

Event Streams using Apache Kafka

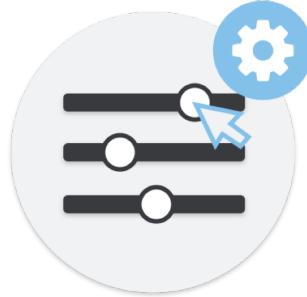
And how it relates to IBM MQ

Andrew Schofield

Chief Architect, Event Streams

STSM, IBM Messaging, Hursley Park

Event-driven systems deliver more engaging customer experiences



Phone company has existing data around customers' usage and buying preferences



Combined with events generated when phones connect to in-store wi-fi



Enables a more engaging and personal customer experience



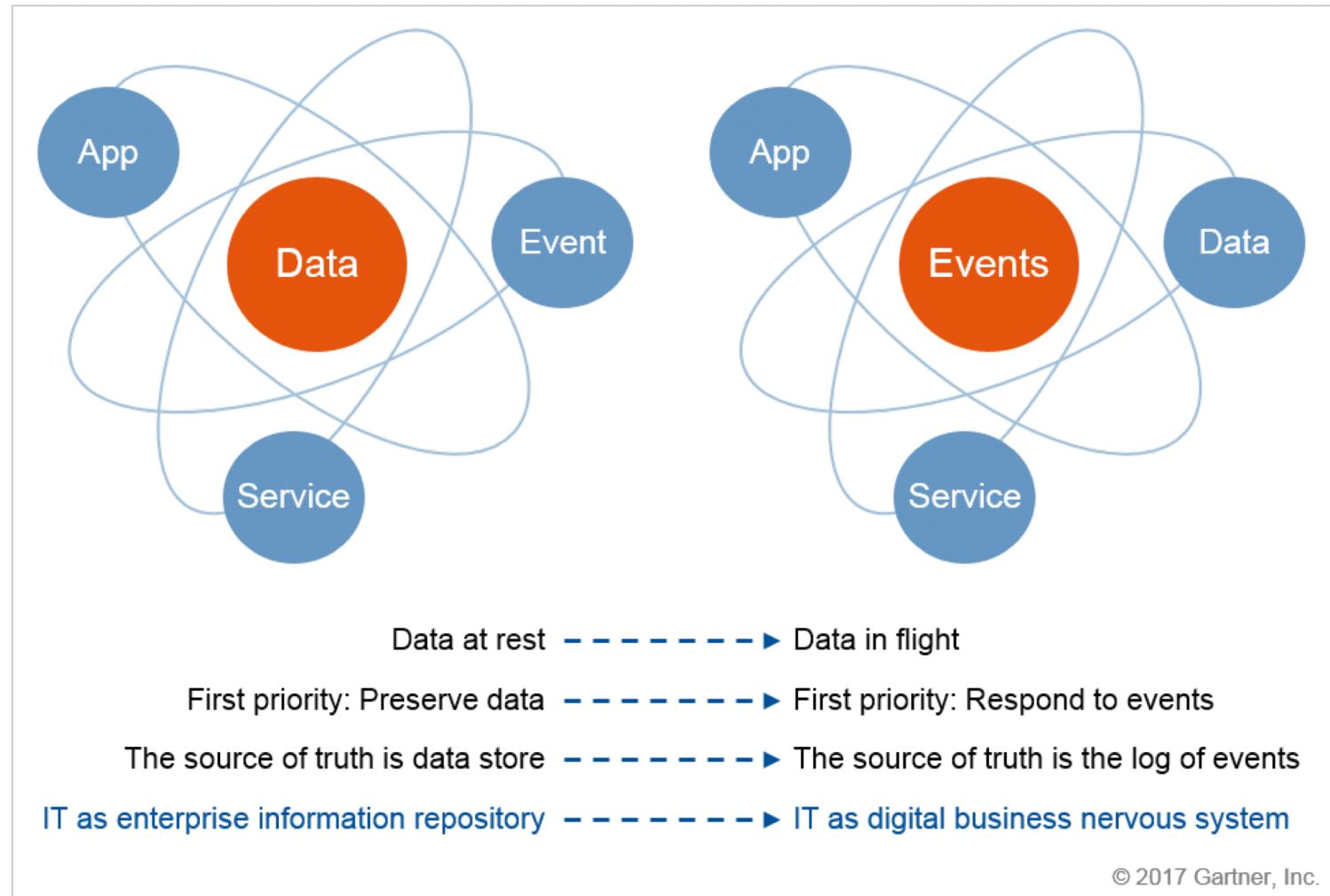
But being event-driven is different

Event driven solutions require
different thinking

Events form the **nervous system** of
the digital business

Application infrastructure needs to
provide event stream processing
capabilities and **support emerging**
event-driven programming models

This is an event-driven journey and
will **underpin the next generation**
of digital customer experiences

