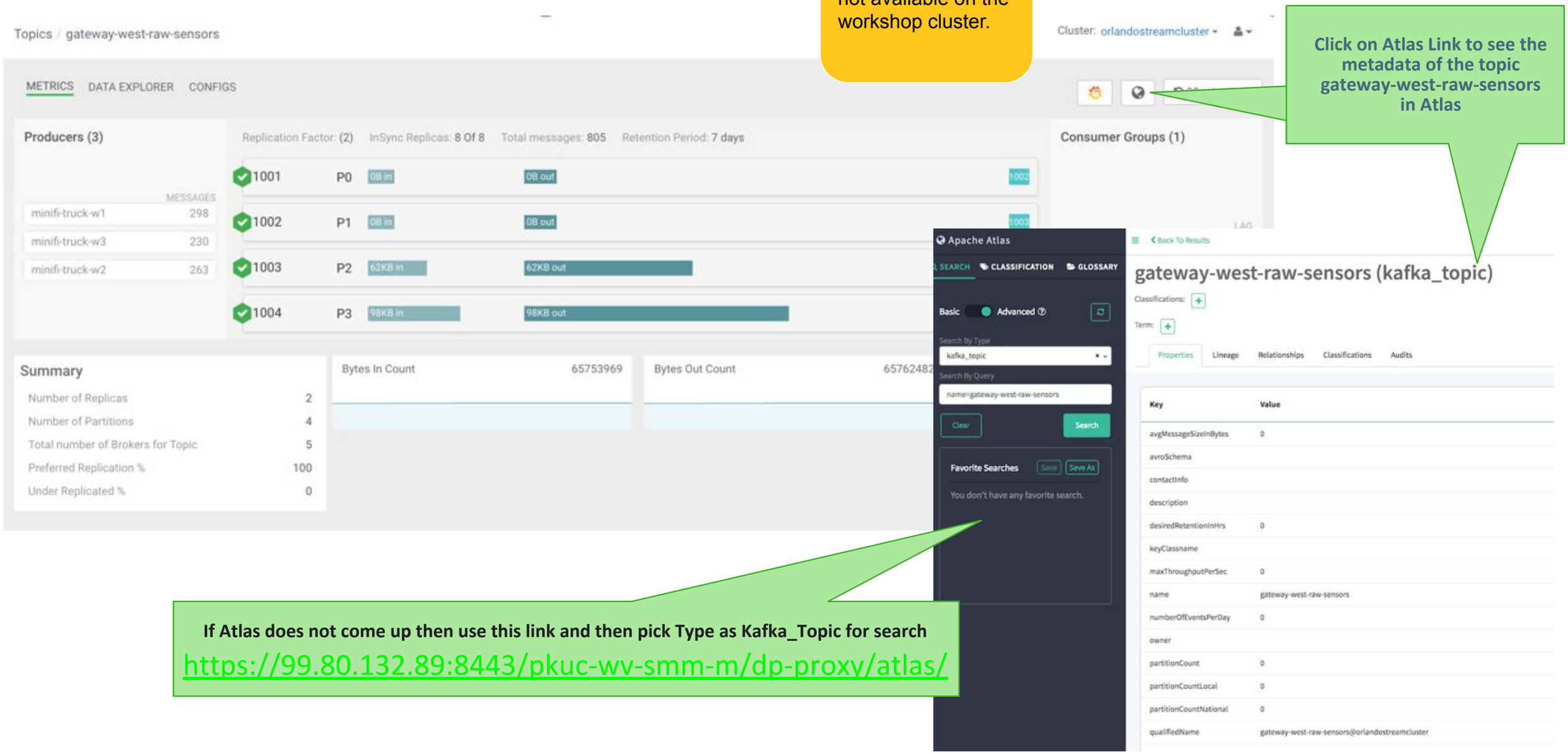
Select the End to End Latency Tab

SMM Atlas Integration

Explore Metadata about the Topic in Atlas

Atlas Integration is not available on the workshop cluster.

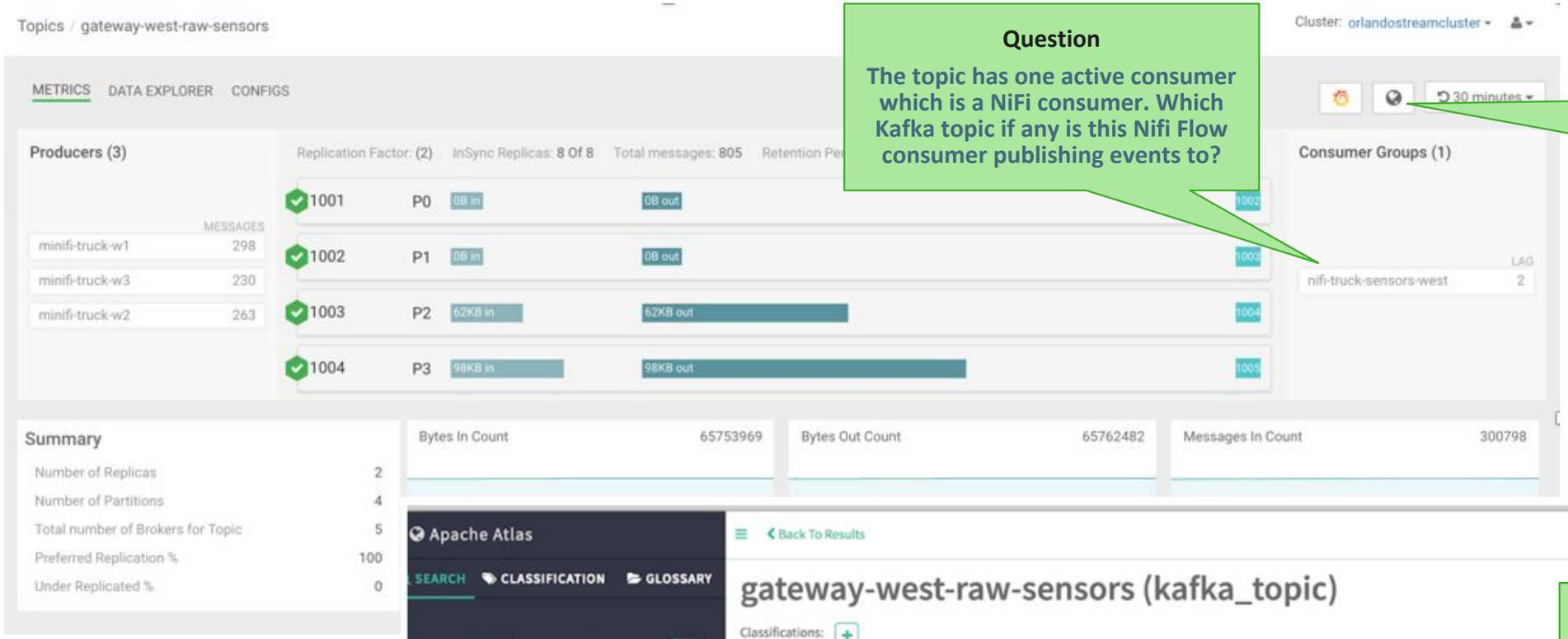
Click on Atlas Link to see the metadata of the topic gateway-west-raw-sensors in Atlas



If Atlas does not come up then use this link and then pick Type as Kafka_Topic for search

https://99.80.132.89:8443/pkuc-wv-smm-m/dp-proxy/atlas/

Traverse the flow of data across multiple Kafka Topics using SMM and Atlas Integration

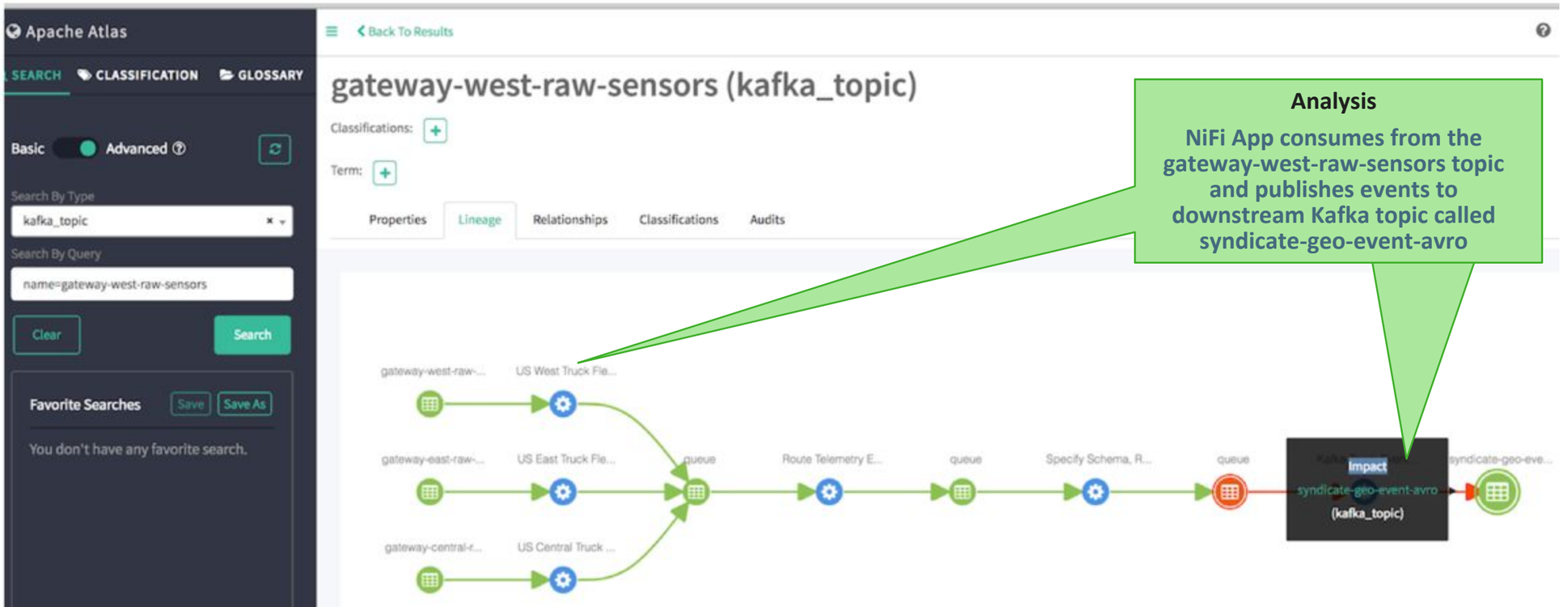


Question

The topic has one active consumer which is a NiFi consumer. Which Kafka topic if any is this Nifi Flow consumer publishing events to?

