Integration Technical Conference 2019

M15 Event-Driven Enterprise using IBM Event Streams

Apache Kafka for the Enterprise

Andrew Schofield
Senior Technical Staff Member
IBM Messaging

IBM Cloud

IBM

Please note

What is Event Streaming?

Event-Driven in Action

Getting data to where it's needed, before it's needed

Respond to events before the moment passes

Responsive & personalized customer experiences

Bring real-time intelligence to your apps

Isn’t This “Just Messaging”?

Message Queuing & Event Streaming focus on different aspects of Messaging

Operations

Messages that represent some **current or future processing**. For instance: request and response messages.

Message Queuing

Events

Messages that represent the **state of the system**. For instance: logging, measurements and notifications.

Event Streaming

Event Streaming & Message Queuing Need Different Capabilities

Event Streaming

Stream history

Scalable consumption

Immutable data

Message Queuing

Transient data persistence

Conversational messaging

Assured delivery

What is Apache Kafka?

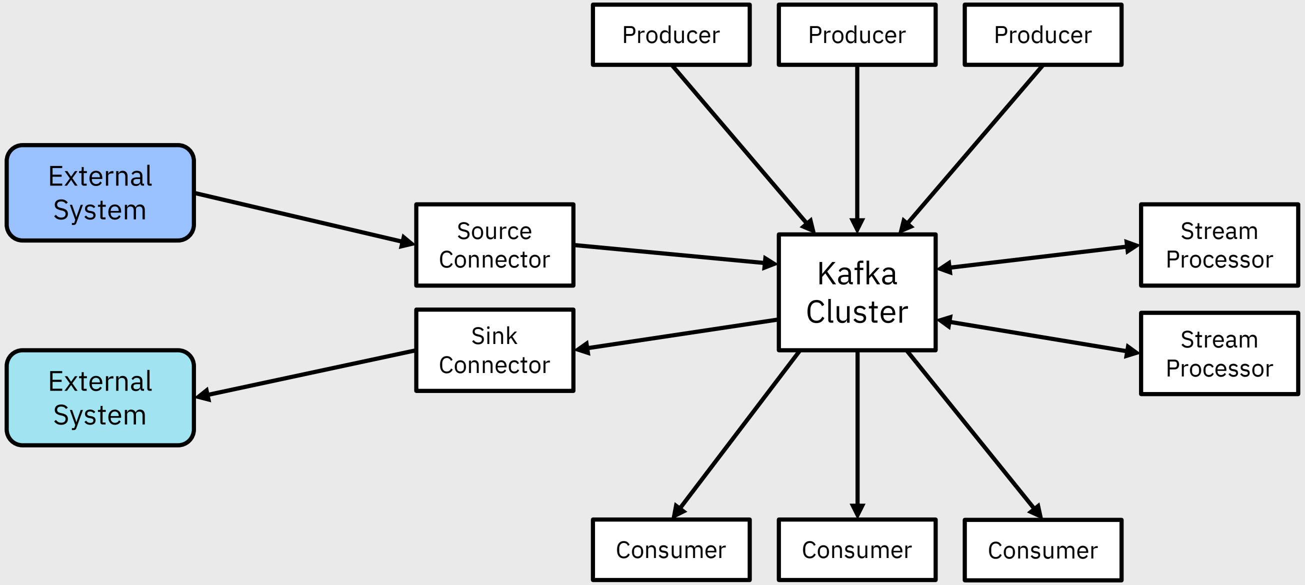
Apache Kafka is an **open-source, distributed streaming platform**

Publish and subscribe to streams of events

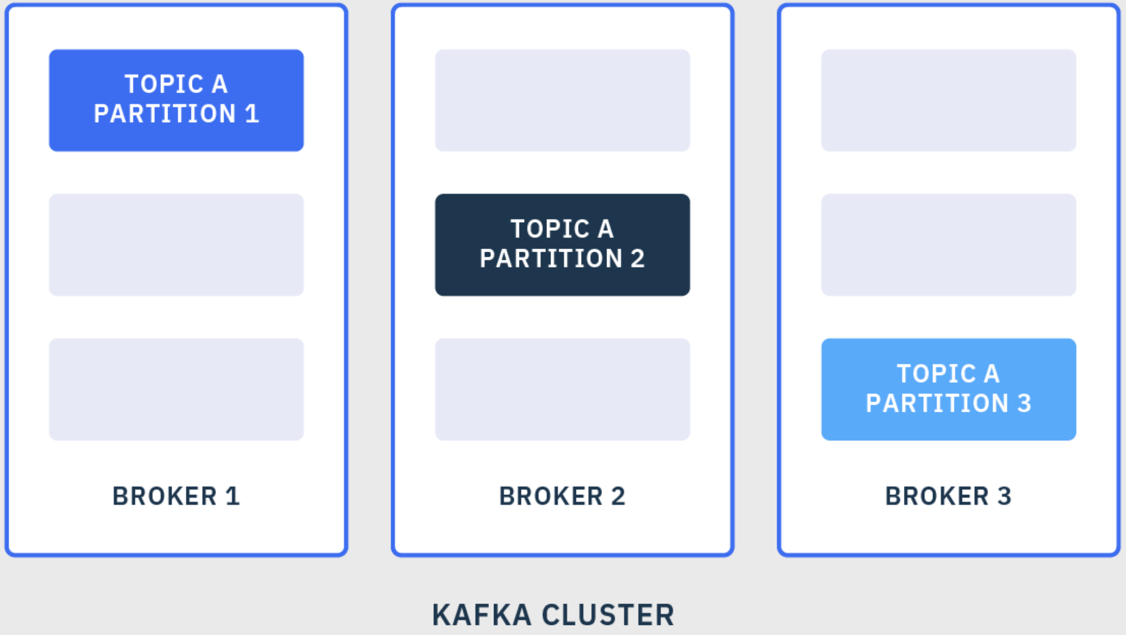
Store events in durable way

Process streams of events as they occur

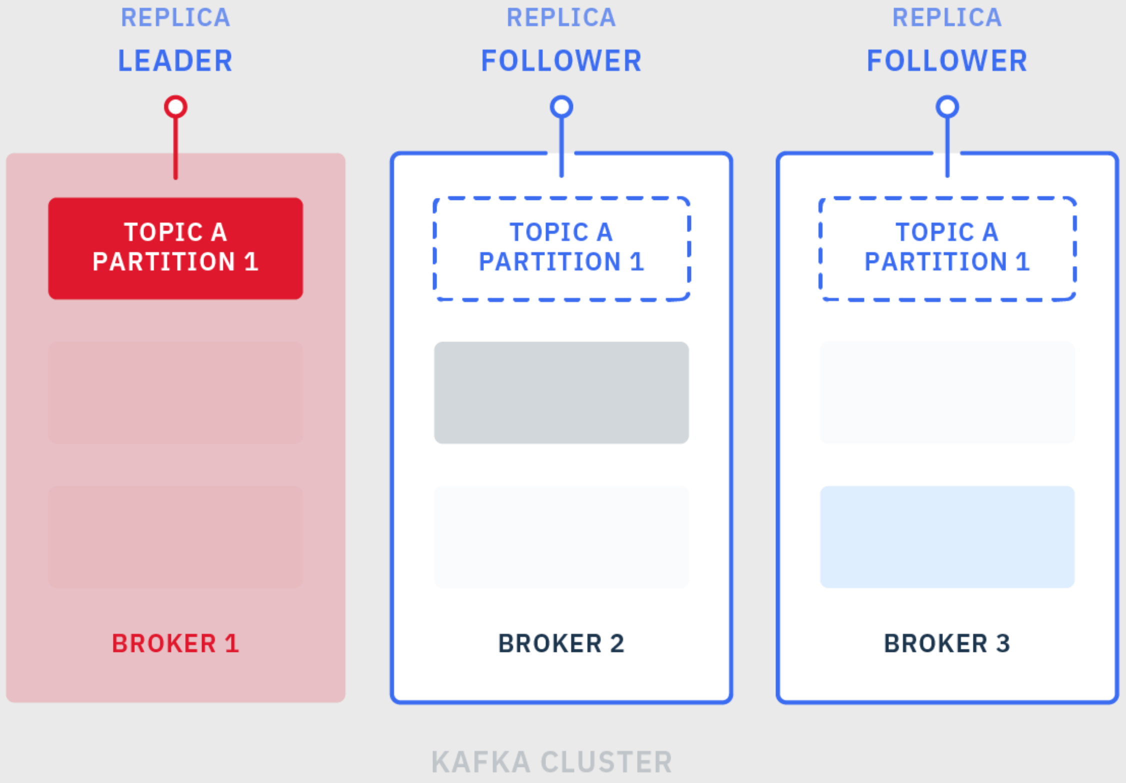
Apache Kafka is an Open-Source Streaming Platform