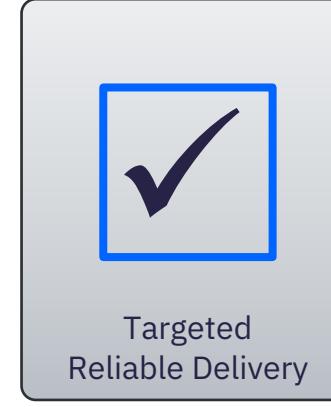
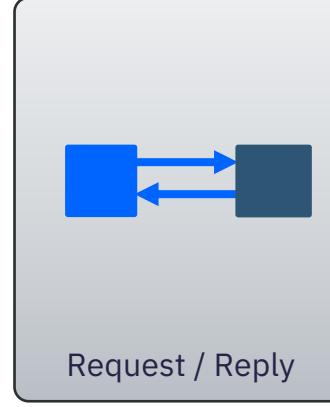
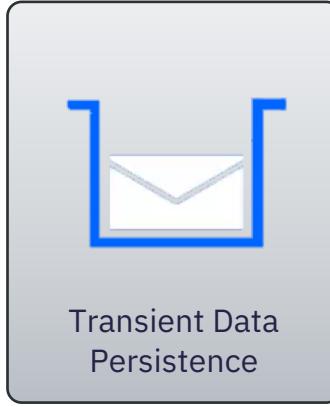


Source: Gartner - From data-centric to event-centric IT priority

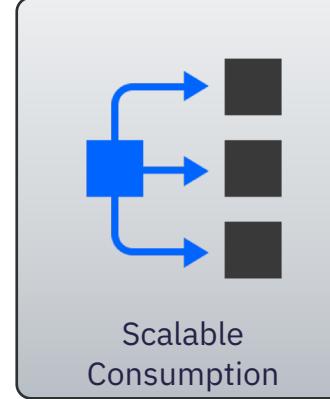
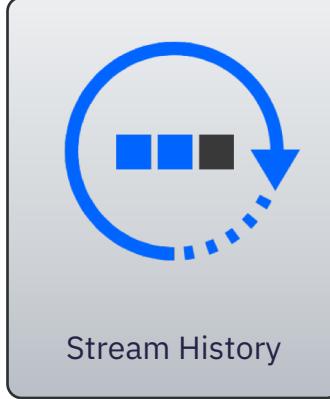
© 2018 IBM Corporation

How does this differ from “messaging”?