Step 1

Click on Atlas Icon to see lineage of the the topic gateway-west-raw-sensors



Analysis

NiFi App consumes from the gateway-west-raw-sensors topic and publishes events to downstream Kafka topic called syndicate-geo-event-avro

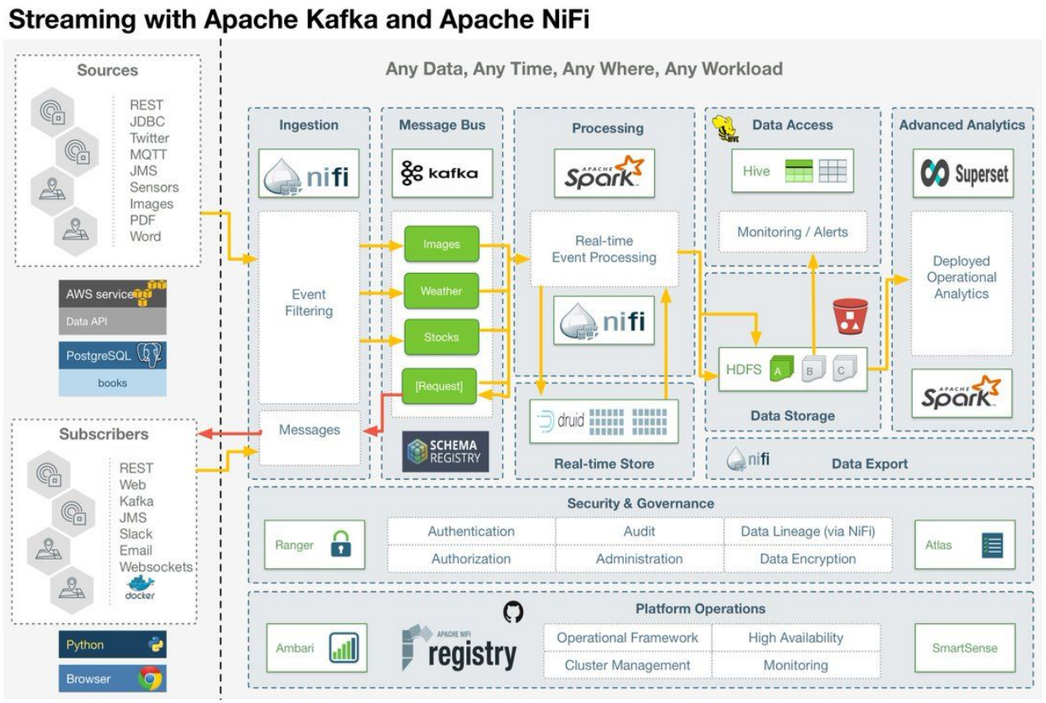
Atlas Integration is not available on the workshop cluster.

CLOUDERA

SMM Workshop - Kafka Streams

Timothy Spann

DZone Zone Leader and Big Data MVB;
Princeton Future of Data Meetup;
ex-Pivotal Field Engineer;
Author of Apache Kafka RefCard
https://github.com/tspannhw



https://community.hortonworks.com/articles/227560/real-time-stock-processing-with-apache-nifi-and-ap.html