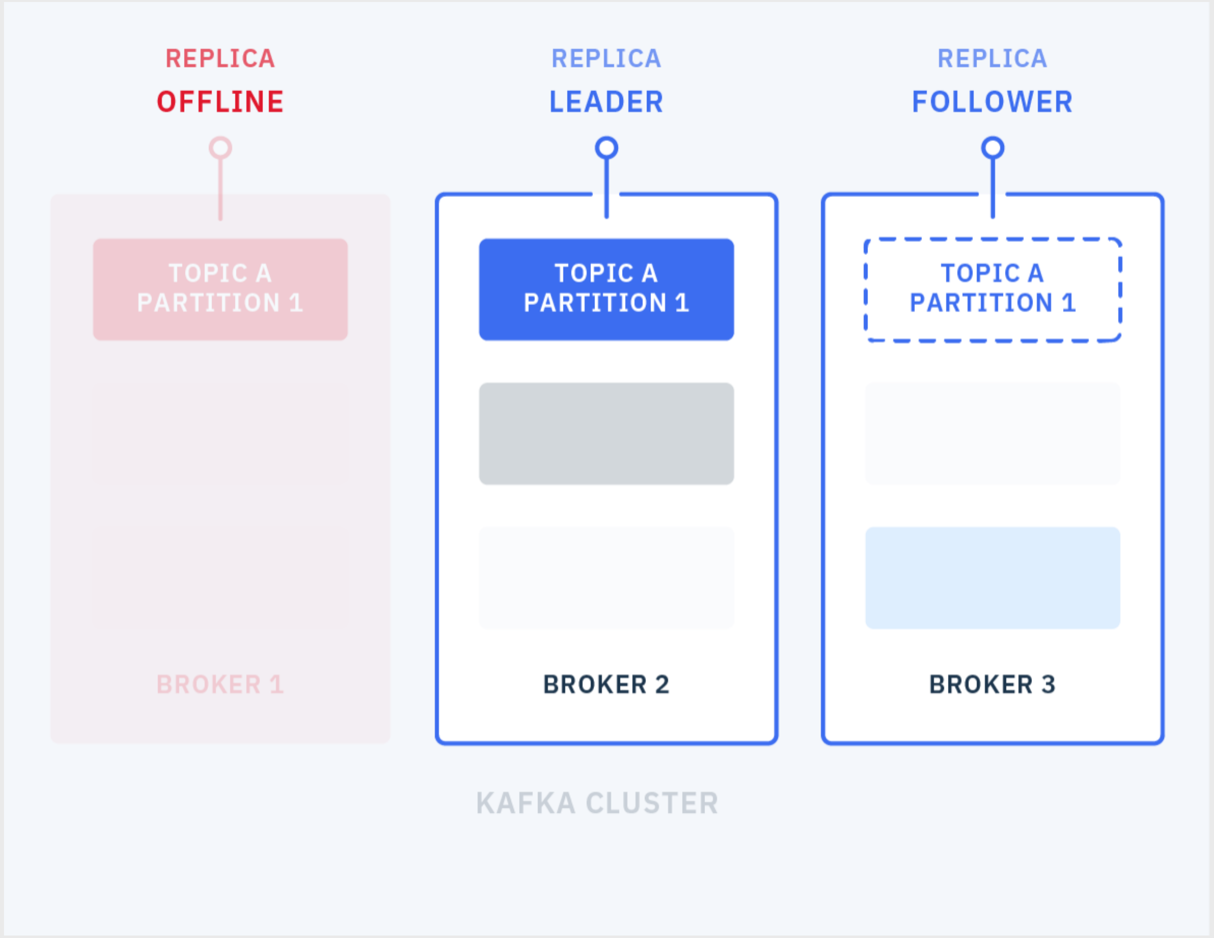
Kafka cluster



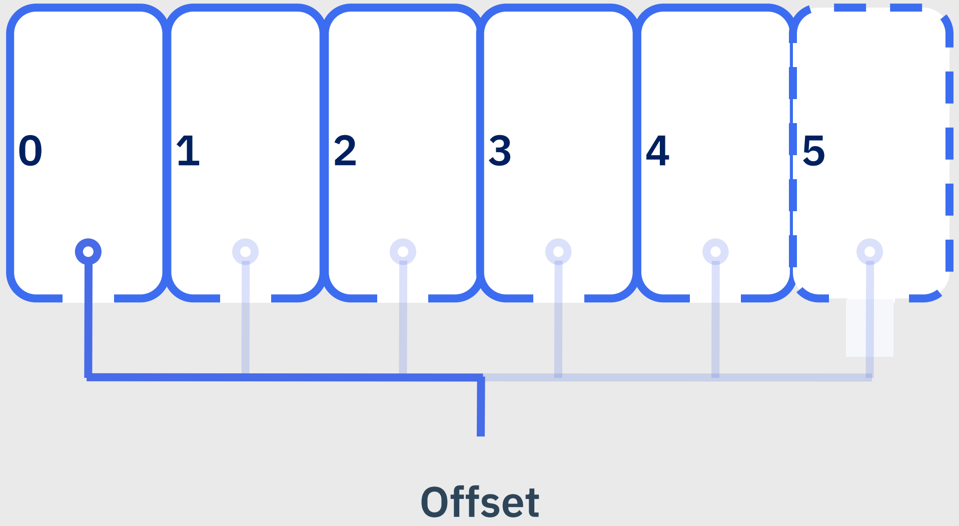
Replication



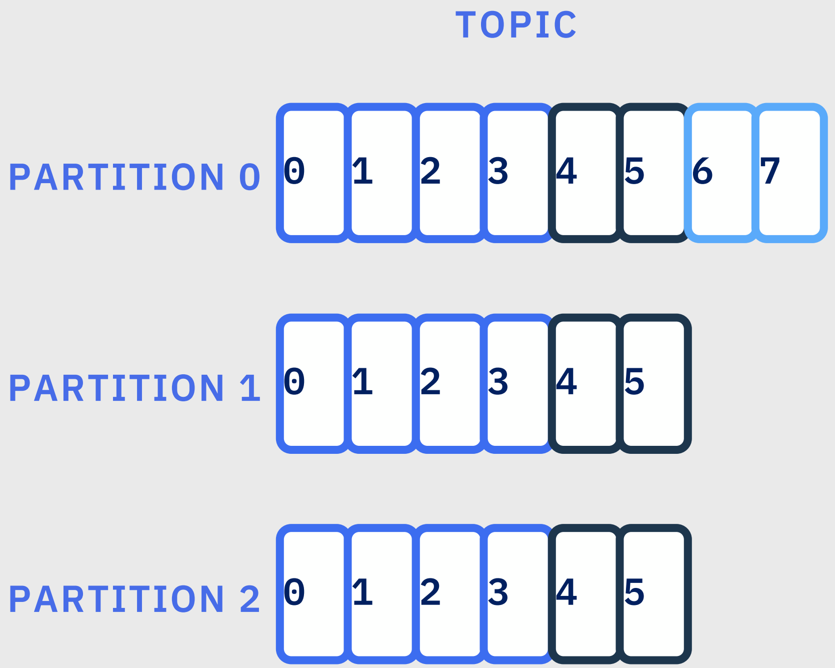
Replication



Topics



Producers



Producer can choose acknowledgement level:

`0`	Fire-and-forget *Fast, but risky*
`1`	Waits for 1 broker to acknowledge
`ALL`	Waits for all replica brokers to acknowledge

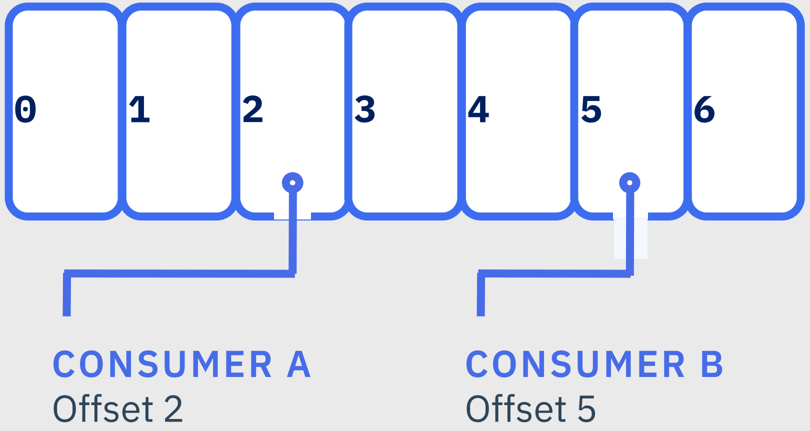
Producer can choose whether to retry:

`0`	Do not retry *Loses messages on error*
`>0`	Retry *Retry, might result in duplicates on error*

Producer can also choose idempotence

Can retry without risking duplicates

Consumers



Consumer can choose how to commit offsets:

`Automatic`	Commits might go faster than processing
`Manual, asynchronous`	Fairly safe, but could re-process messages
`Manual, synchronous`	Safe, but slows down processing

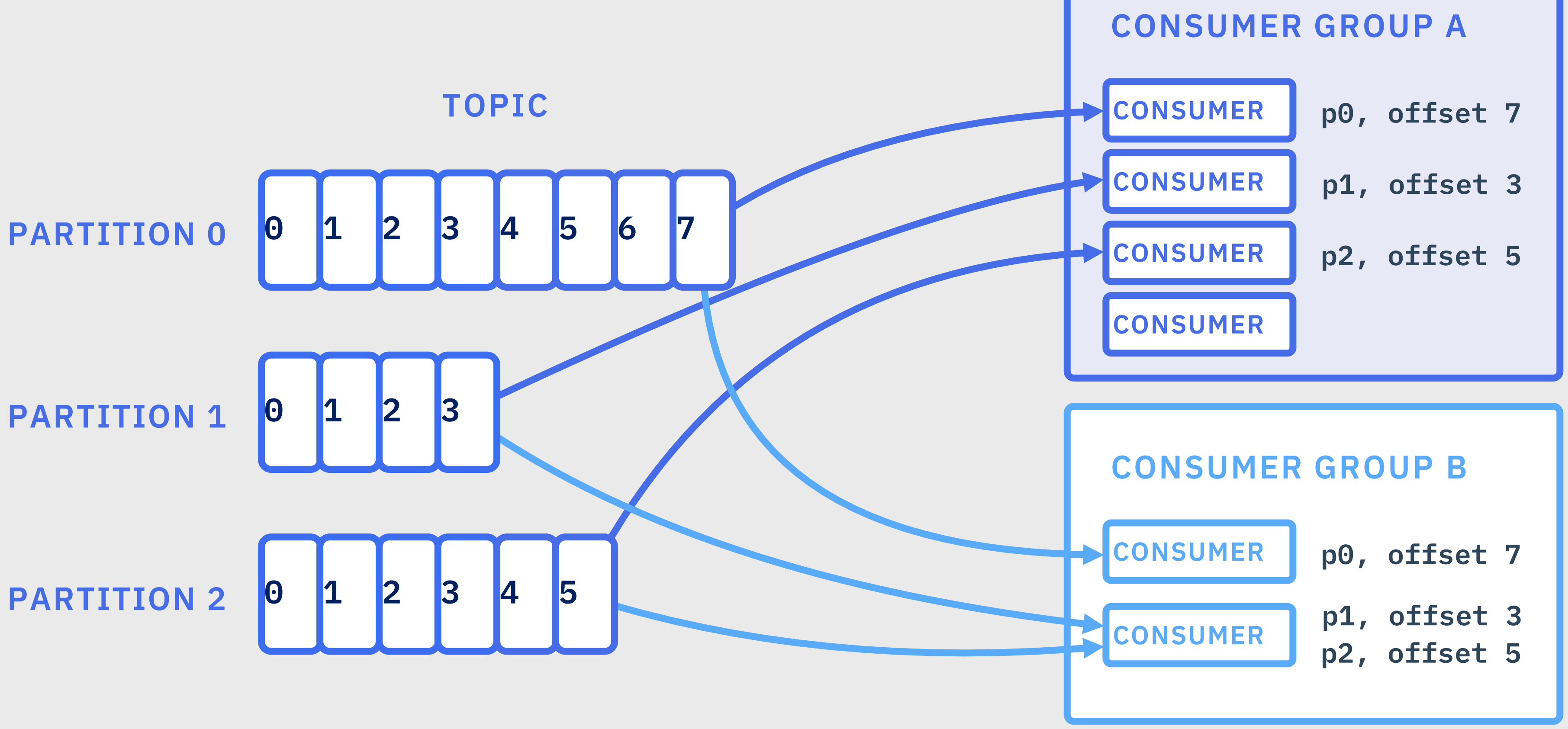
A common pattern is to commit offsets on a timer