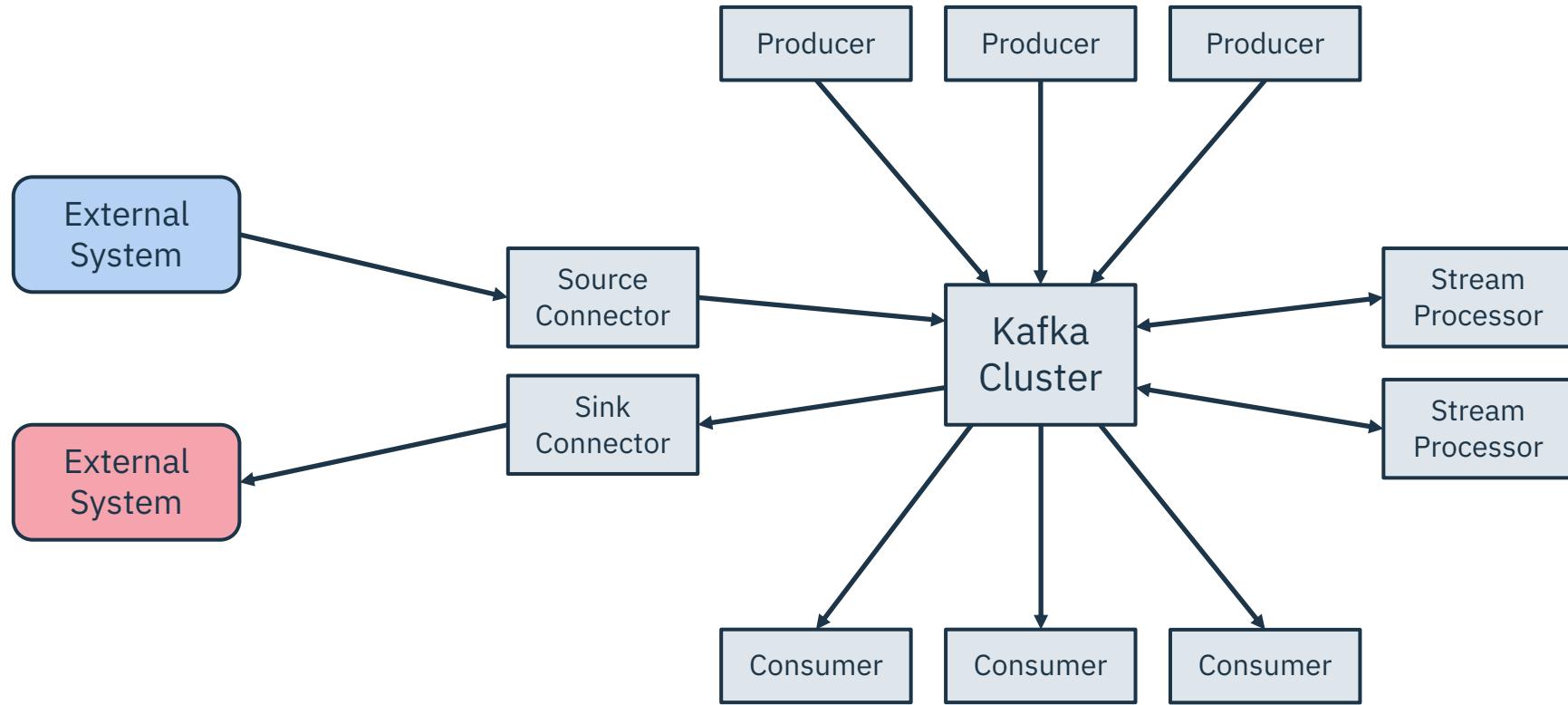
Message Queuing



Event Streaming



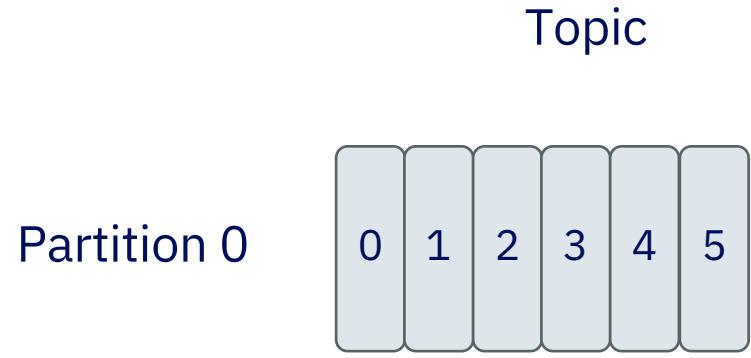
Apache Kafka is an Open-Source Streaming Platform



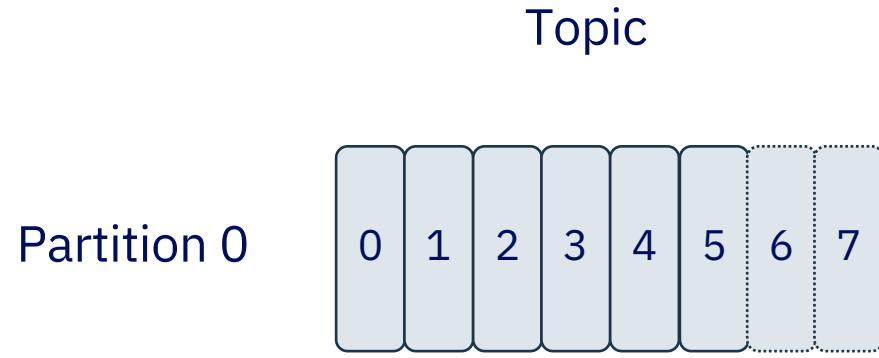
Use cases

- Pub/sub messaging
- Website activity tracking
- Metrics
- Log aggregation
- Stream processing
- Event sourcing
- Commit log