https://community.hortonworks.com/users/9304/tspann.html

https://dzone.com/users/297029/bunkertor.html

https://www.datainmotion.dev/

CLOUDERA

Contents

Kafka CRUD

Kafka Streams Overview

Kafka Streams Walk Through

Kafka Topics and Testing Your Microservice

Kafka Streams Architecture

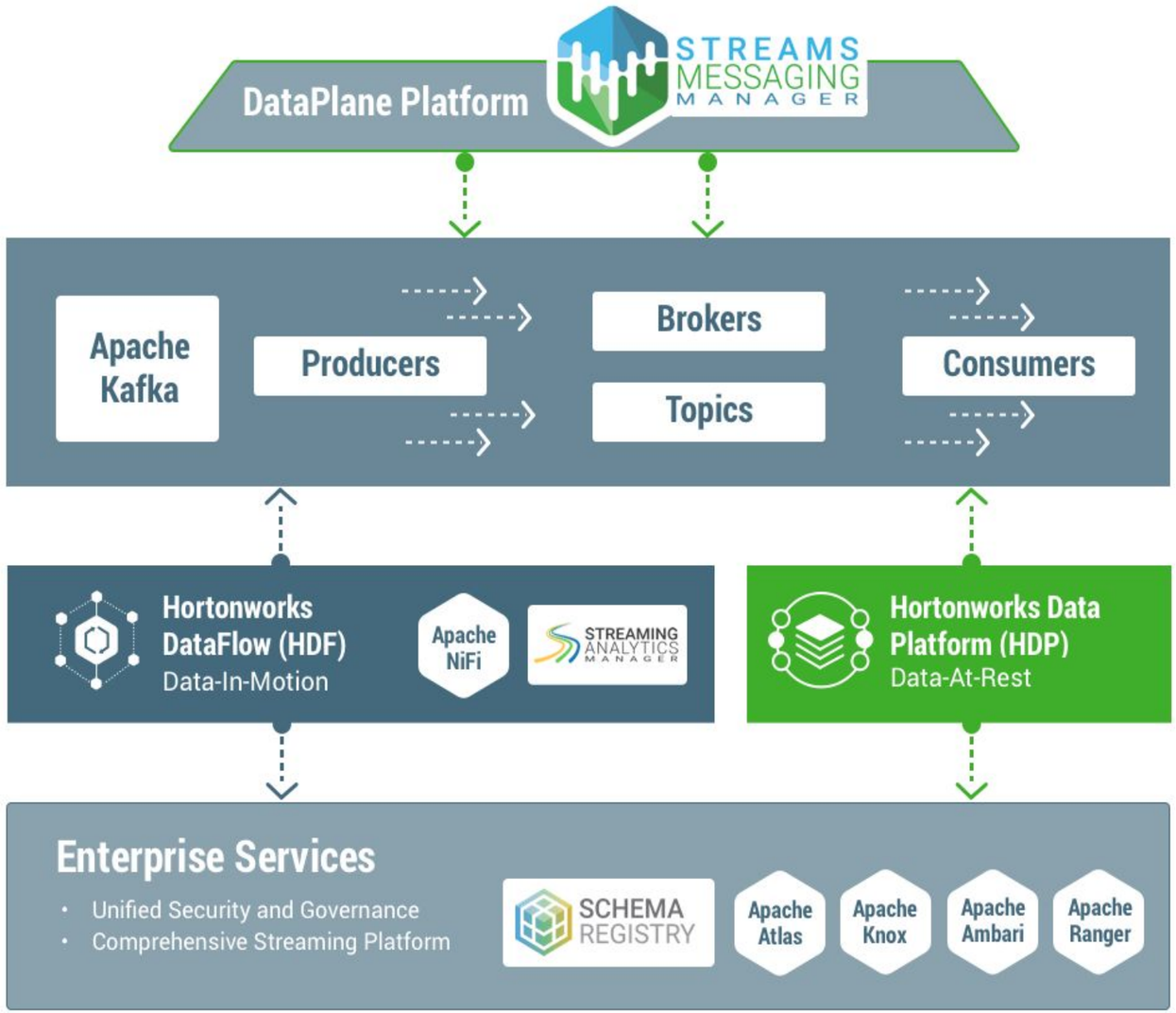
Kafka Streams Example

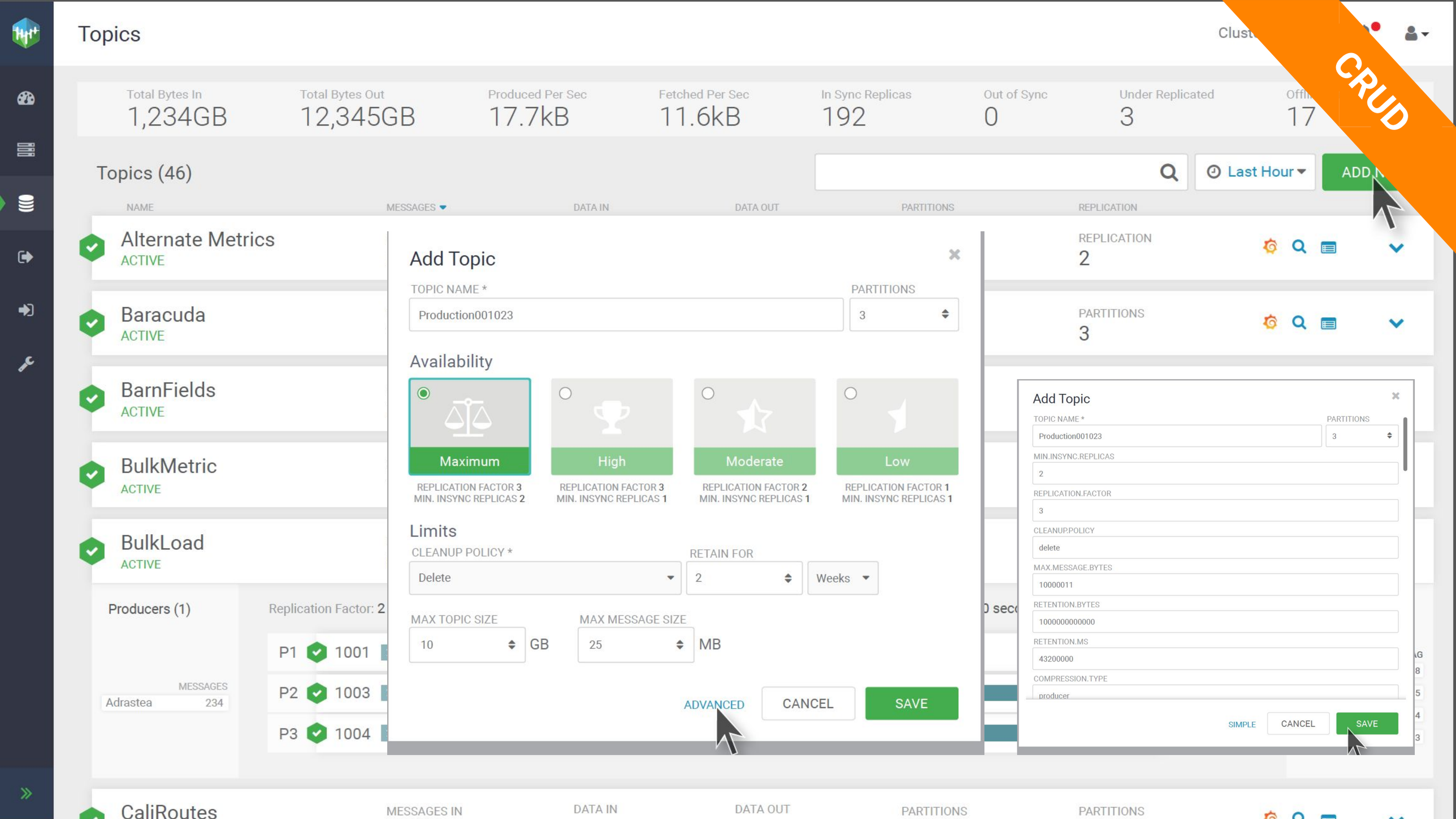
https://github.com/purn1mak/SMM-NewYork

KAFKA CRUD

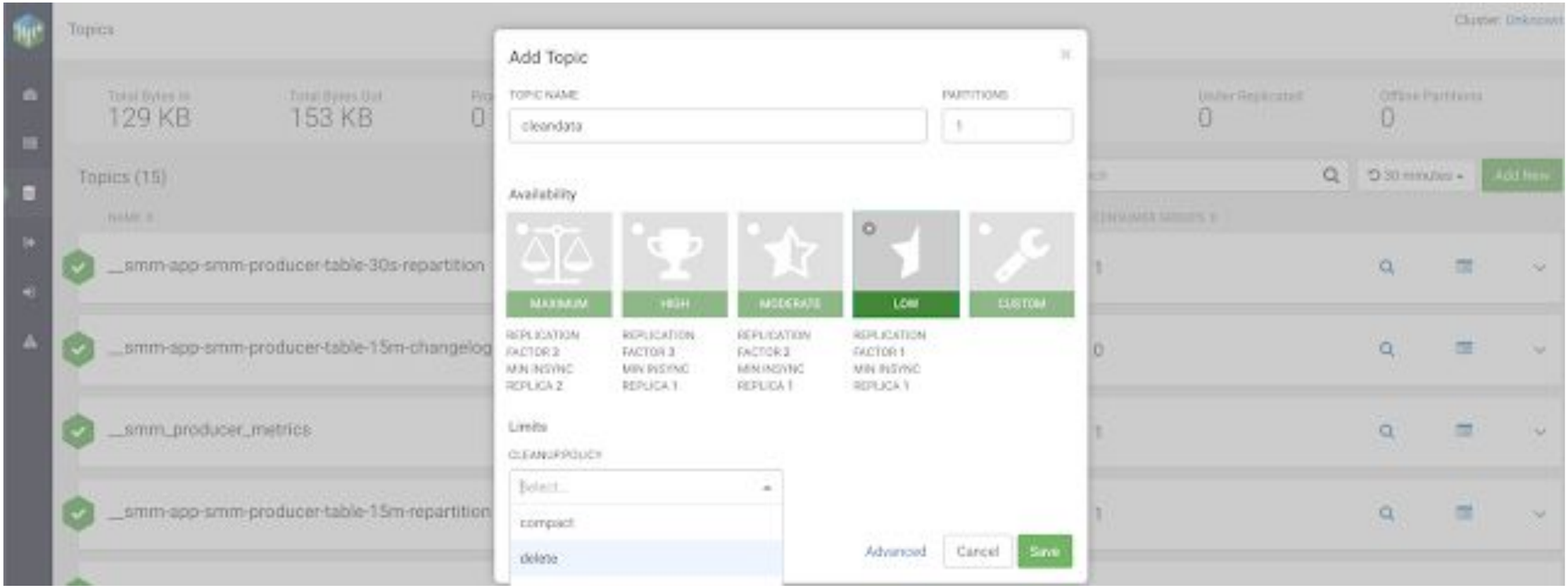
Streams Messaging Manager (SMM)

- Open Source tool to Cure the “Kafka Blindness”
- Single Monitoring Dashboard for all your Kafka Clusters across 4 entities
- Notification on available metrics
- Designed for the Enterprise
  - Support for Secure Kafka cluster
  - Rich Access Control Policies (ACLS)
  - Supports multiple Kafka Clusters
- REST as a First Class Citizen