Exactly once semantics

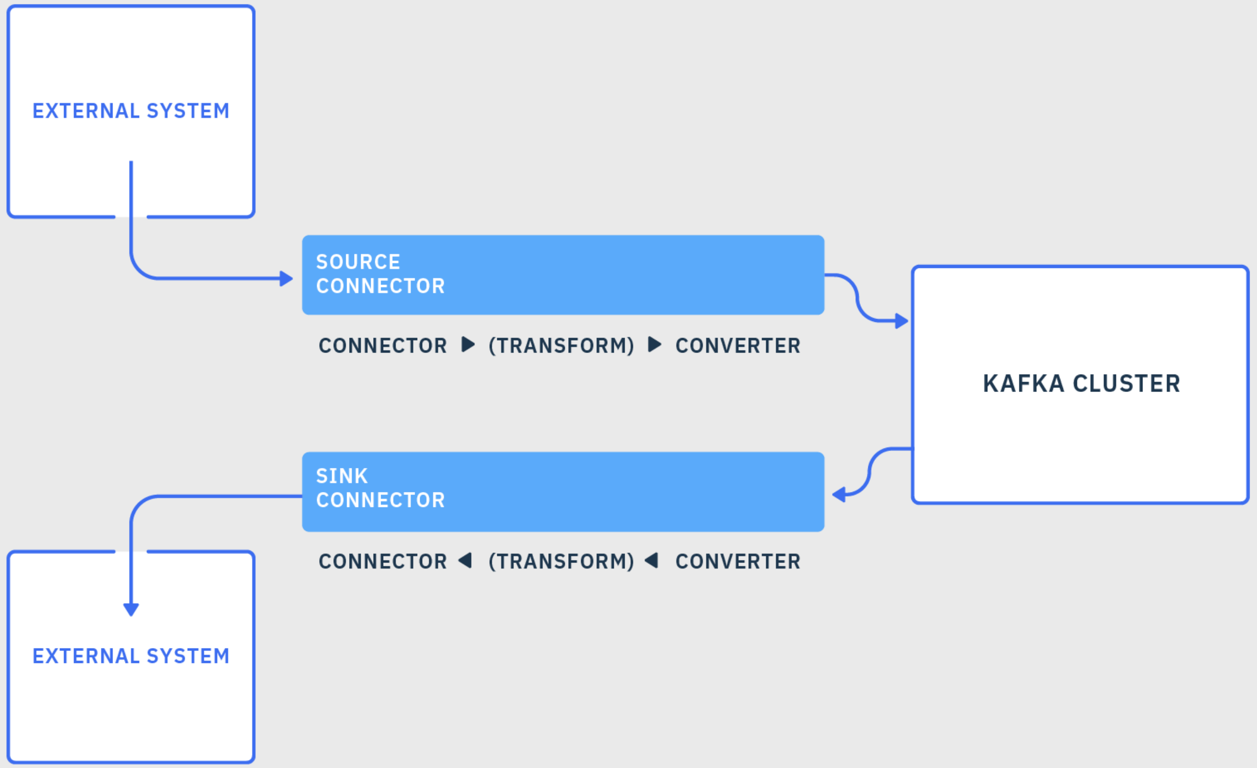
Can group sending messages and committing offsets into transactions

Primarily aimed at stream processing applications

Consumer Groups



Kafka Connect



Over 80 connectors

HDFS

Elasticsearch

MySQL

JDBC

IBM MQ

MQTT

CoAP

+ many others

What is IBM Event Streams?

IBM Event Streams

Fully supported Apache Kafka® with value-add capabilities

APACHE kafka®
A distributed streaming platform

IBM Event Streams Delivers Differentiated Value

IBM offers **Event Streams** in several form factors:

Private Cloud

Container–native Software

Red Hat OpenShift — X86_64

IBM Cloud Private — X86_64 & **zLinux**

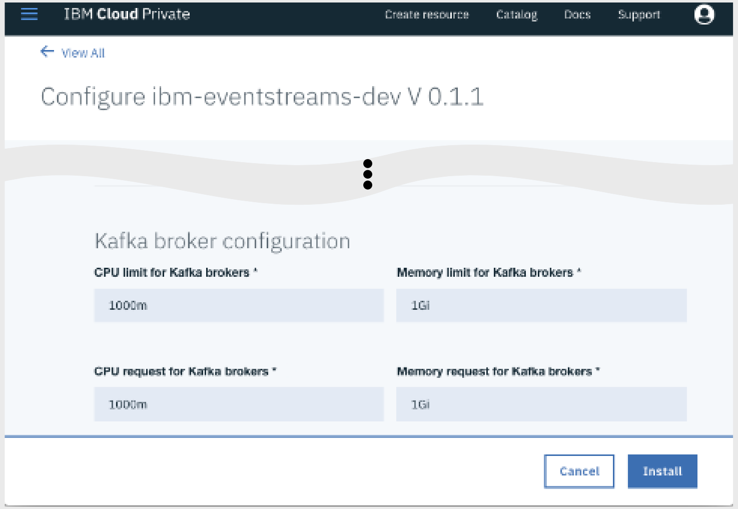
IBM Public Cloud

Isolated aaS

Multi-tenant aaS

- **IBM has years of operational expertise** running Apache Kafka for Enterprises
    - This experience has been embedded in the DNA of Event Streams
- Event Streams makes Kafka easy to run, manage & consume, **reducing skill requirements** and increasing speed of deployment for **faster time to value**
- IBM Cloud Private security integration **simplifies Kafka access control** using roles and policies
- **IBM's experience in enterprise-critical software** has shaped features like geo-replication for Disaster Recovery & integration with IBM MQ, to give confidence deploying **mission-critical workloads**
- **Support you can trust** – IBM has decades of experience supporting the World's toughest environments

Making Apache Kafka intuitive and easy

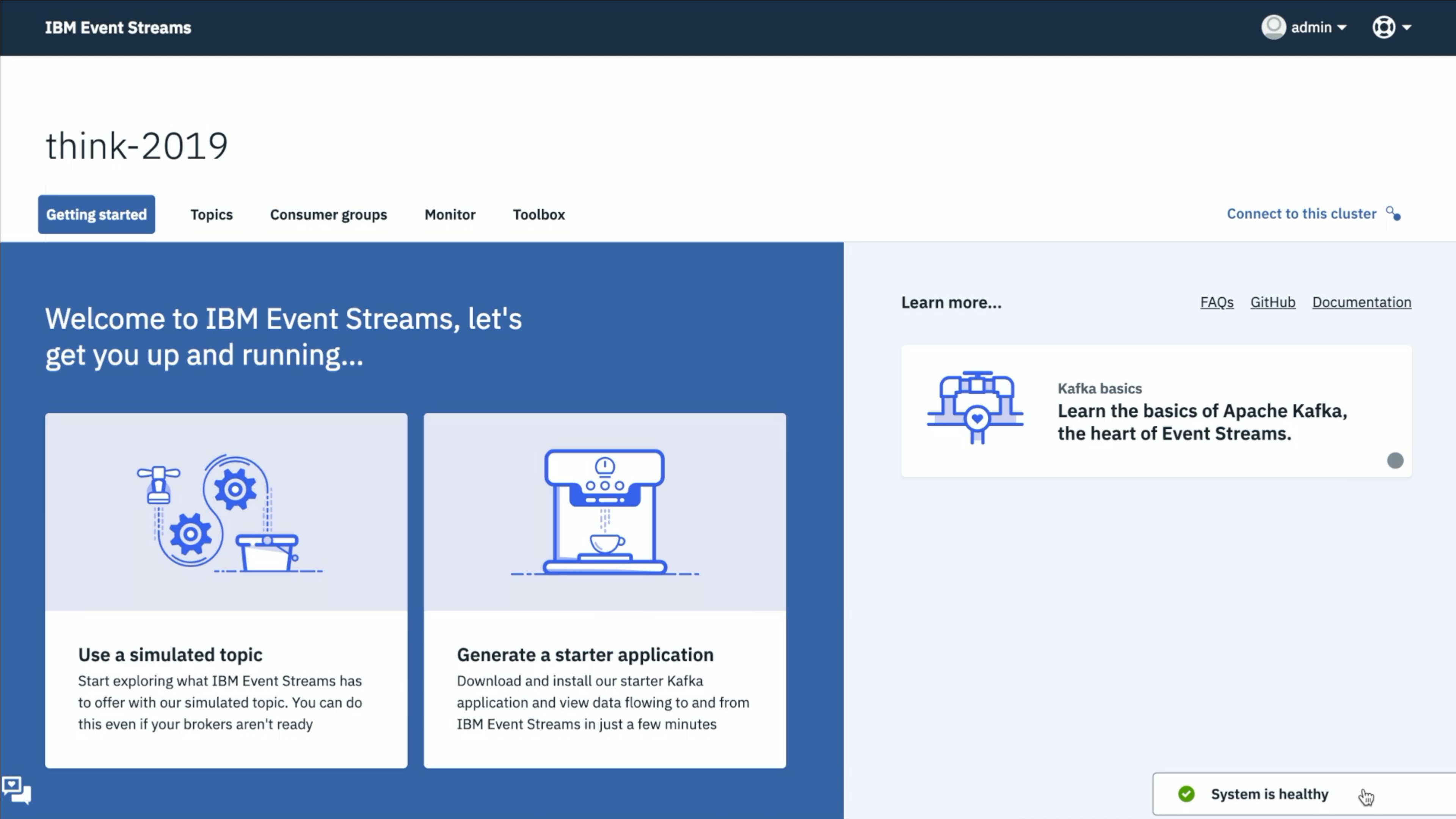


Configure ibm-eventstreams-dev V 0.1.1

Kafka broker configuration

CPU limit for Kafka brokers *	Memory limit for Kafka brokers *
1000m	1Gi
CPU request for Kafka brokers *	Memory request for Kafka brokers *
1000m	1Gi

- Many distinct components to deploy, configure and coordinate secure connectivity
- Container placement critical to ensure production-level availability
- Secured network traffic ingress
- Ensuring consistent and repeatable deployment



IBM Event Streams

admin

think-2019

Getting started

Topics

Consumer groups

Monitor

Toolbox

Connect to this cluster

Welcome to IBM Event Streams, let's get you up and running...