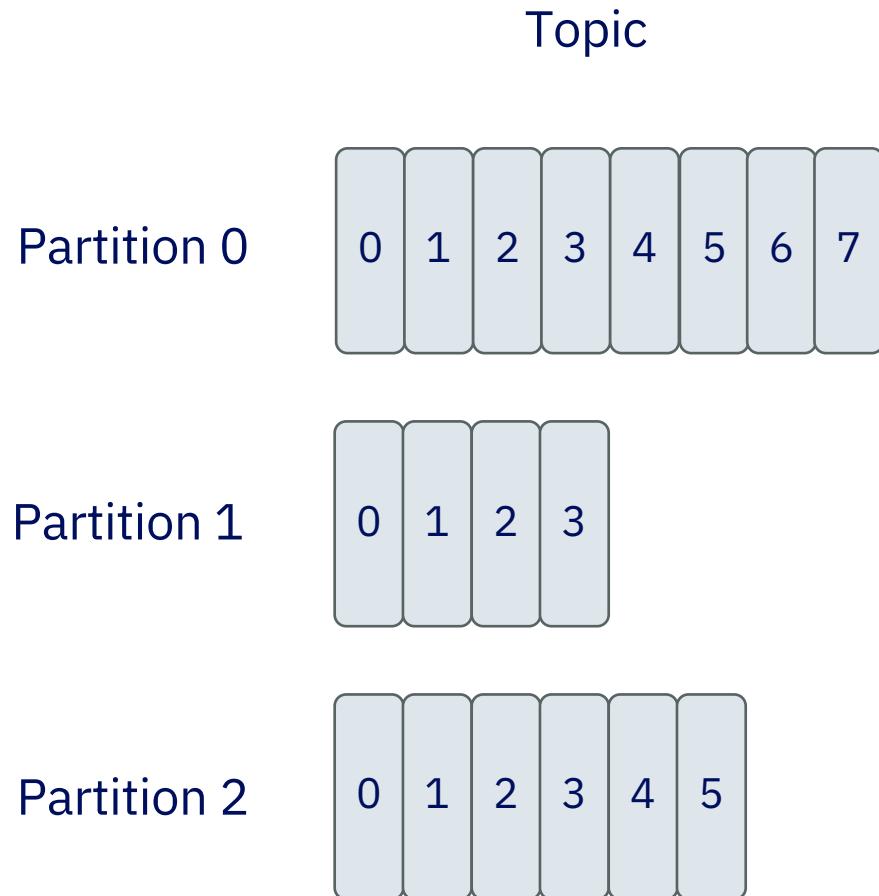
Kafka is built for scale



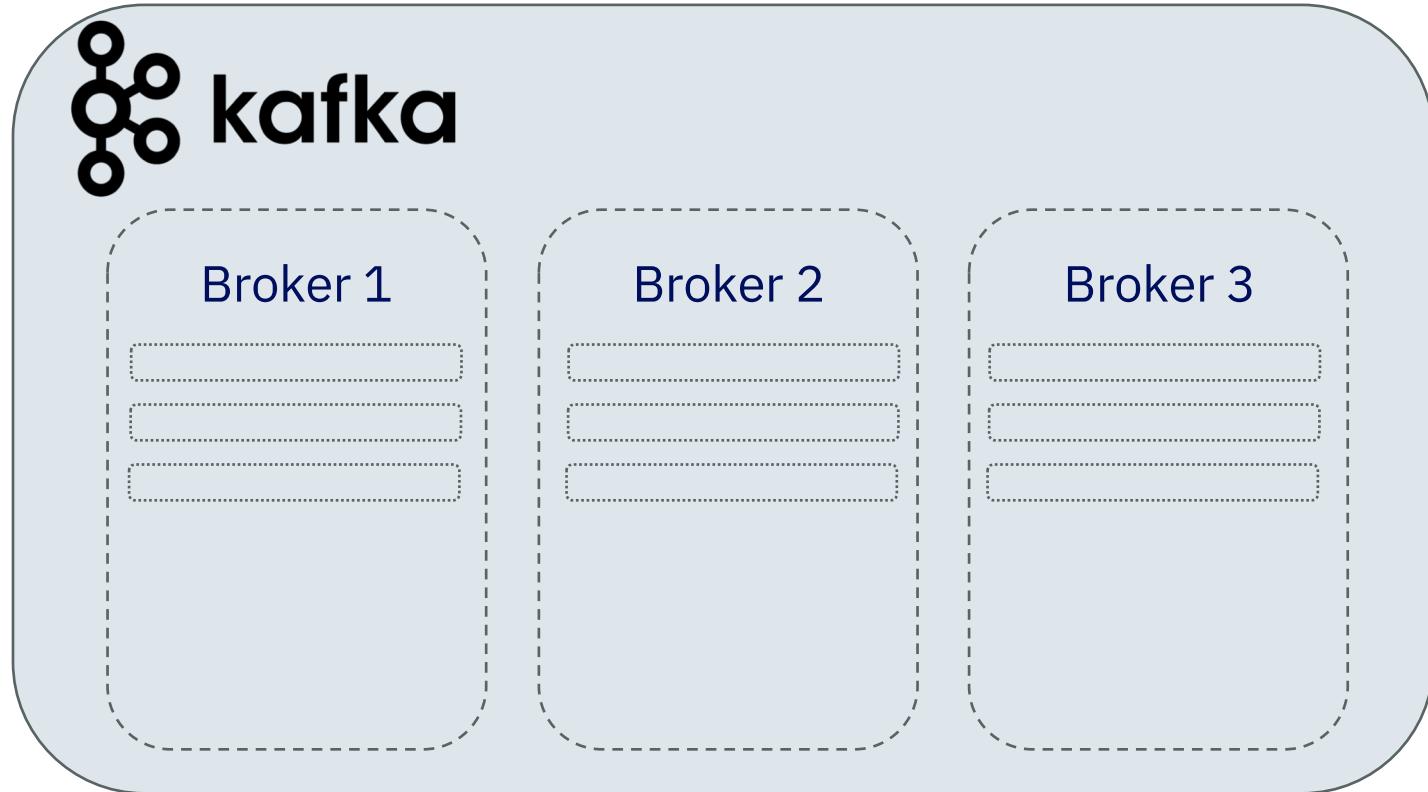