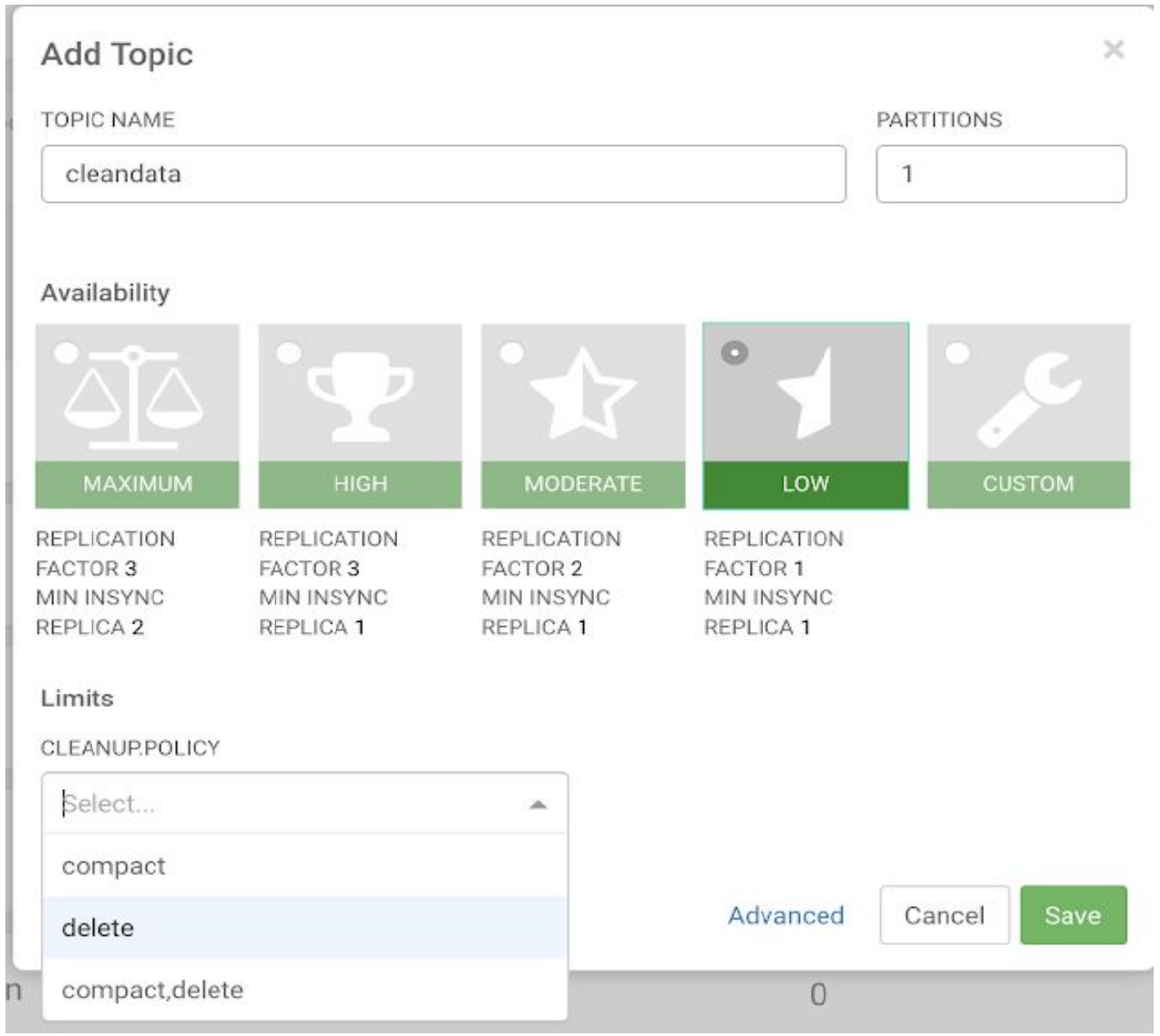

Topics
Total Bytes In
1,234GB
Total Bytes Out
12,345GB
Produced Per Sec
17.7kB
Fetched Per Sec
11.6kB
In Sync Replicas
192
Out of Sync
0
Under Replicated
3
17
CRUD
Topics (46)
Last Hour
NAME
MESSAGES
DATA IN
DATA OUT
PARTITIONS
REPLICATION
Alternate Metrics
ACTIVE
REPLICATION
2
Baracuda
ACTIVE
PARTITIONS
3
BarnFields
ACTIVE
BulkMetric
ACTIVE
BulkLoad
ACTIVE
Producers (1)
Replication Factor: 2
MESSAGES
Adrastea 234
P1 1001
P2 1003
P3 1004
Add Topic
TOPIC NAME *
Production001023
PARTITIONS
3
Availability
Maximum
REPLICATION FACTOR 3
MIN. INSYNC REPLICAS 2
High
REPLICATION FACTOR 3
MIN. INSYNC REPLICAS 1
Moderate
REPLICATION FACTOR 2
MIN. INSYNC REPLICAS 1
Low
REPLICATION FACTOR 1
MIN. INSYNC REPLICAS 1
Limits
CLEANUP POLICY *
Delete
RETAIN FOR
2
Weeks
MAX TOPIC SIZE
10
GB
MAX MESSAGE SIZE
25
MB
ADVANCED
CANCEL
SAVE
Add Topic
TOPIC NAME *
Production001023
PARTITIONS
3
MIN.INSYNC.REPLICAS
2
REPLICATION.FACTOR
3
CLEANUP.POLICY
delete
MAX.MESSAGE.BYTES
10000011
RETENTION.BYTES
1000000000000
RETENTION.MS
43200000
COMPRESSION.TYPE
producer
SIMPLE
CANCEL
SAVE
CaliRoutes
MESSAGES IN
DATA IN
DATA OUT
PARTITIONS
PARTITIONS

Create a Kafka Topic



https://www.datainmotion.dev/2019/08/using-cloudera-streams-messaging.html

Create a Kafka Topic



- Topic Name
- Partitions (1 or more)
- Pick a Template with replicas and replication factor
- Carefully select a Cleanup Policy **DELETE**
  - (your messages need a key)

https://www.datainmotion.dev/2019/08/using-cloudera-streams-messaging.html

Create Kafka Topics For Kafka Streams Application

- Topic Name: **streams-plaintext-input**
- Partitions: 1
- Replicas: 1
- Delete

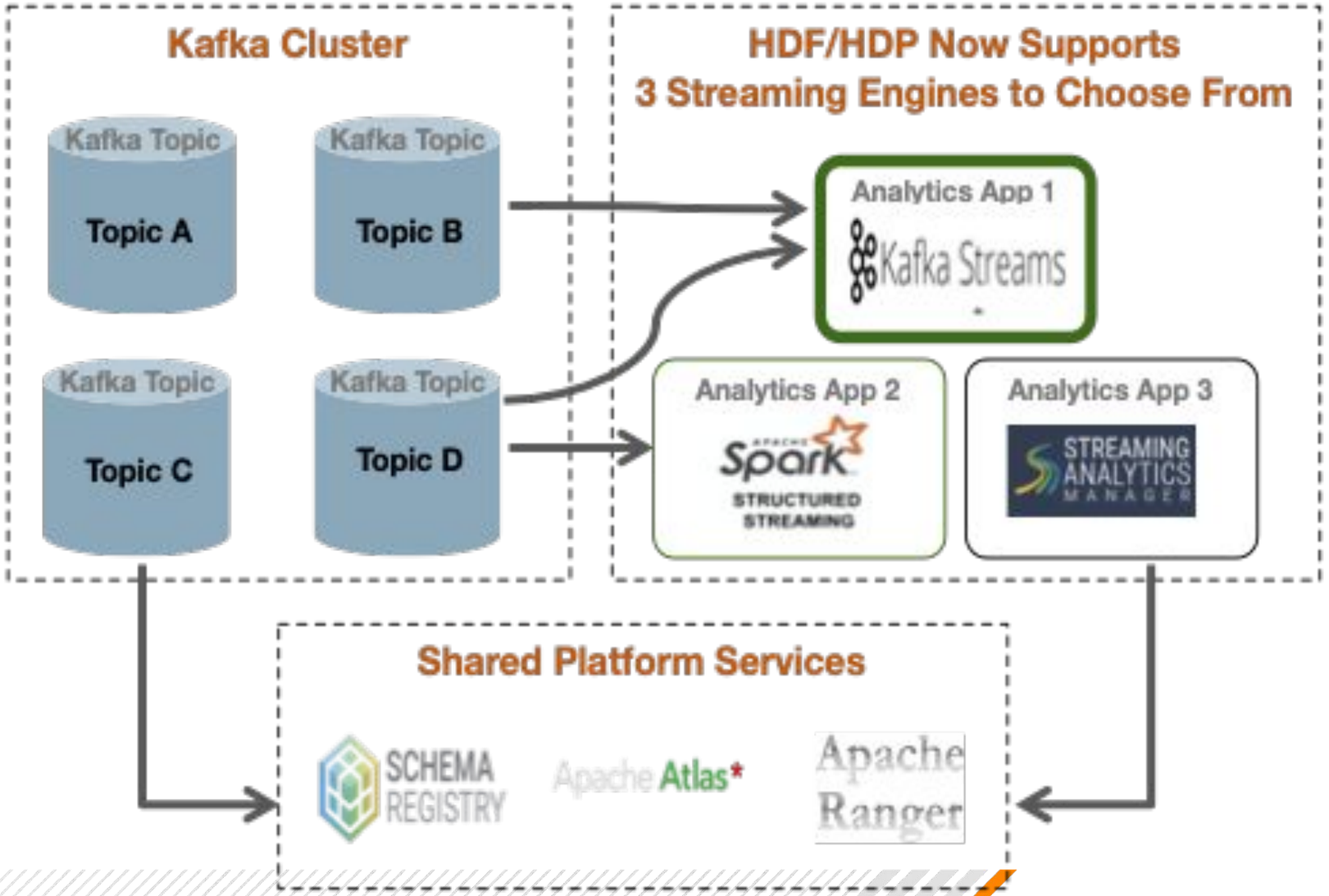
- Topic Name: **streams-wordcount-output**
- Partitions: 1
- Replicas: 1
- Delete

KAFKA STREAMS

Secure and Governed Microservices with Kafka Streams Support

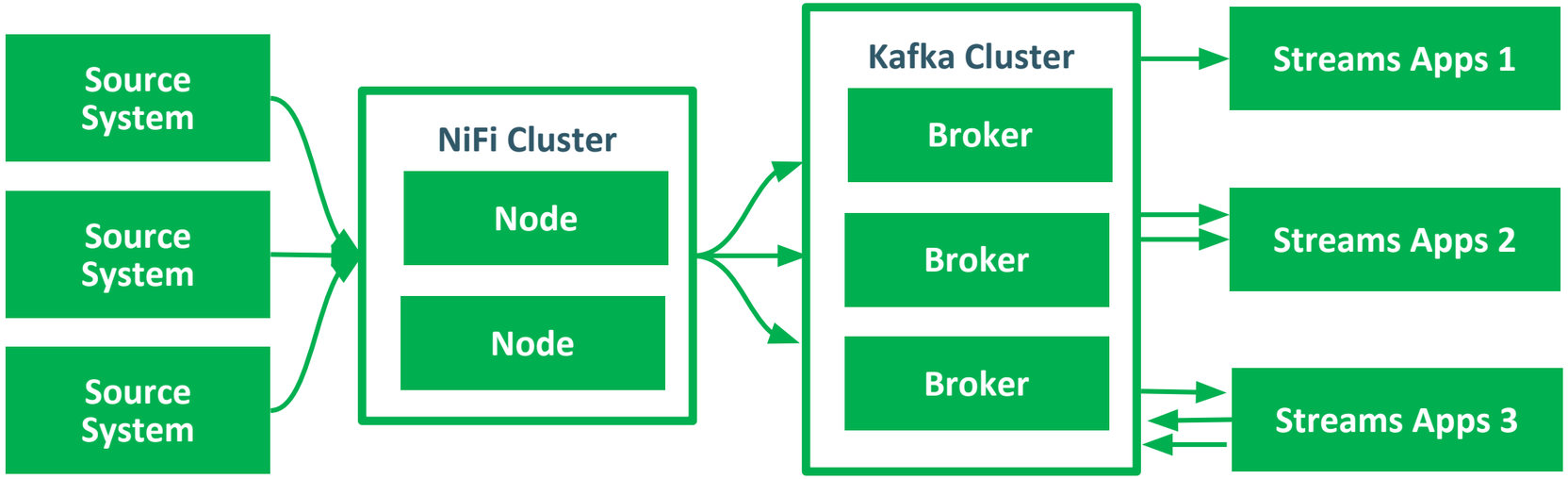
APACHE kafka Streams

API for building real-time applications and microservices on top of Kafka



Kafka Streams

- A **client library** for applications and microservices on top of Kafka
- Written in Java or Scala
- Elastic, highly scalable, fault-tolerant
- Supports At-Least-Once and Exactly-once semantics inside Kafka
- **Deploy** to containers, VMs, bare metal, cloud, K8, devices.