Use a simulated topic

Start exploring what IBM Event Streams has to offer with our simulated topic. You can do this even if your brokers aren't ready

Generate a starter application

Download and install our starter Kafka application and view data flowing to and from IBM Event Streams in just a few minutes

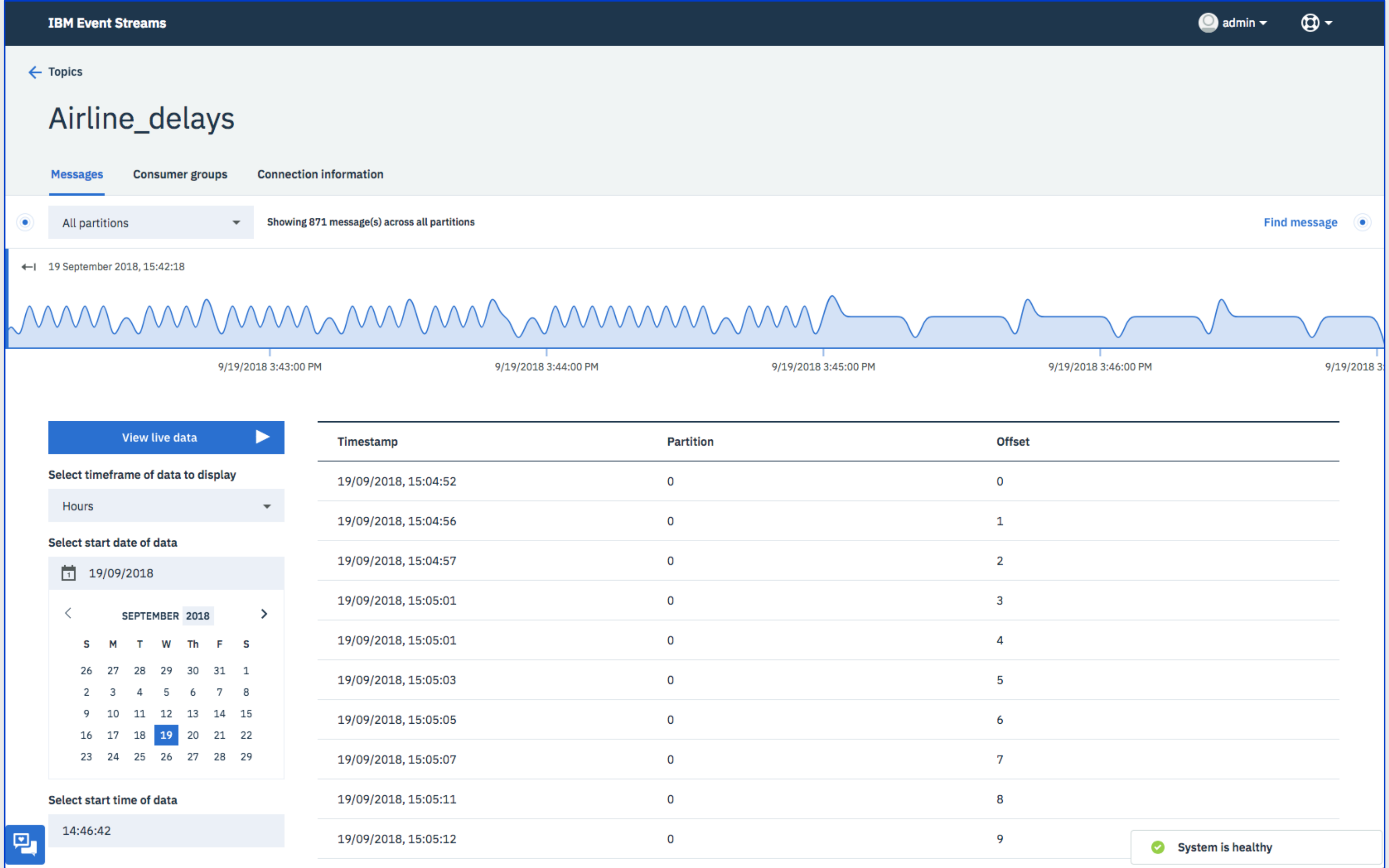
Learn more...

FAQs GitHub Documentation

Kafka basics

Learn the basics of Apache Kafka, the heart of Event Streams.

System is healthy



IBM Event Streams

admin

← Topics

Airline_delays

Messages | Consumer groups | Connection information

All partitions

Showing 871 message(s) across all partitions

Find message

19 September 2018, 15:42:18

View live data

Select timeframe of data to display

Hours

Select start date of data

19/09/2018

SEPTEMBER 2018

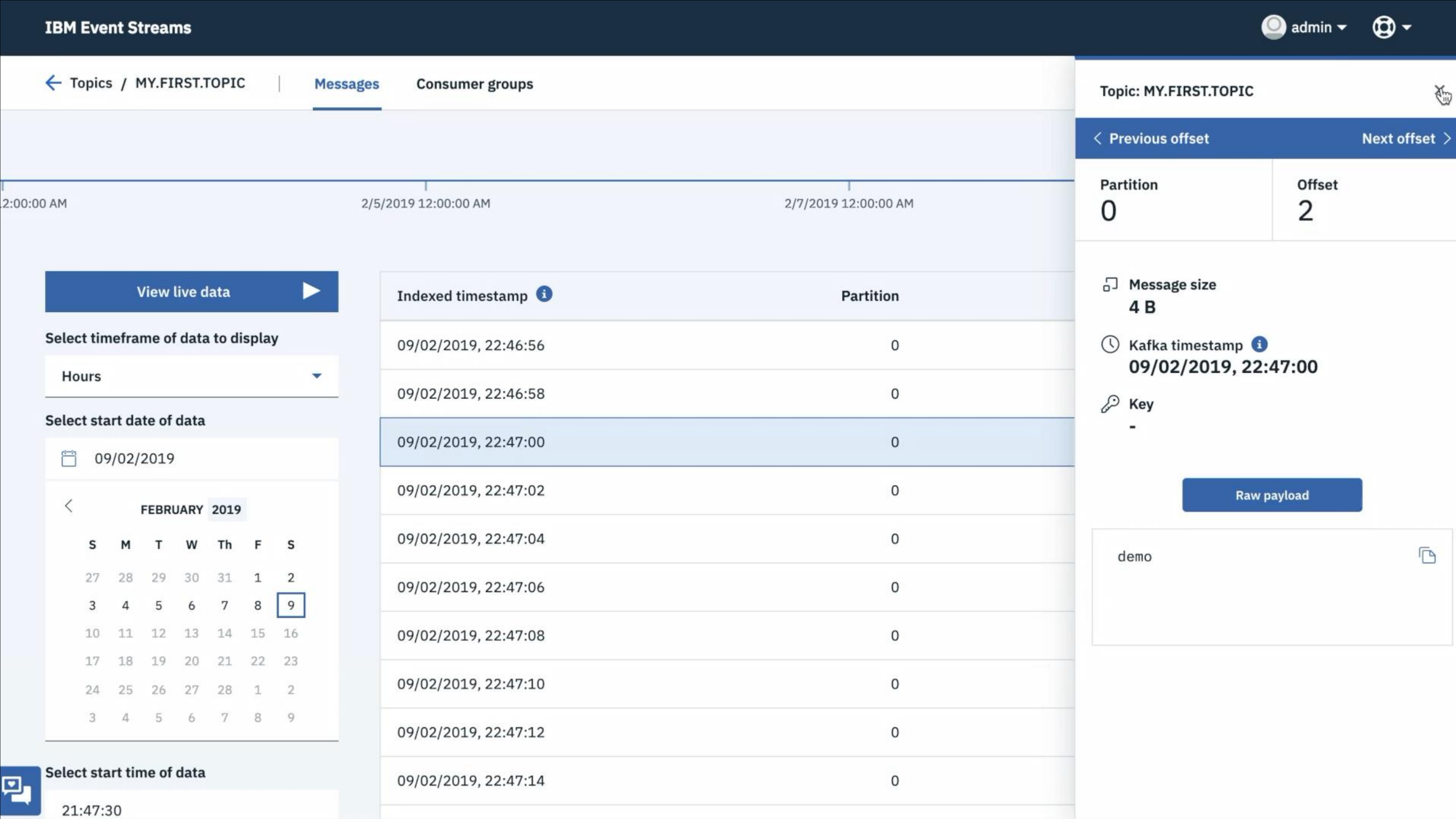
S	M	T	W	Th	F	S
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

Select start time of data

14:46:42

Timestamp	Partition	Offset
19/09/2018, 15:04:52	0	0
19/09/2018, 15:04:56	0	1
19/09/2018, 15:04:57	0	2
19/09/2018, 15:05:01	0	3
19/09/2018, 15:05:01	0	4
19/09/2018, 15:05:03	0	5
19/09/2018, 15:05:05	0	6
19/09/2018, 15:05:07	0	7
19/09/2018, 15:05:11	0	8
19/09/2018, 15:05:12	0	9

System is healthy



IBM Event Streams

admin

Topics / MY.FIRST.TOPIC

Messages

Consumer groups

2:00:00 AM

2/5/2019 12:00:00 AM

2/7/2019 12:00:00 AM

View live data

Select timeframe of data to display

Hours

Select start date of data

09/02/2019

FEBRUARY 2019

S	M	T	W	Th	F	S
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2
3	4	5	6	7	8	9

Select start time of data

21:47:30

Indexed timestamp	Partition
09/02/2019, 22:46:56	0
09/02/2019, 22:46:58	0
09/02/2019, 22:47:00	0
09/02/2019, 22:47:02	0
09/02/2019, 22:47:04	0
09/02/2019, 22:47:06	0
09/02/2019, 22:47:08	0
09/02/2019, 22:47:10	0
09/02/2019, 22:47:12	0
09/02/2019, 22:47:14	0