Kafka is built for scale



Kafka is built for scale



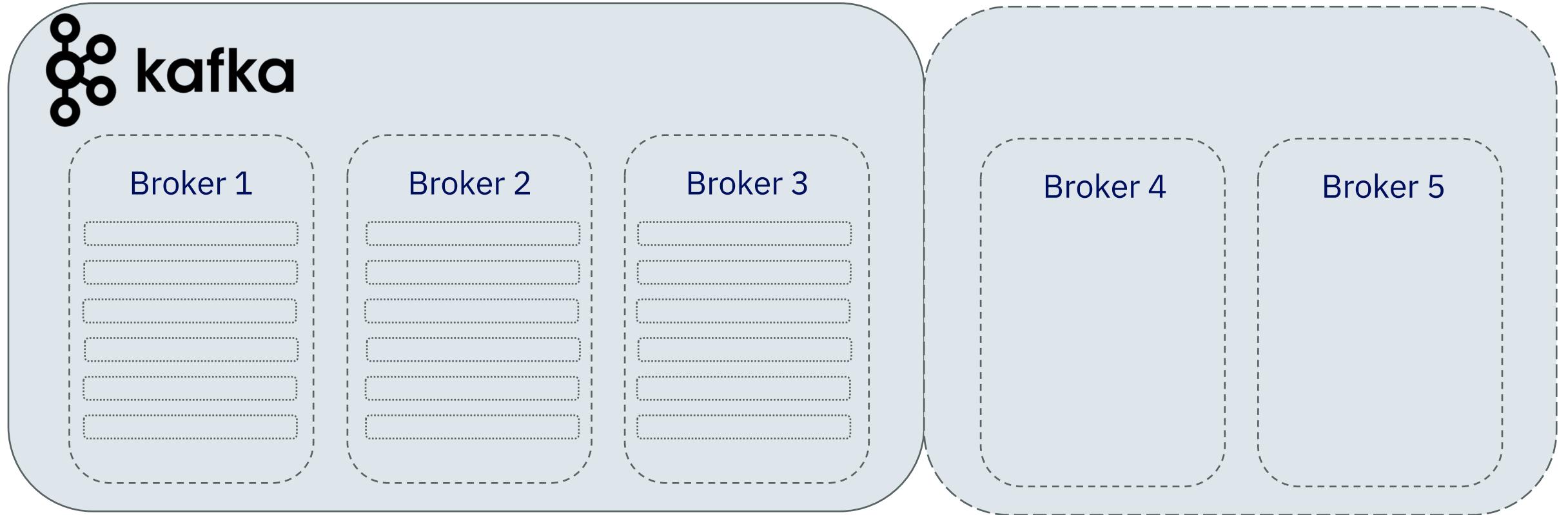
Kafka is built for scale