Kafka Streams Library

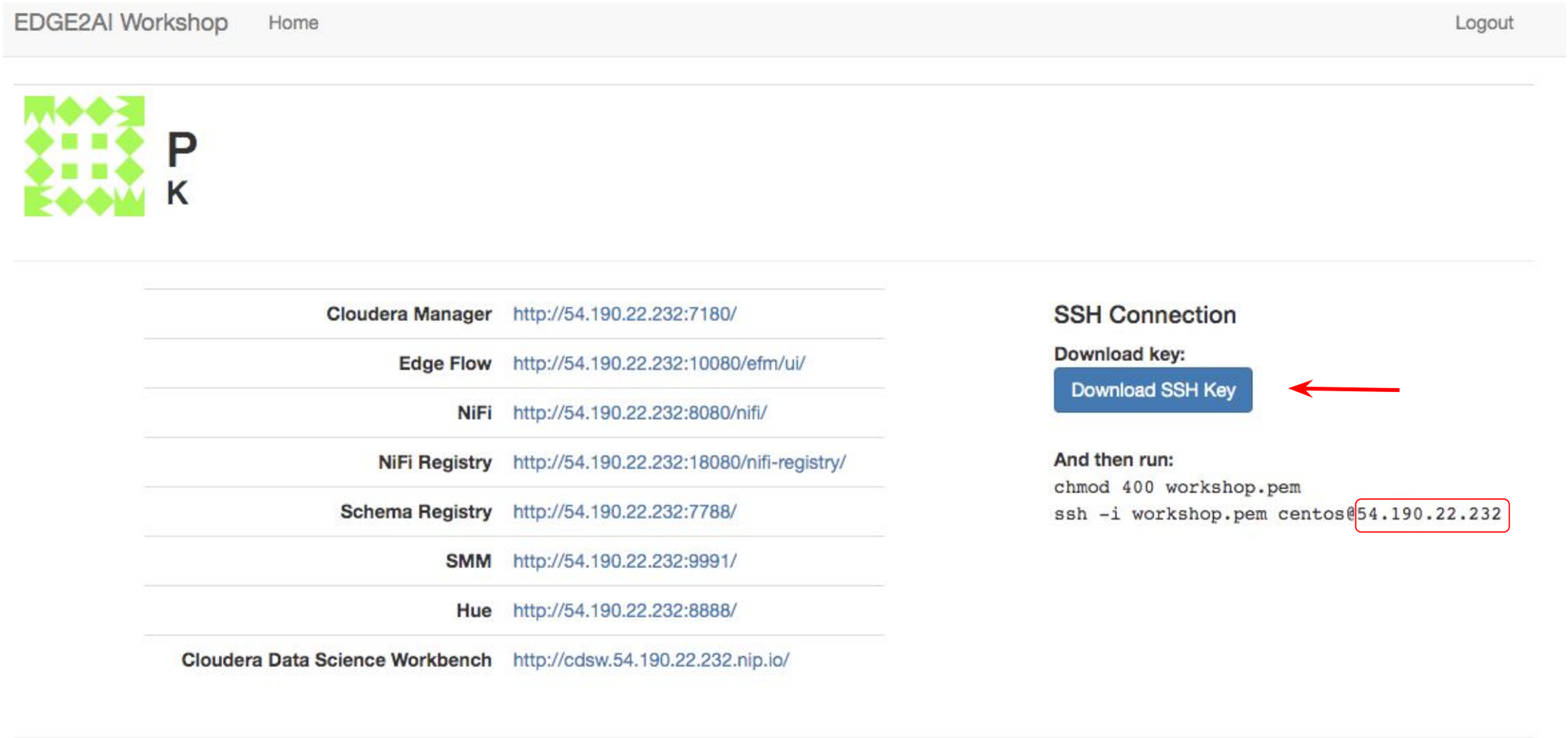
Stream Processing Style	Event a time
Delivery Guarantee	Exactly Once (within Kafka)
State Management	Implicit support with RocksDB
Fault Tolerance	Implicit support with internal Kafka topics / ZK
Advanced feature: Joins/ Aggregations/ Windowing	Stream/Stream joins, Stream/Table joins, joins with only message key, no OOO aggregation processors
Advanced Feature: Watermarking (late arriving data)	Not fully supported
Latency	Low
Throughput	Medium
APIS	Simple APIs (Compositional)

Kafka Streams Library

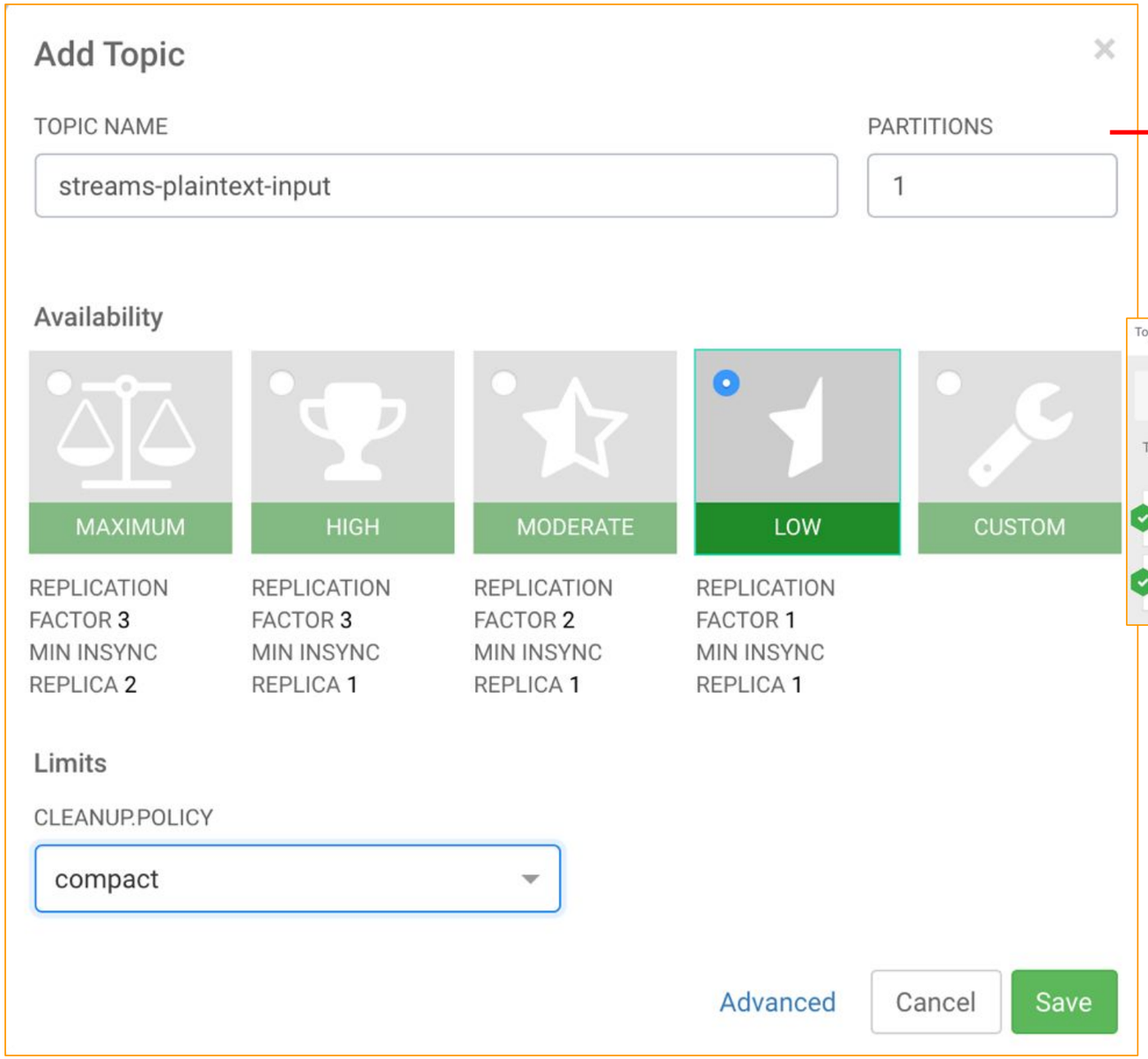
SQL DSL	Nothing in Apache Kafka.
Lambda Architecture	No Supported
GUI Tooling / Code-Less Approach	Not Supported
Maturity	Relatively New
Use Cases	Lightweight eventing based microservices, ETL
Cluster Requirement	No

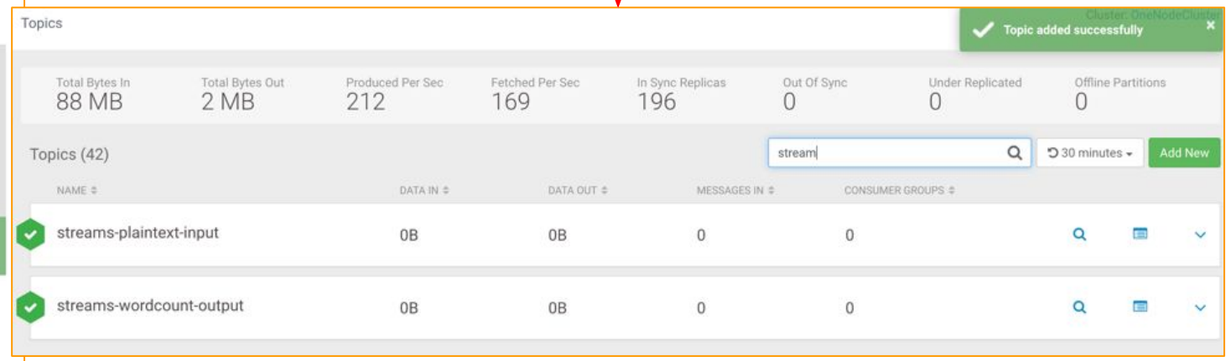
KAFKA STREAMS WALK THROUGH

Tutorial - Log In Instructions



Topic - Create





Setup Your Development Environment

```
sudo su
yum install maven curl wget unzip zip git -y
mkdir /opt/demo
chmod -R 777 /opt/demo
cd /opt/demo
```