Topic: MY.FIRST.TOPIC

Previous offset

Next offset

Partition
0

Offset
2

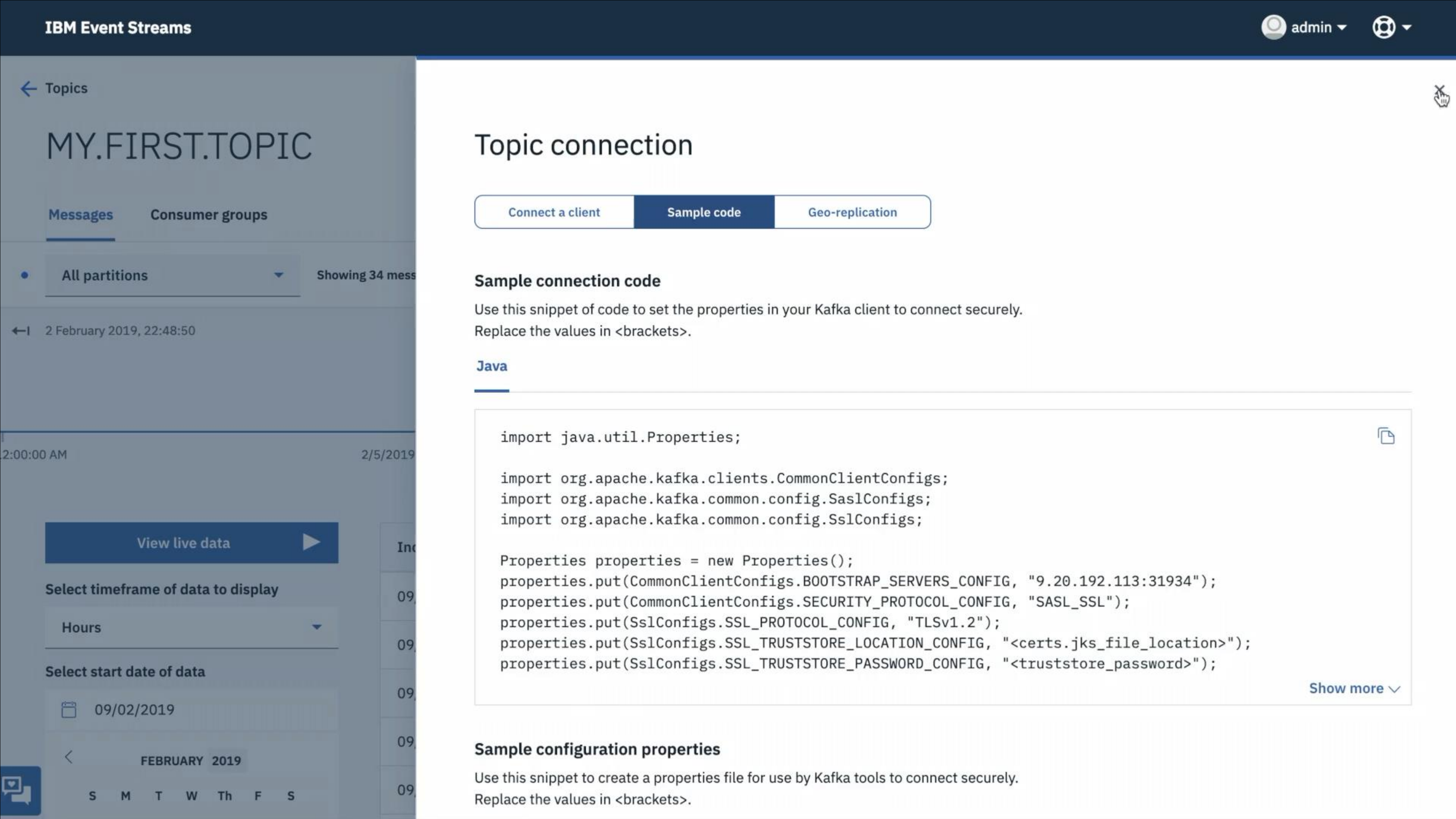
Message size
4 B

Kafka timestamp
09/02/2019, 22:47:00

Key
-

Raw payload

demo



Topic connection

Connect a client | Sample code | Geo-replication

Sample connection code

Use this snippet of code to set the properties in your Kafka client to connect securely.
Replace the values in <brackets>.

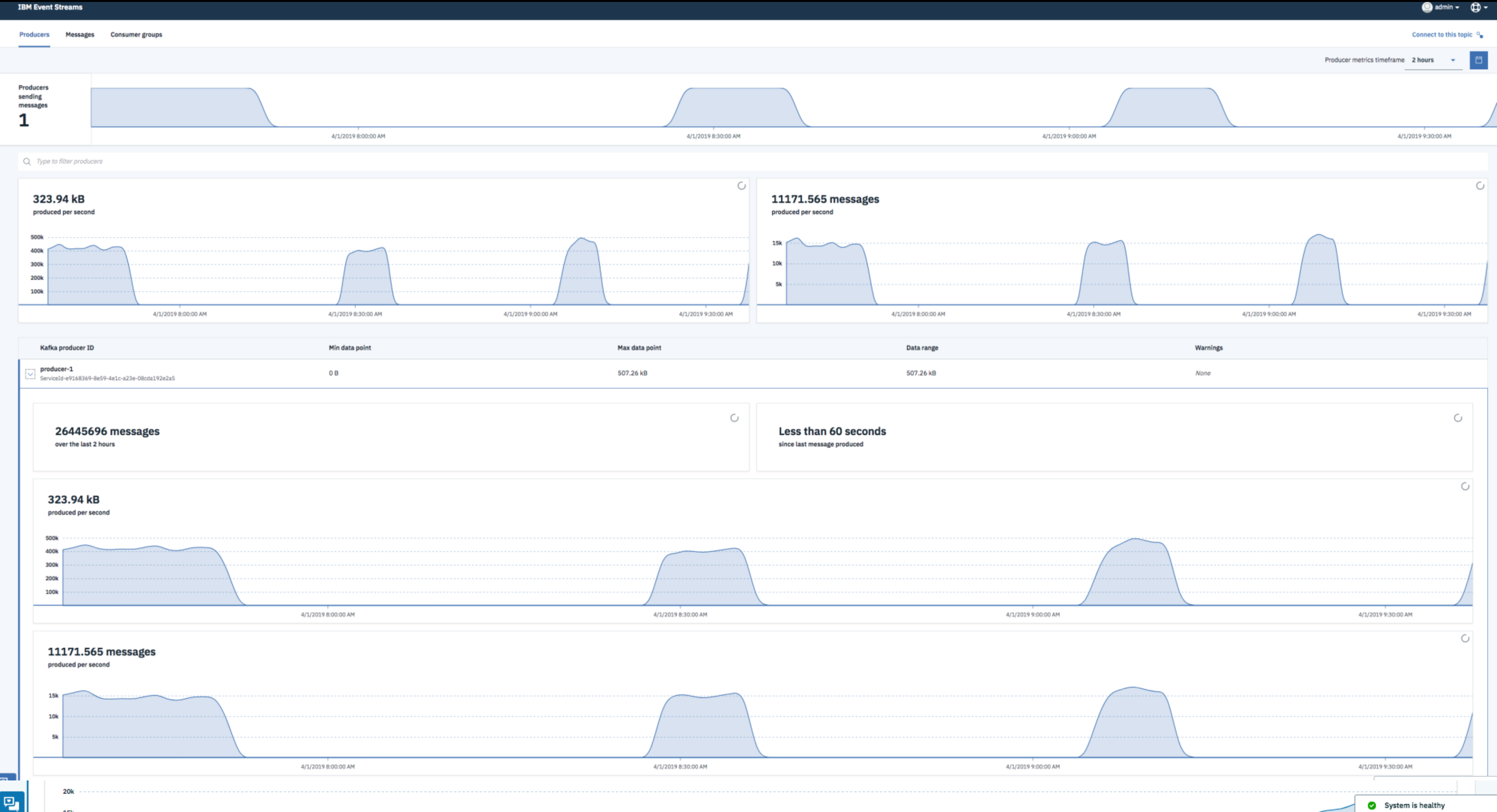
Java

```
import java.util.Properties;

import org.apache.kafka.clients.CommonClientConfigs;
import org.apache.kafka.common.config.SaslConfigs;
import org.apache.kafka.common.config.SslConfigs;

Properties properties = new Properties();
properties.put(CommonClientConfigs.BOOTSTRAP_SERVERS_CONFIG, "9.20.192.113:31934");
properties.put(CommonClientConfigs.SECURITY_PROTOCOL_CONFIG, "SASL_SSL");
properties.put(SslConfigs.SSL_PROTOCOL_CONFIG, "TLSv1.2");
properties.put(SslConfigs.SSL_TRUSTSTORE_LOCATION_CONFIG, "<certs.jks_file_location>");
properties.put(SslConfigs.SSL_TRUSTSTORE_PASSWORD_CONFIG, "<truststore_password>");
```

Show more

Sample configuration properties

Use this snippet to create a properties file for use by Kafka tools to connect securely.
Replace the values in <brackets>.



Producers sending messages
1

323.94 kB
produced per second

11171.565 messages
produced per second

Kafka producer ID	Min data point	Max data point	Data range	Warnings
producer-1 ServiceId-e9168369-8e59-4e1c-a23e-08cda192e2a5	0 B	507.26 kB	507.26 kB	*None*

26445696 messages
over the last 2 hours

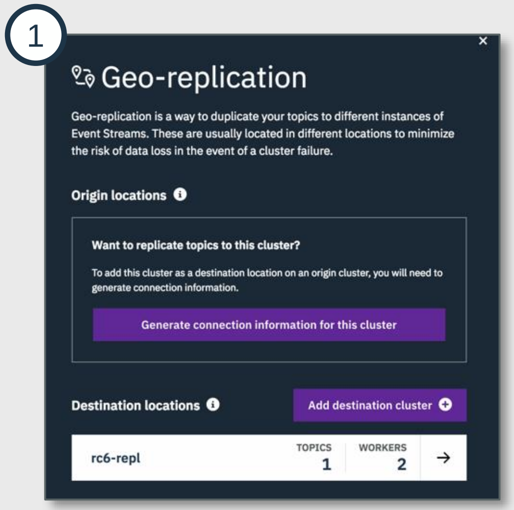
Less than 60 seconds
since last message produced

323.94 kB
produced per second

11171.565 messages
produced per second

Effortless geo-replication

1



Geo-replication

Geo-replication is a way to duplicate your topics to different instances of Event Streams. These are usually located in different locations to minimize the risk of data loss in the event of a cluster failure.

Origin locations

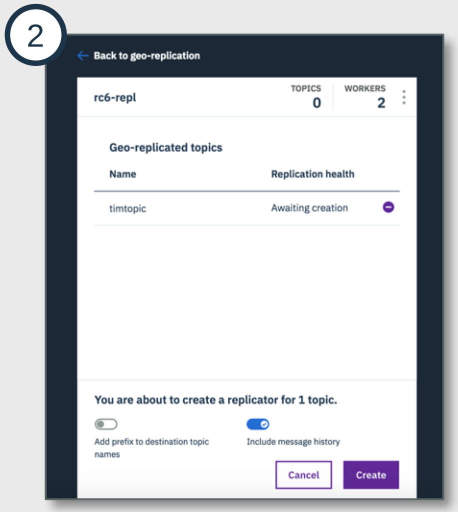
Want to replicate topics to this cluster?

To add this cluster as a destination location on an origin cluster, you will need to generate connection information.