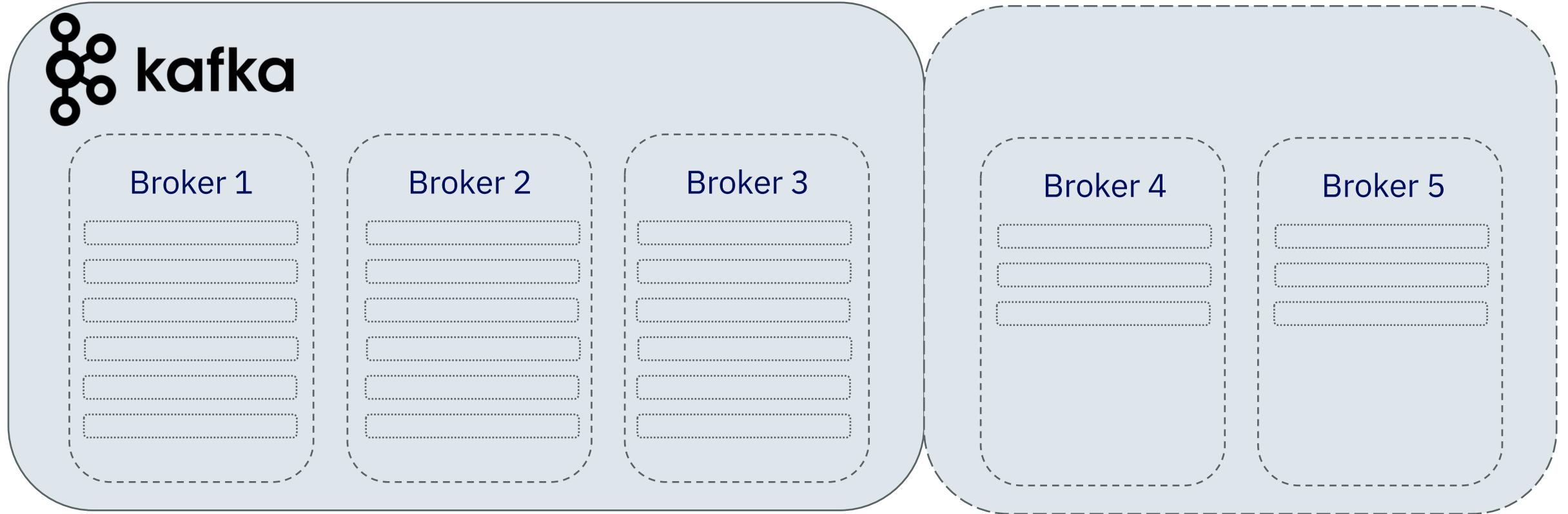
Kafka is built for scale



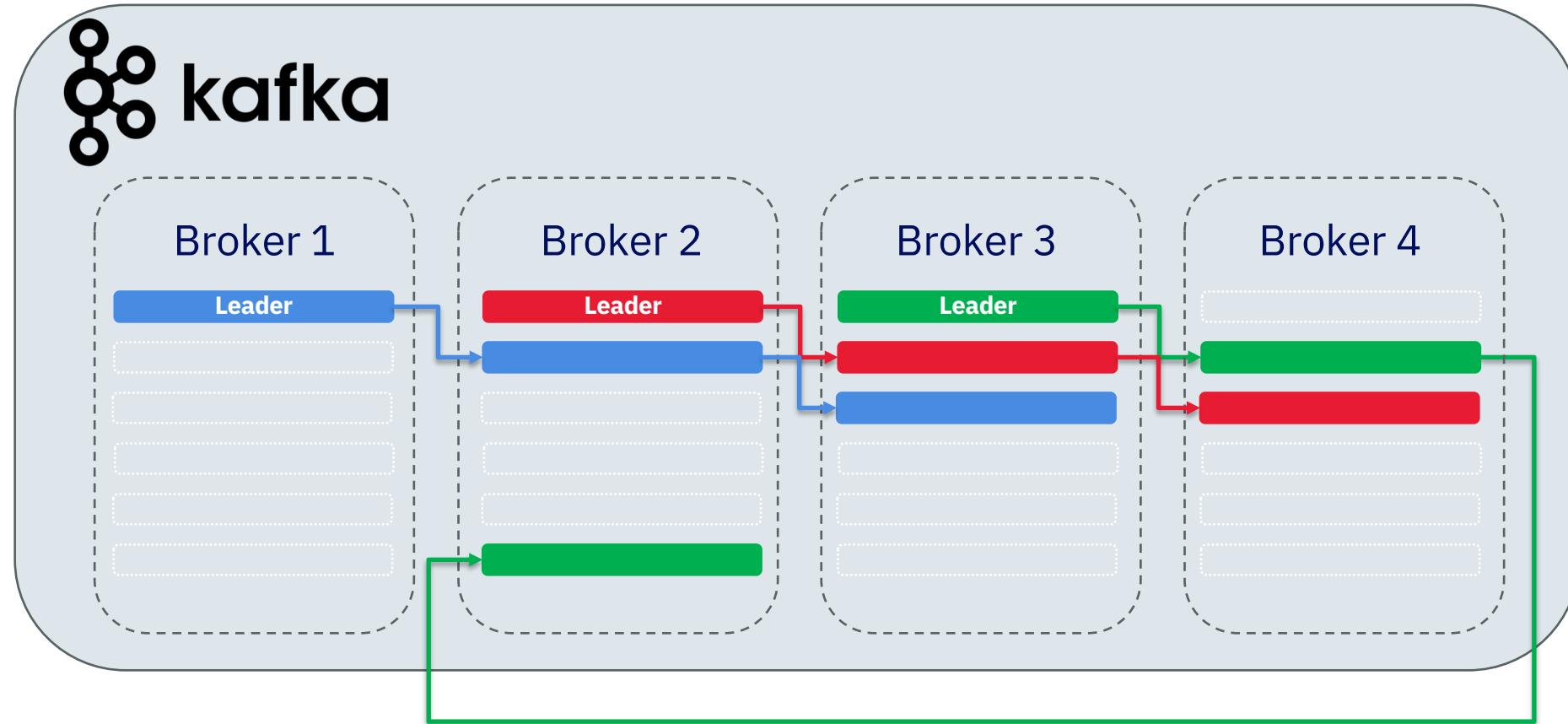