Generate a Kafka Streams Project

```
mvn archetype:generate \
    -DarchetypeGroupId=org.apache.kafka \
    -DarchetypeArtifactId=streams-quickstart-java \
    -DarchetypeVersion=2.2.0 \
    -DgroupId=streams.examples \
    -DartifactId=streams.examples \
    -Dversion=0.1 \
    -Dpackage=myapps
```

Update Word Count Kafka Connection

Edit the file
/opt/demo/streams.examples/src/main/java/myapps/WordCount.java

```
Properties props = new Properties();
props.put(StreamsConfig.APPLICATION_ID_CONFIG,
"streams-pipe");
props.put(StreamsConfig.BOOTSTRAP_SERVERS_CONFIG,
"<YOURAWSINTERNALIPNAME>:9092");
```

Build a Kafka Streams Project

```
rm /opt/demo/streams.examples/src/main/java/myapps/Pipe.java
rm /opt/demo/streams.examples/src/main/java/myapps/LineSplit.java
mvn clean package
```

KAFKA TOPIC INTERACTION

Produce Kafka Messages

/opt/cloudera/parcels/CDH/lib/kafka/bin/kafka-console-producer.sh --broker-list **<YOURAWSINTERNALIPNAME>**:9092 --topic streams-plaintext-input

Instructions: Then type messages with a return.

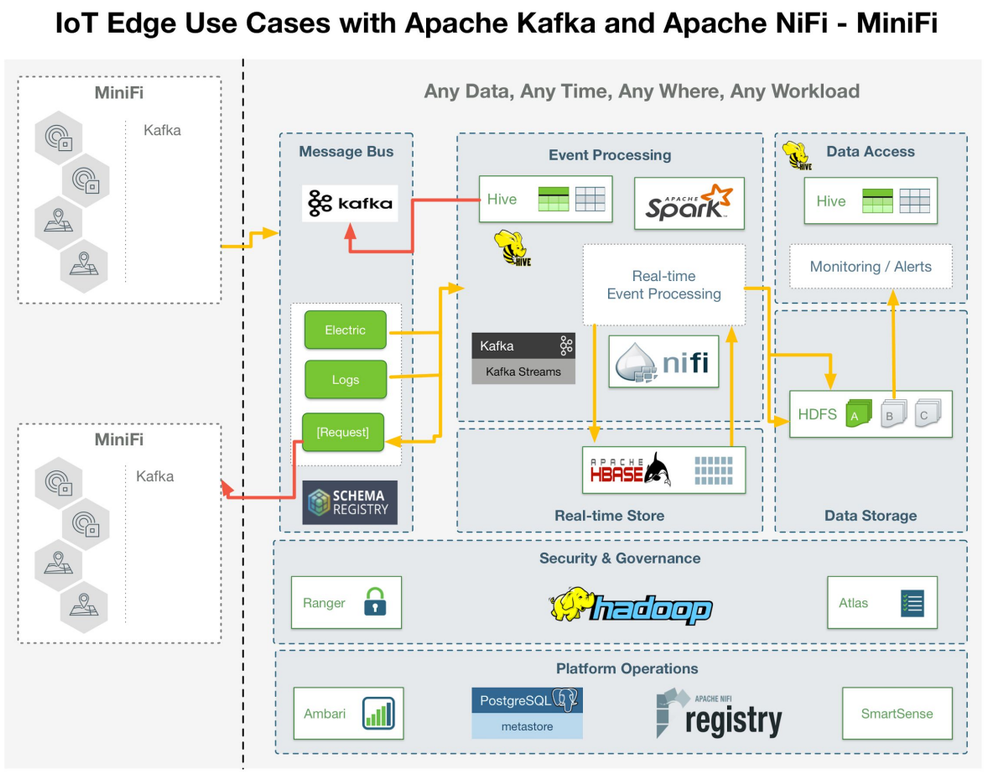
Run Kafka Streams Java Application

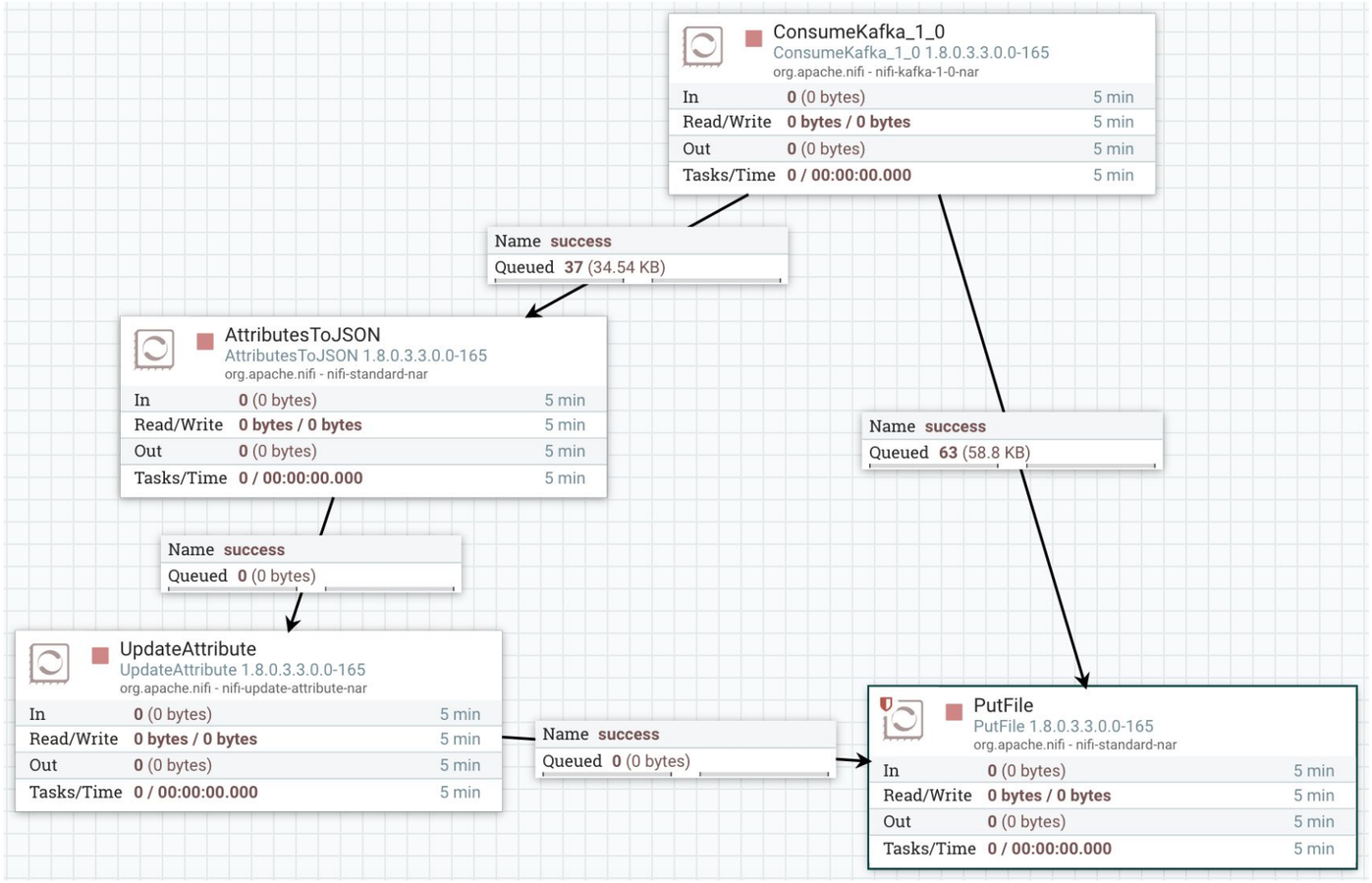
```
mvn exec:java -Dexec.mainClass=myapps.WordCount
```

Consume Kafka Messages

```
/opt/cloudera/parcels/CDH-6.3.0-1.cdh6.3.0.p0.1279813/lib/kafka/bin/kafka-console-cons
umer.sh --bootstrap-server <YOURAWSINTERNALIPNAME>:9092 \
   --topic streams-wordcount-output \
   --from-beginning \
   --formatter kafka.tools.DefaultMessageFormatter \
   --property print.key=true \
   --property print.value=true \
   --property key.deserializer=org.apache.kafka.common.serialization.StringDeserializer \
   --property value.deserializer=org.apache.kafka.common.serialization.LongDeserializer
```