Destination locations

	TOPICS	WORKERS
rc6-repl	1	2

2



	TOPICS	WORKERS
rc6-repl	0	2

Geo-replicated topics

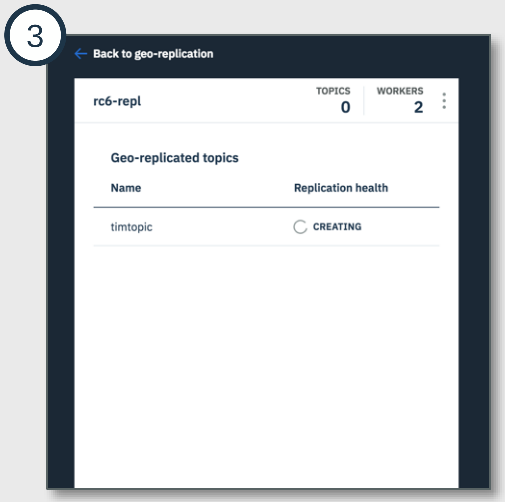
Name	Replication health
timtopic	Awaiting creation

You are about to create a replicator for 1 topic.

Add prefix to destination topic names

Include message history

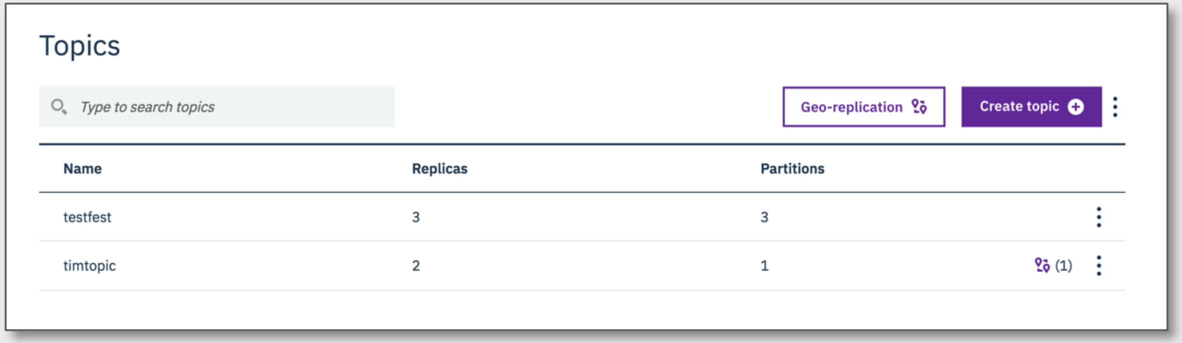
3



	TOPICS	WORKERS
rc6-repl	0	2

Geo-replicated topics

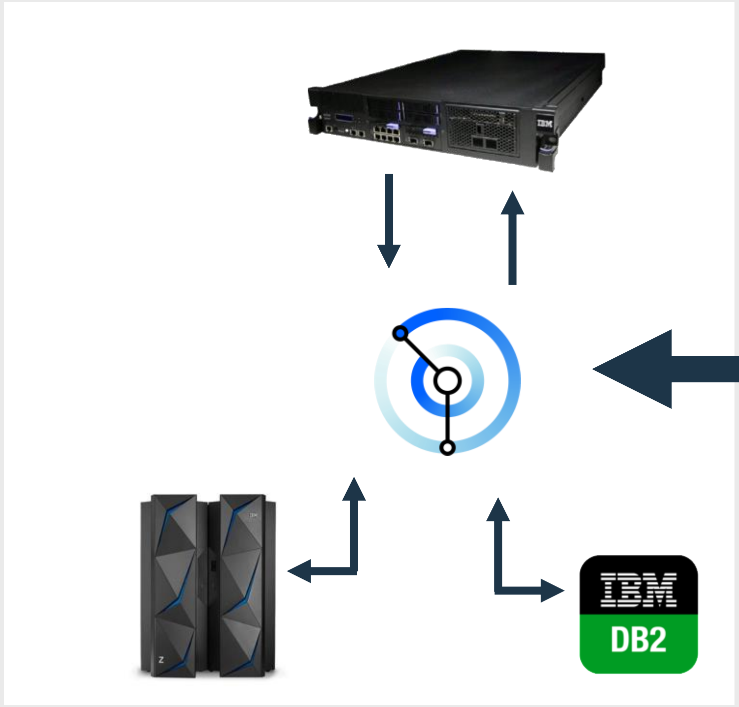
Name	Replication health
timtopic	CREATING

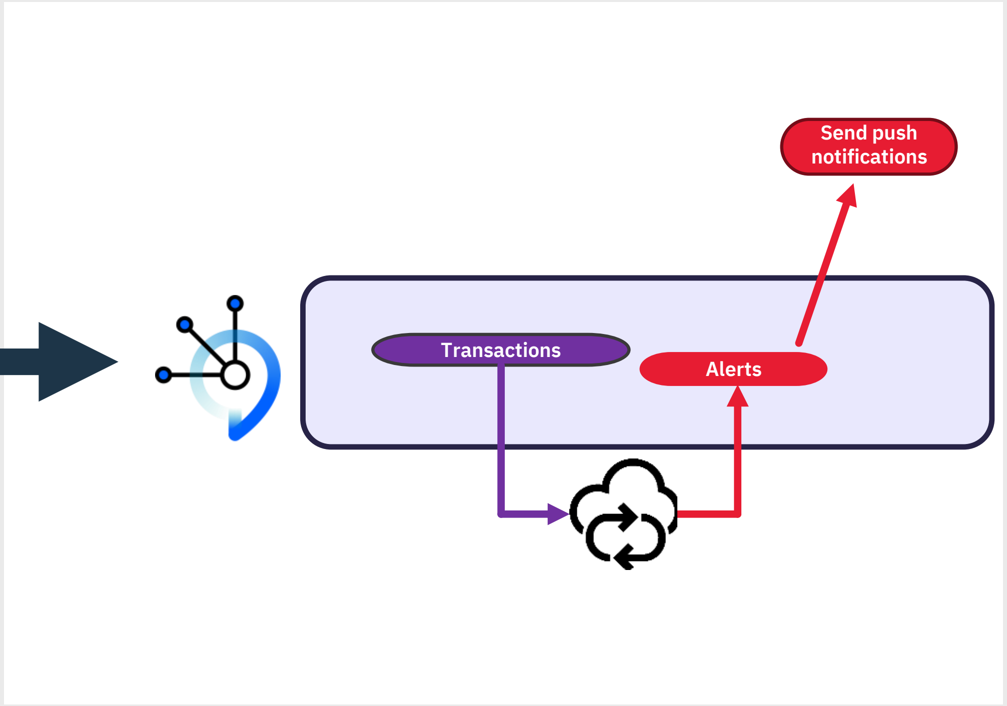


Topics

Name	Replicas	Partitions	
testfest	3	3	
timtopic	2	1	(1)

Integrates with IBM MQ





IBM MQ connects mission-critical Systems of Record, requiring **transactional, once-only delivery**

IBM Event Streams distributes and processes streams of events in real-time to intelligently engage with customers

It’s Easy to Connect IBM MQ to Apache Kafka

IBM has created a pair of connectors, available as source code or as part of IBM Event Streams

Source Connector

From MQ queue to Kafka topic

https://github.com/ibm-messaging/kafka-connect-mq-source

Sink Connector

From Kafka topic to MQ queue

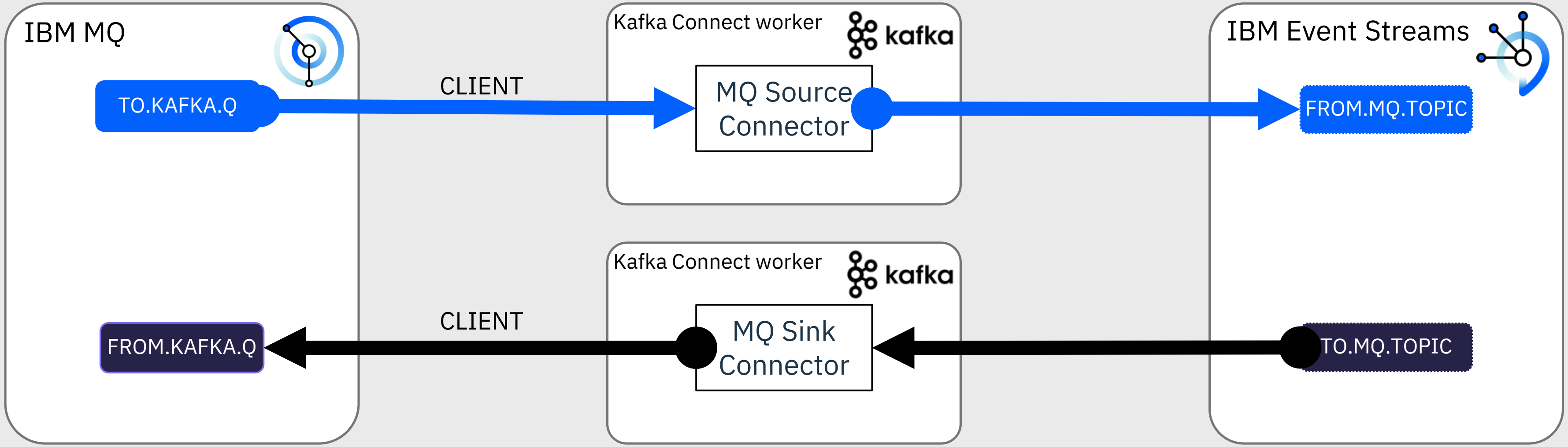
https://github.com/ibm-messaging/kafka-connect-mq-sink

- Copies messages from MQ queues to Event Streams topics and vice versa
- Supports all current MQ versions (MQ v8 or later, all platforms)
- Extend the connector to support any business-specific message format
- Fully supported by IBM for customers with support entitlement for IBM Event Streams

Running the Connectors for IBM MQ

The connectors are deployed into a component of Apache Kafka called a Kafka Connect worker