Kafka is built for scale



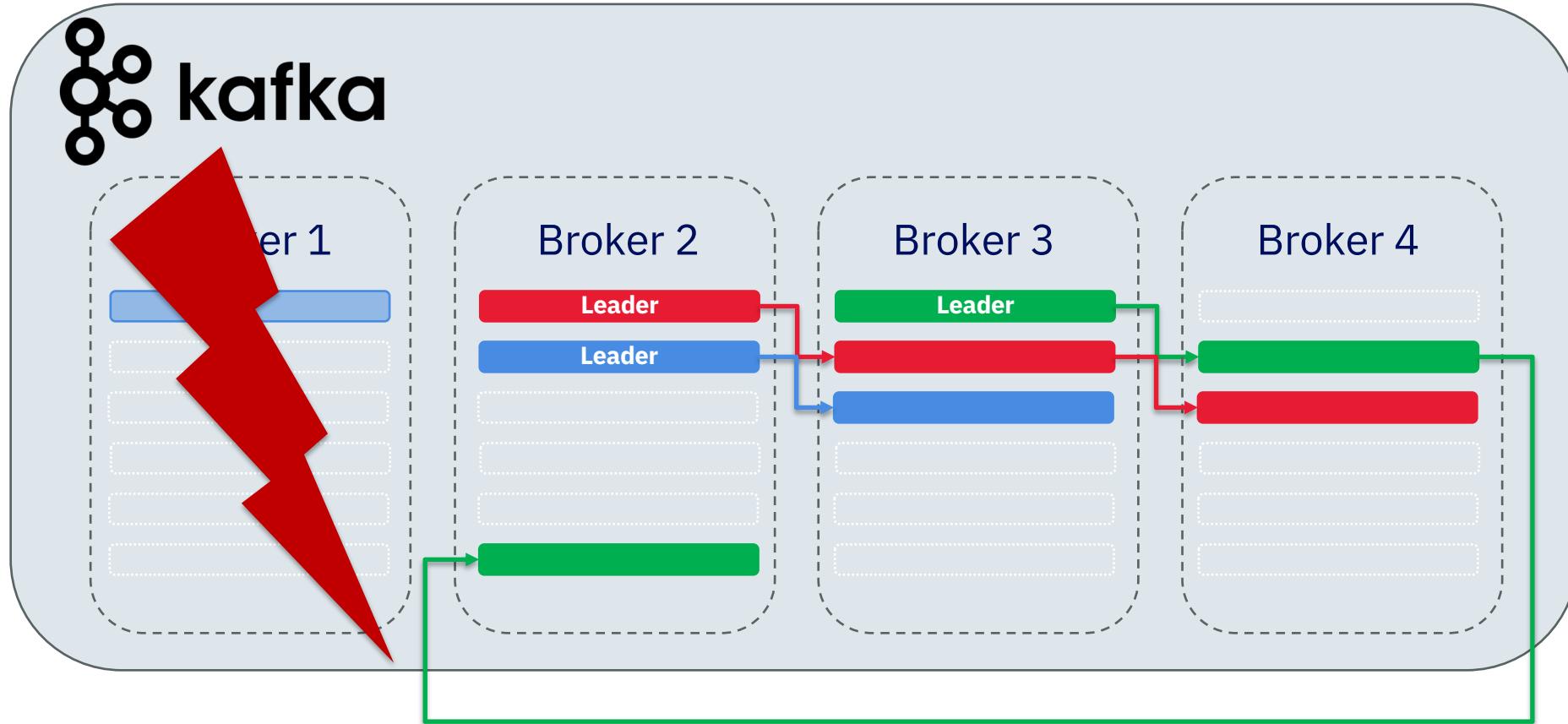
Kafka is built for scale



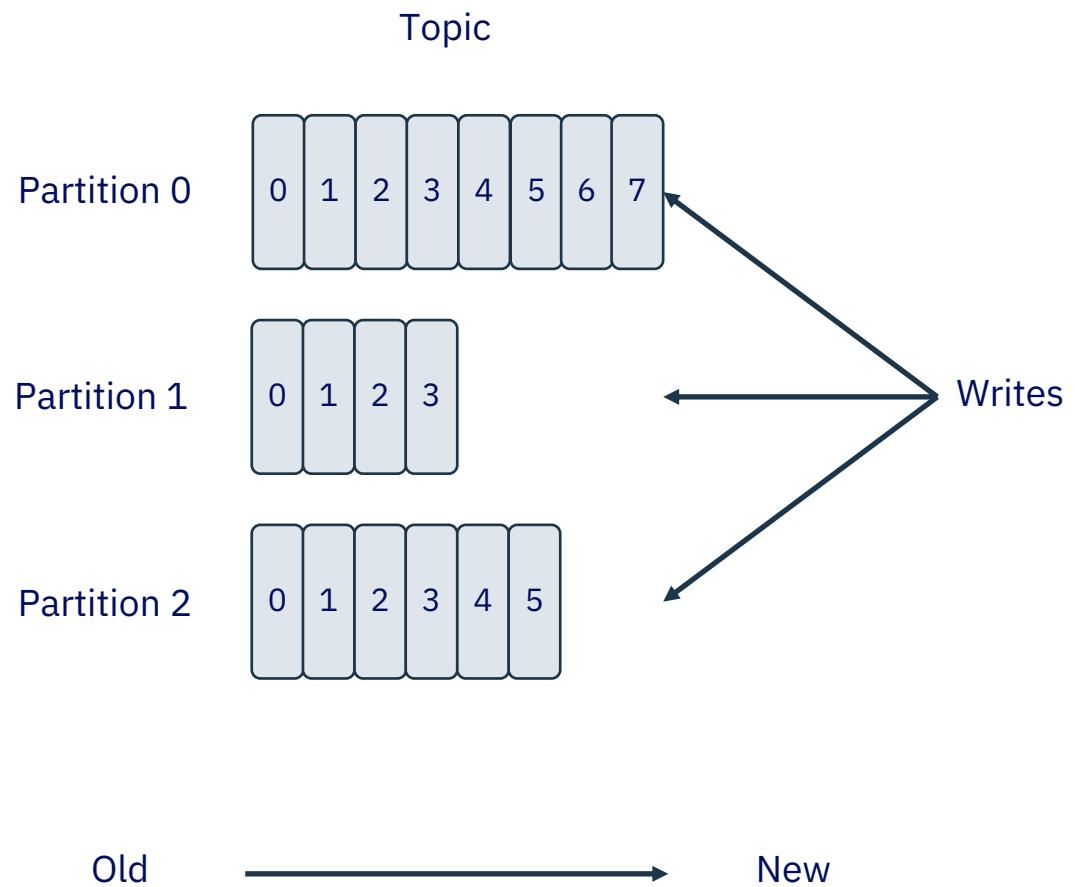