CONSUME AND PRODUCE KAFKA MESSAGES

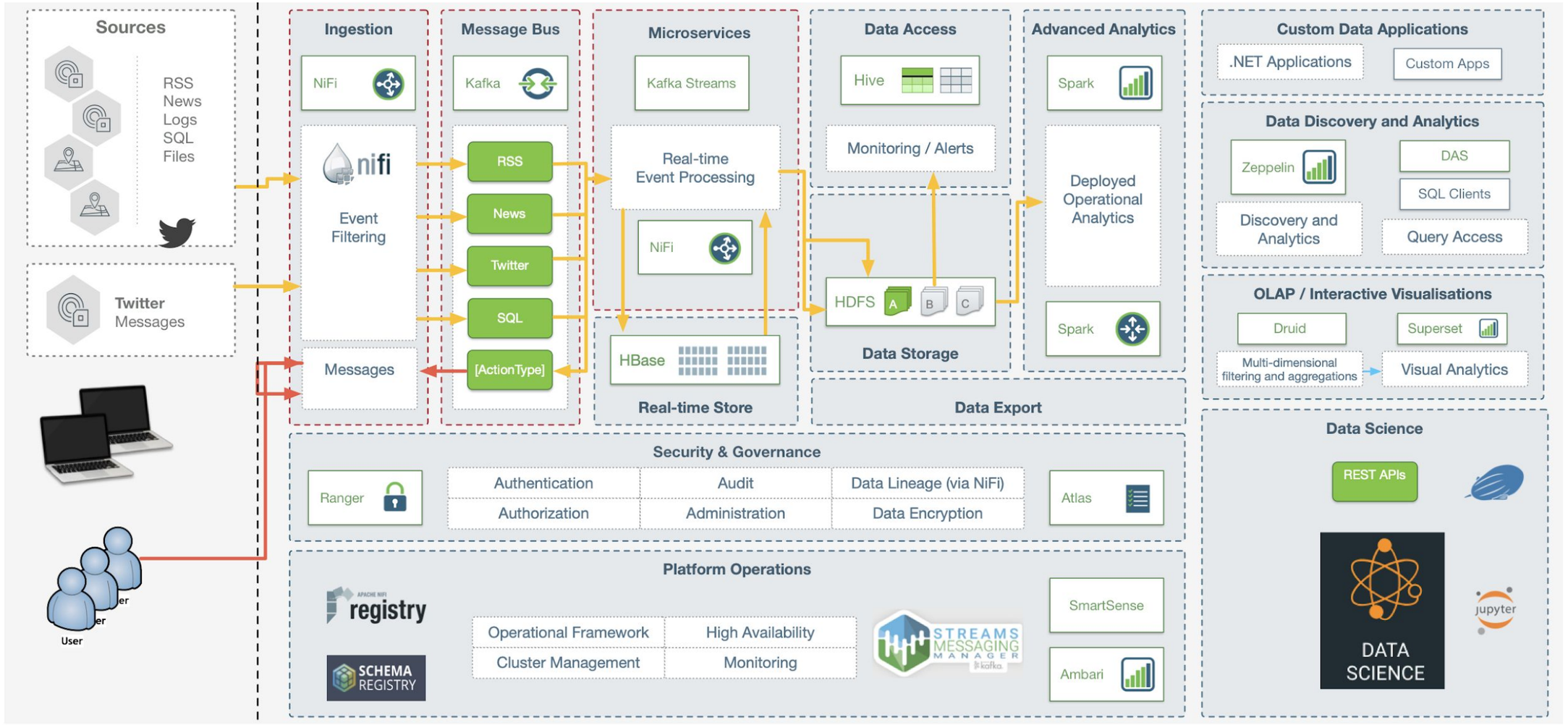




https://community.cloudera.com/t5/Community-Articles/IoT-Edge-Use-Cases-with-Apache-Kafka-and-Apache-NiFi-MiniFi/ta-p/249232

KAFKA STREAMS ADDITIONAL EXAMPLE

Kafka Streams Example Architecture



https://github.com/tspannhw/kstreams

Kafka Streams

```
Timer timer = new Timer();
timer.schedule(new DisplayStatus(), 0, 120000);

final StreamsBuilder builder = new StreamsBuilder();
KStream<String, String> source = builder.stream("bme680");

source.foreach((key, value) -> processValues(key, value));
source.to("bme680out");

final Topology topology = builder.build();
final KafkaStreams streams = new KafkaStreams(topology, props);
final CountDownLatch latch = new CountDownLatch(1);
```

https://www.datainmotion.dev/2019/03/iot-series-sensors-utilizing-breakout_28.html

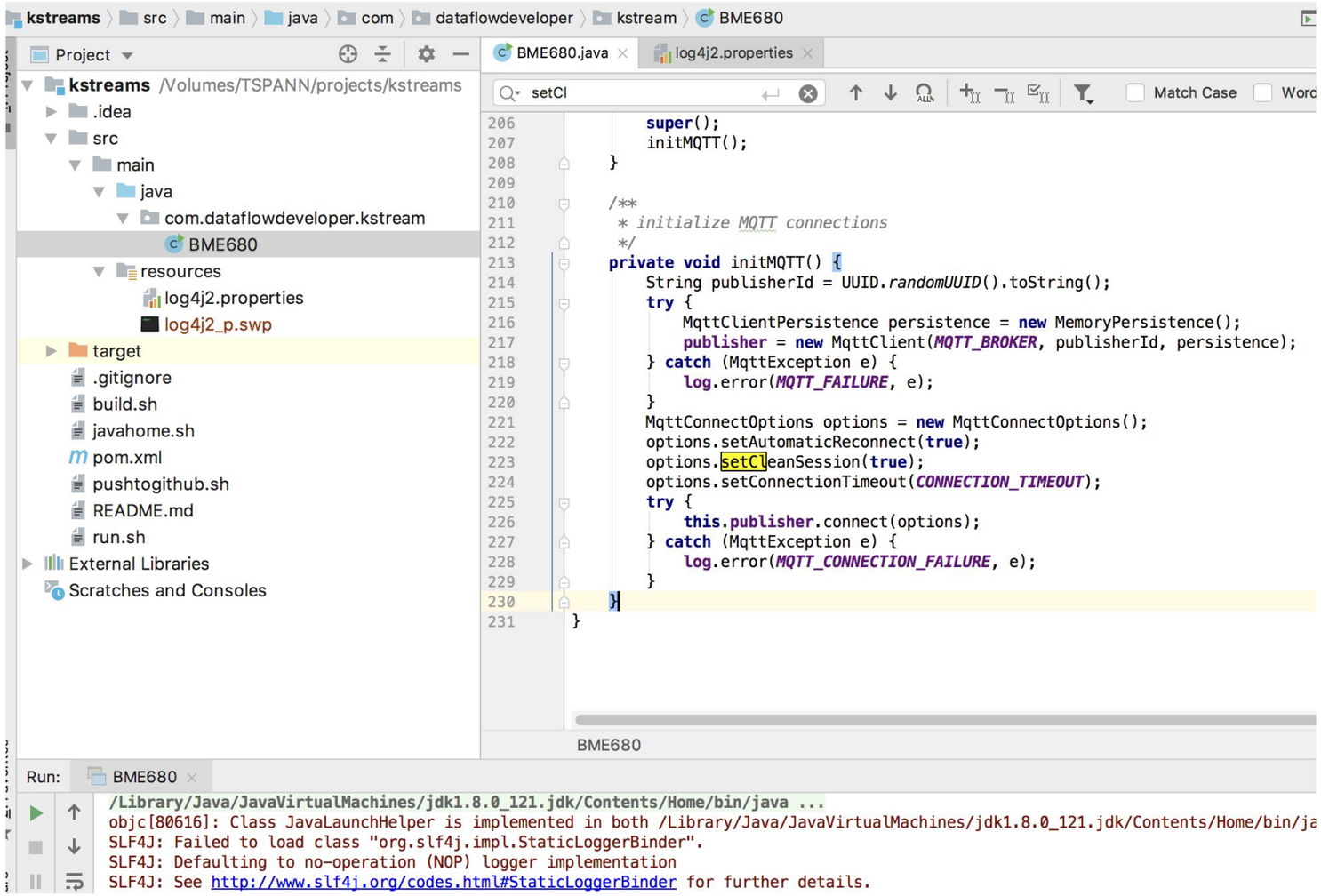
Kafka Streams

```
Runtime.getRuntime().addShutdownHook(new Thread(STREAMS_SHUTDOWN_HOOK)  {
        @Override
        public void run() {
            log.error(SHUTDOWN);
            streams.close();
            latch.countDown();
        }
    });

    try {
        streams.start();
        latch.await();
    } catch (Throwable e) {
        System.exit(1);
    }
```