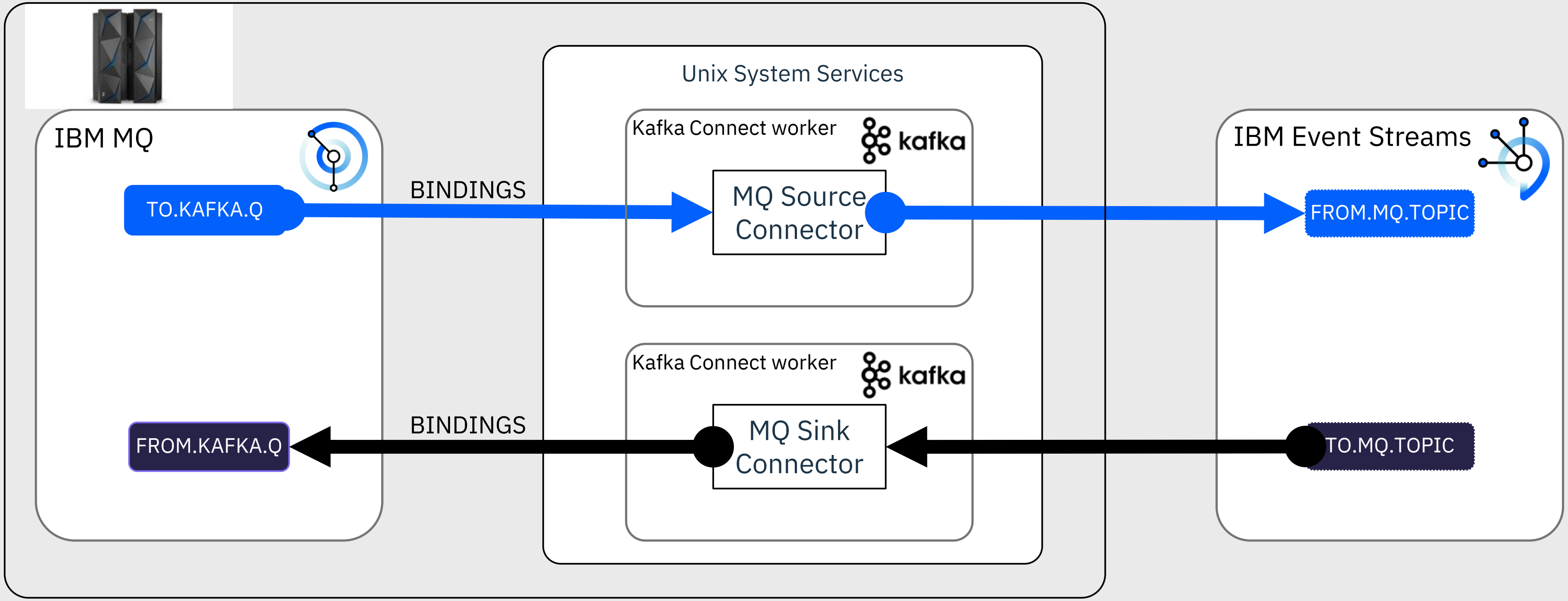
This runs between IBM MQ and IBM Event Streams (or open-source Apache Kafka)



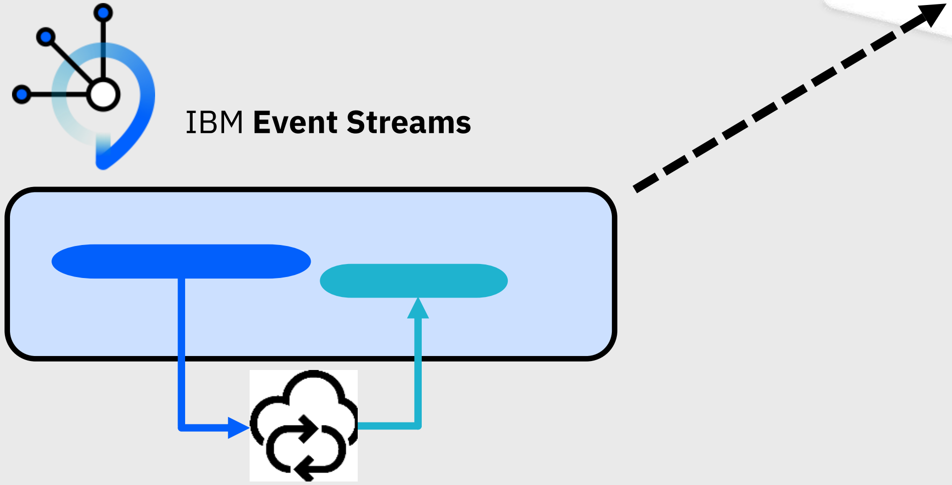
Running the Connectors for IBM MQ on z/OS

The Kafka Connect workers can be deployed onto z/OS Unix System Services

Then, the connection to MQ can be a bindings connection



IBM Event Streams | Integrated with Key Monitoring Tools



External monitoring tools
Datadog, Splunk, etc

Additional capabilities in 1Q19

IBM Event Streams 2019.1.1 | **GA**: 29th March

OpenShift support

What:

Deploy IBM Event Streams into existing OpenShift private cloud environments

Clients that use Red Hat OpenShift can now deploy Event Streams into the same platform as their applications

Kafka REST API

Scalable REST interface for inbound event data

Unlock events from systems that cannot connect easily to Kafka

→ REST connectivity helps Mainframe and DataPower users especially

Monitoring

Integration points for 3rd party monitoring tools

Connect existing monitoring tools to Event Streams to monitor it alongside the other components of their application

→ Get a single dashboard view of the environment

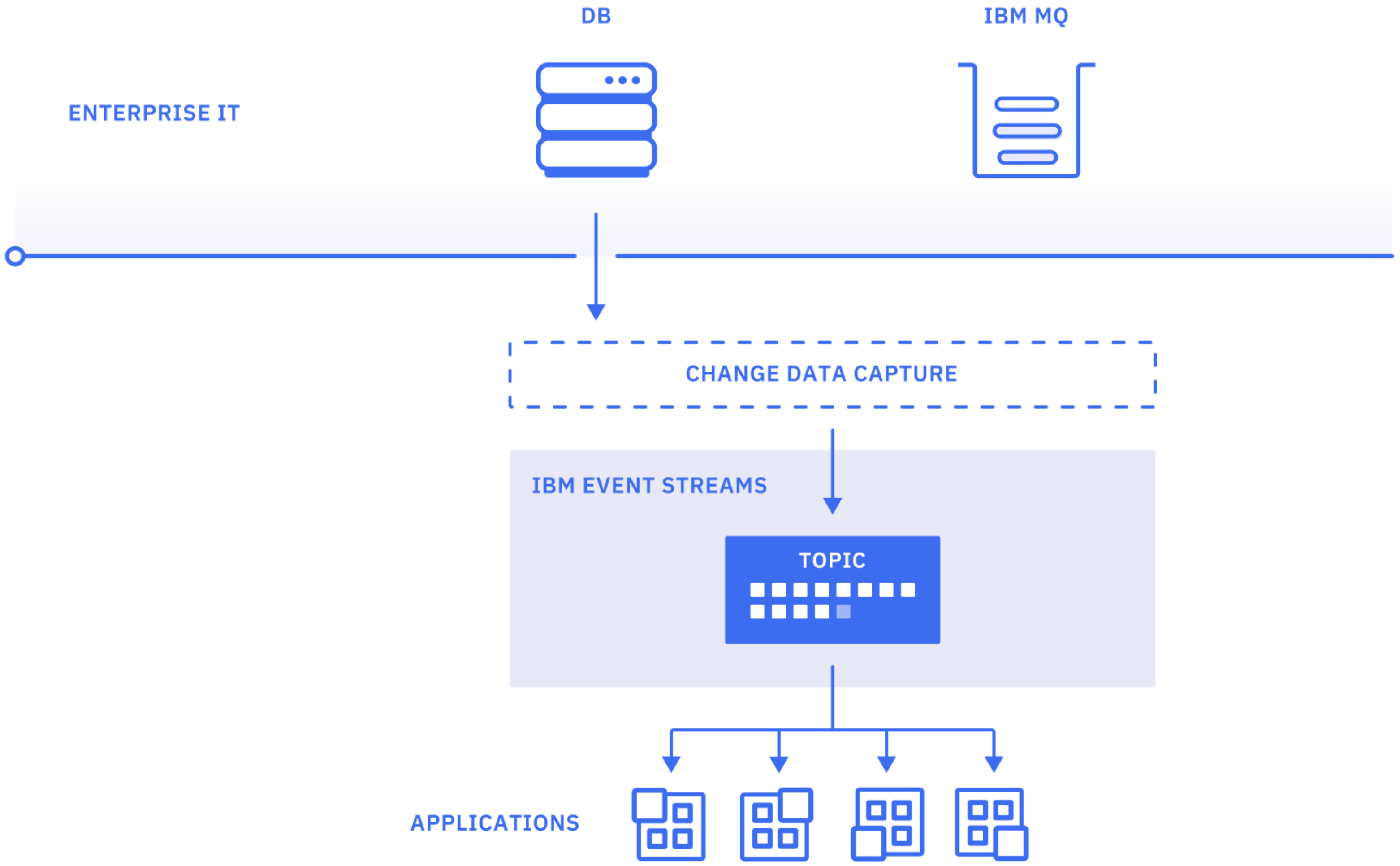
COMING SOON! (Statement of Direction: 1Q19)

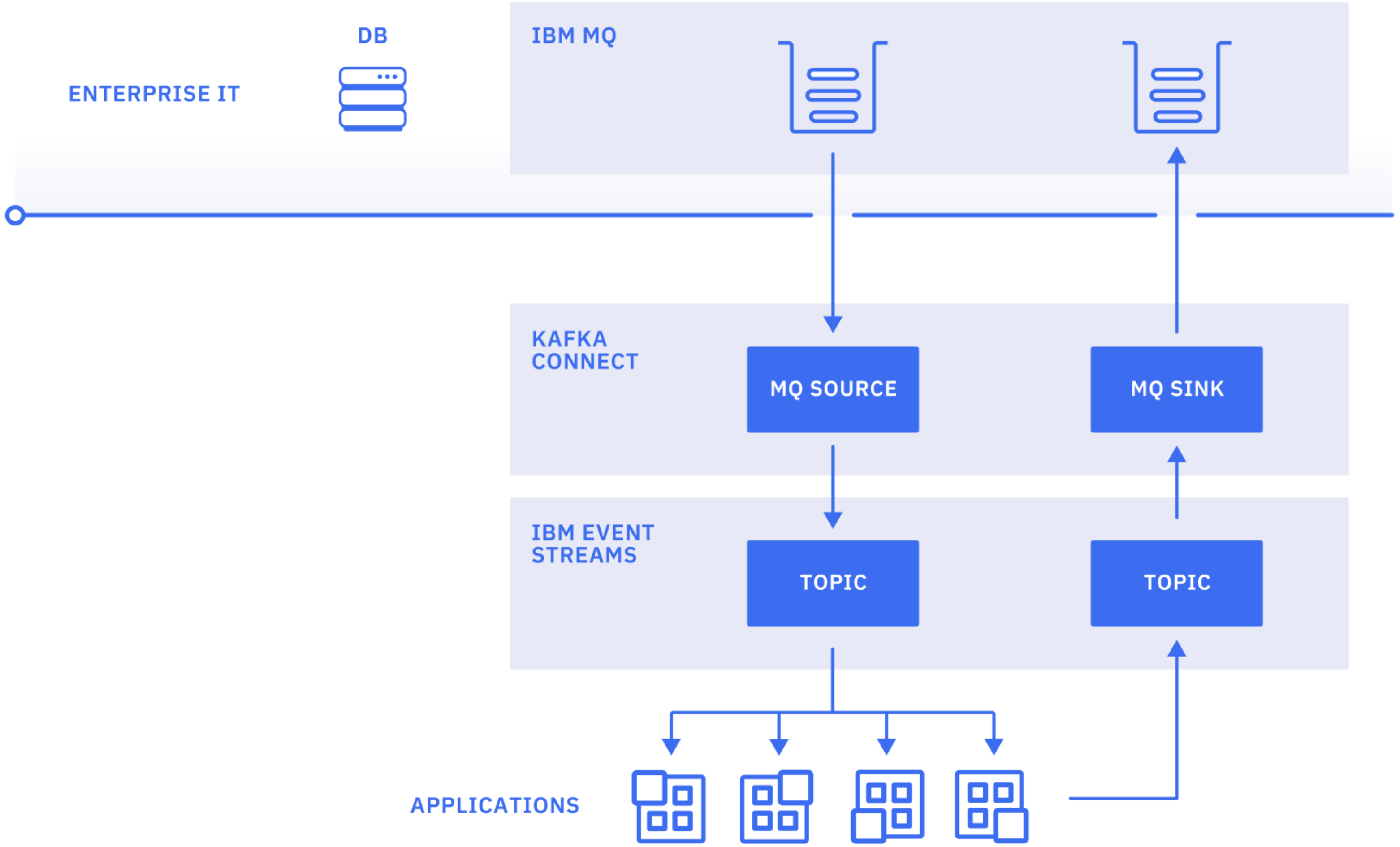
Schema Registry

Associate data schemas with topics to ensure messages are well-formed

Speeds up the rate of new application development as expected data formats are recorded and understood

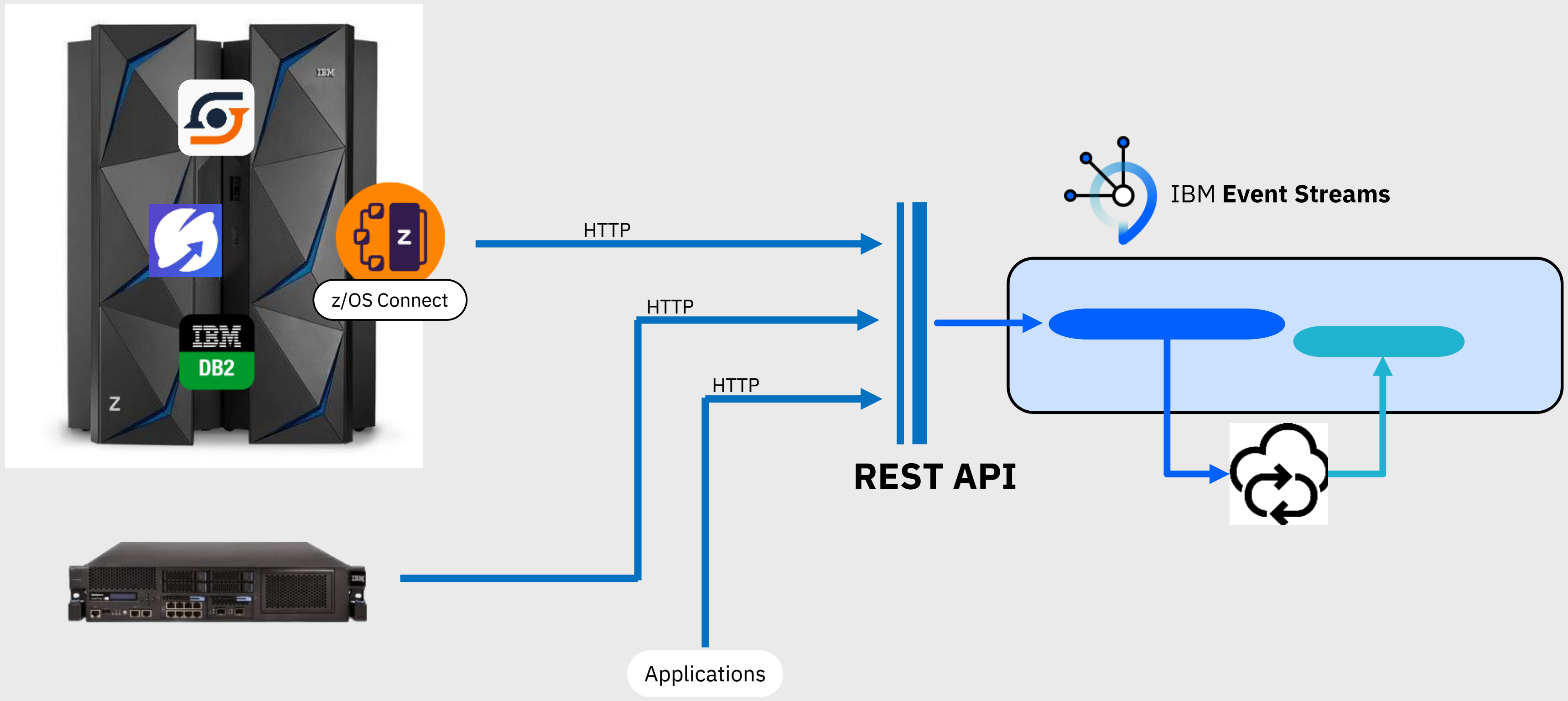
Unlocking events in existing systems


DB
IBM MQ
ENTERPRISE IT
CHANGE DATA CAPTURE
IBM EVENT STREAMS
TOPIC
APPLICATIONS


ENTERPRISE IT
DB
IBM MQ
KAFKA CONNECT
MQ SOURCE
MQ SINK
IBM EVENT STREAMS
TOPIC
TOPIC
APPLICATIONS

Unlock Events from Systems where Kafka clients are not available

REST API for Inbound Data



Publish Events from Anywhere with the REST Producer API

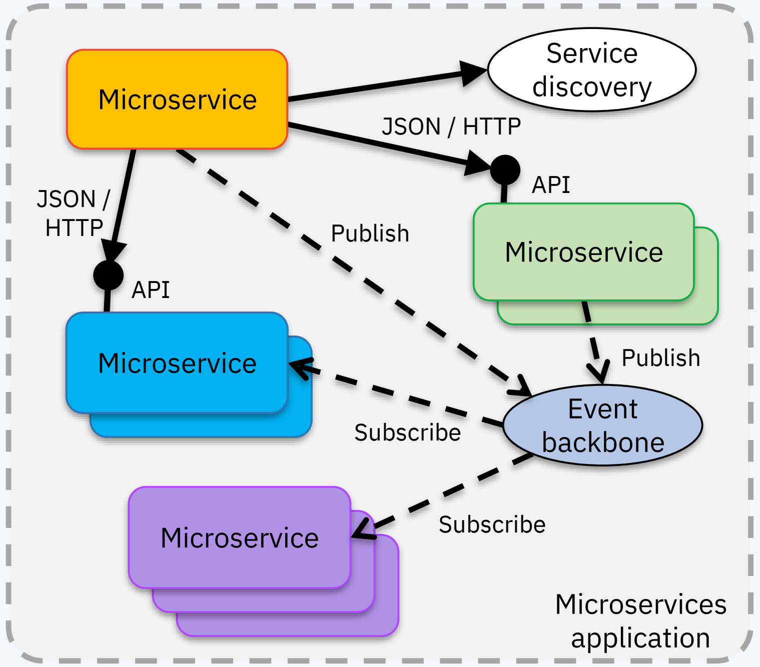
IBM has created a new easy-to-use REST Producer API

```
POST /topics/{topic_name}/records
Content-Type: text/plain
Authorization: Bearer {bearer_token}
Hello Event Streams
```

- Use it wherever it’s difficult to use a real Kafka client e.g. DataPower, z/OS
- Straightforward design makes it easy to use from the command line and developer tools
- Supports partitioning keys and headers
- So easy you can use it from the command line with cURL

Event-driven microservices

Event-driven microservices



Microservices communicate primarily using events, with APIs where required

Microservices can produce and consume events using the publish/subscribe pattern

Events are delivered using an event backbone

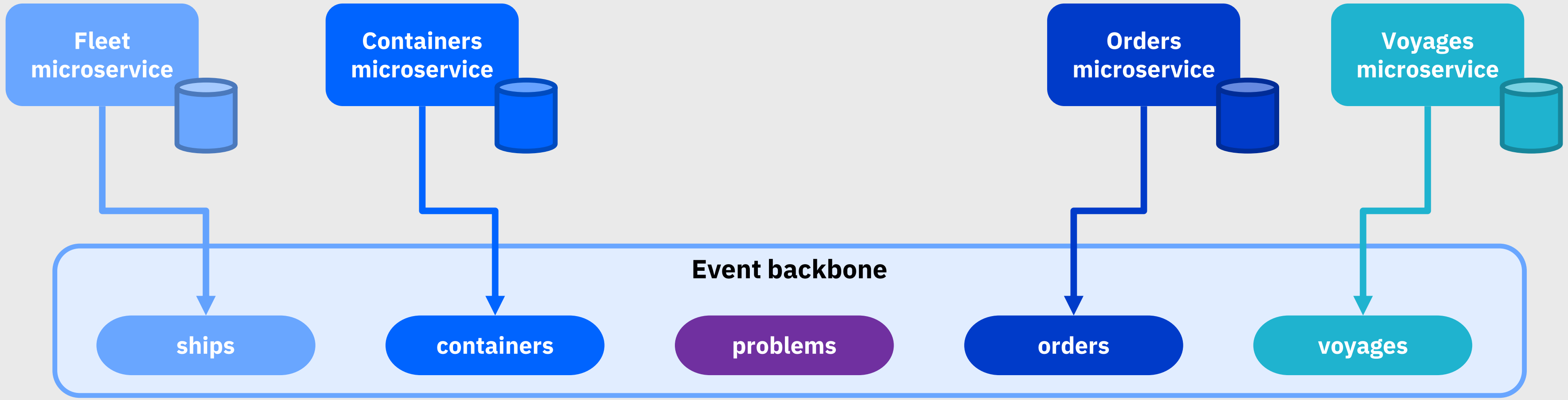
Data is eventually consistent

Pattern – database per microservice

Each microservice persists its own data

Protects independence of the microservice against external change

An event stream (topic) is associated with each microservice for a log of events



Summary

Event-Driven Enterprise using IBM Event Streams

- Event streaming is a powerful new paradigm for messaging

- It’s ideal for building responsive applications, particularly using microservices principles

- IBM Event Streams offers an easy, supported way to use Apache Kafka in your enterprise

Find out more

- Try out Event Streams
  - https://ibm.github.io/event-streams/installing/trying-out/ (On premise)
  - https://console.bluemix.net/catalog/services/event-streams (IBM hosted)

- Get in touch
  - https://ibm.github.io/event-streams/support/

- Apache Kafka
  - https://kafka.apache.org/

- Find out more
  - https://www.ibm.com/cloud/event-streams

Thank You