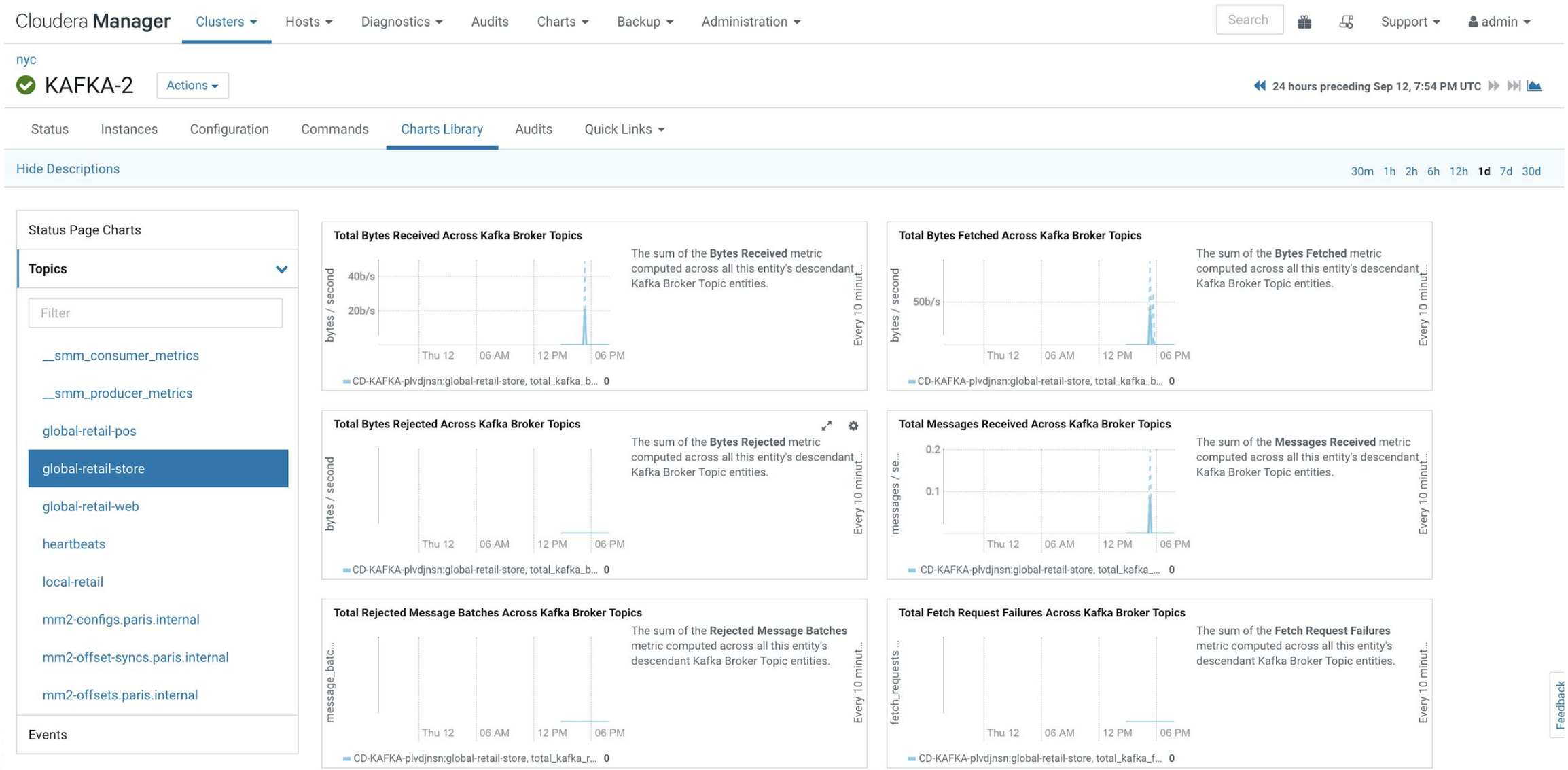
https://github.com/tspannhw/kstreams

Kafka Streams Example 2



https://github.com/tspannhw/kstreams

Cloudera Managed Kafka



CLOUDERA

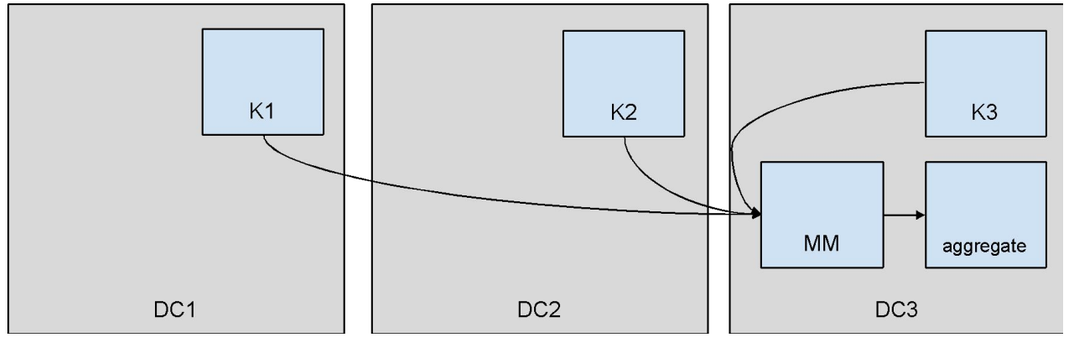
Kafka Replication

Kafka Replication Use Cases

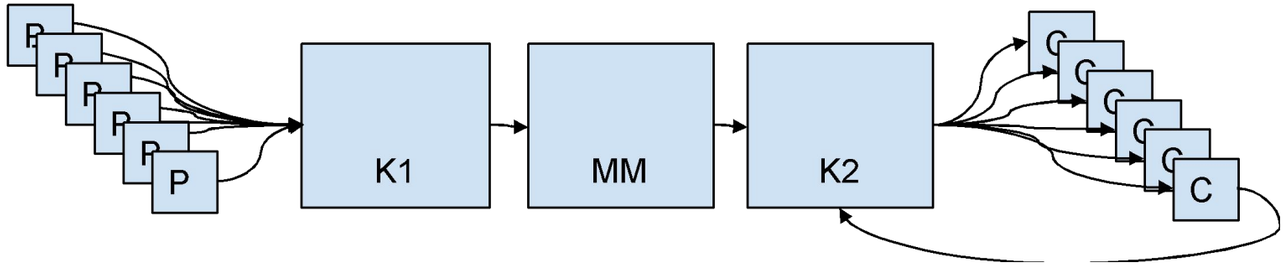
Disaster Recovery	Geo-Locality	Data Movement / Deployment
In an event of a partial or complete datacenter disaster, providing failover/failback to a secondary cluster in a different region / DC	Active-active geo-localized deployments allows users to access a near-by data center to optimize their architecture for low latency and high performance.	Use Kafka to synchronize data between on-prem applications and cloud deployments
Centralized Analytics	**Workload Isolation**	**Legal / Compliance**
Aggregate data from multiple Kafka clusters into one location for organization-wide analytics	Creation of different envs for SDLC: Dev, Test, Prod. Clusters for specific use case cases (ETL, ingestion, analytics, etc)	Since different regions have different data storage and security requirements, clusters need to be created in region but data still needs to be shared.

Various Replication Deployments Based on Use Cases

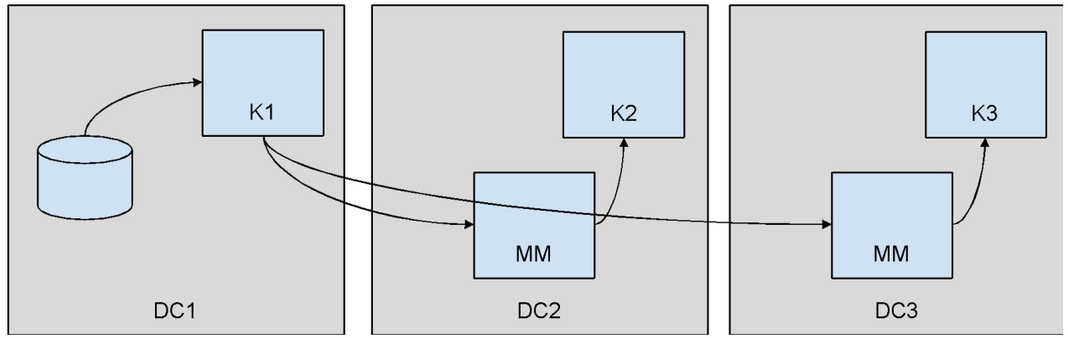
Aggregation



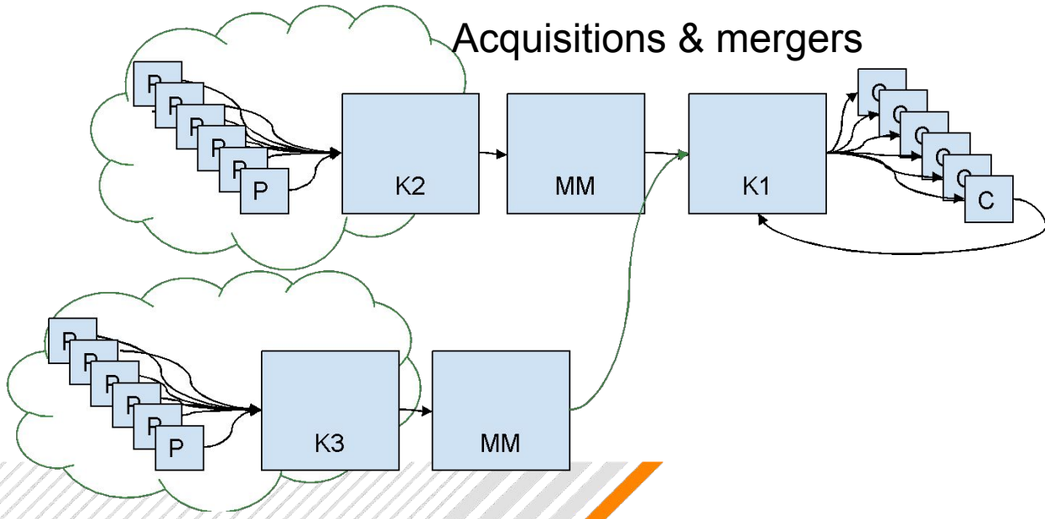
Segmentation



Data Deployment



Acquisitions & mergers



Legacy MirrorMaker

Limitations with Legacy MirrorMaker(MM) for Replication Use Cases

Challenge / Limitation	Description
Static Whitelists and Blacklists	Changes to replication jobs require restart of MirrorMaker cluster.
No Configuration/Metadata Synch	Topic configuration changes in the origin cluster are not detected and propagated to the destination cluster. New Topics are not detected
Scalability and Throughput Limitations due to Rebalances	Cannot scale replication processes as Kafka traffic increases
Lack of Monitoring and Operational Support	No tooling to to install and manage replication cluster. Difficult to monitor replication lag, consumer and producer metrics for replication workflows
No Disaster Recovery, Migration, Failover	MM doesn't support active/active without capabilities such as topic renaming, etc. No support for client failover/failover out of the box.