Kafka is built for availability



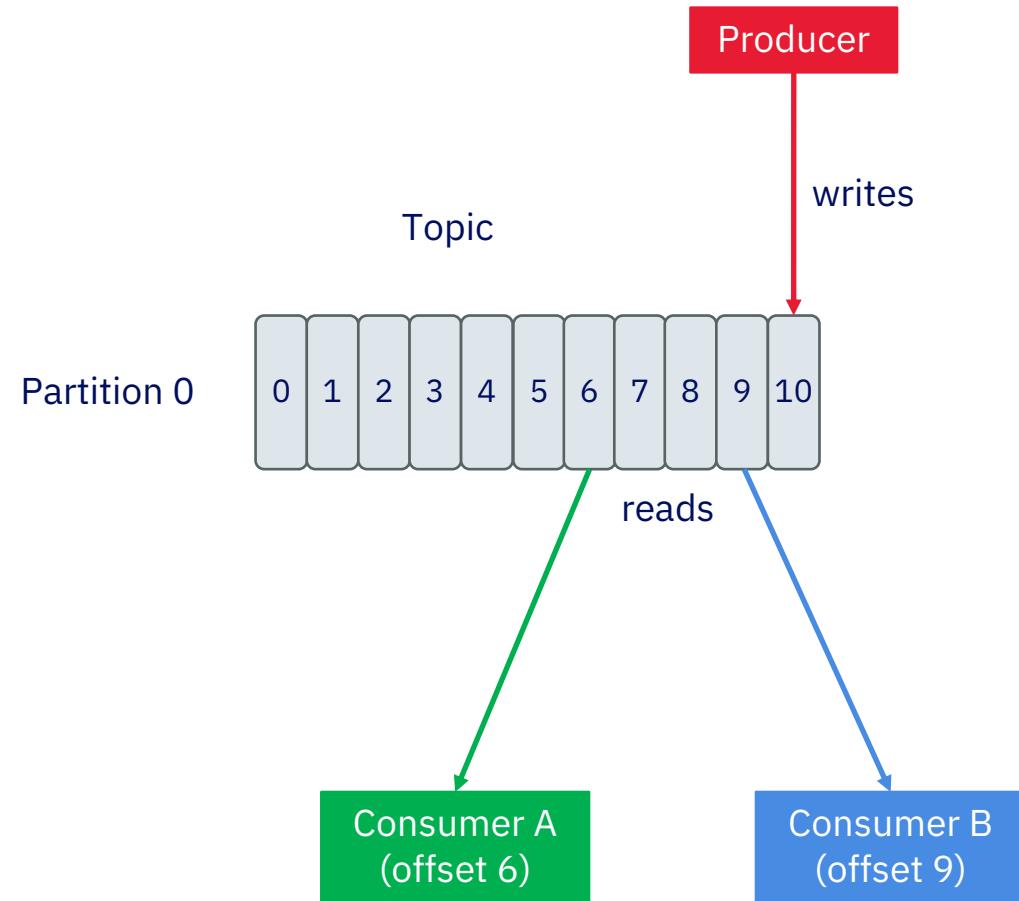
Kafka is built for availability



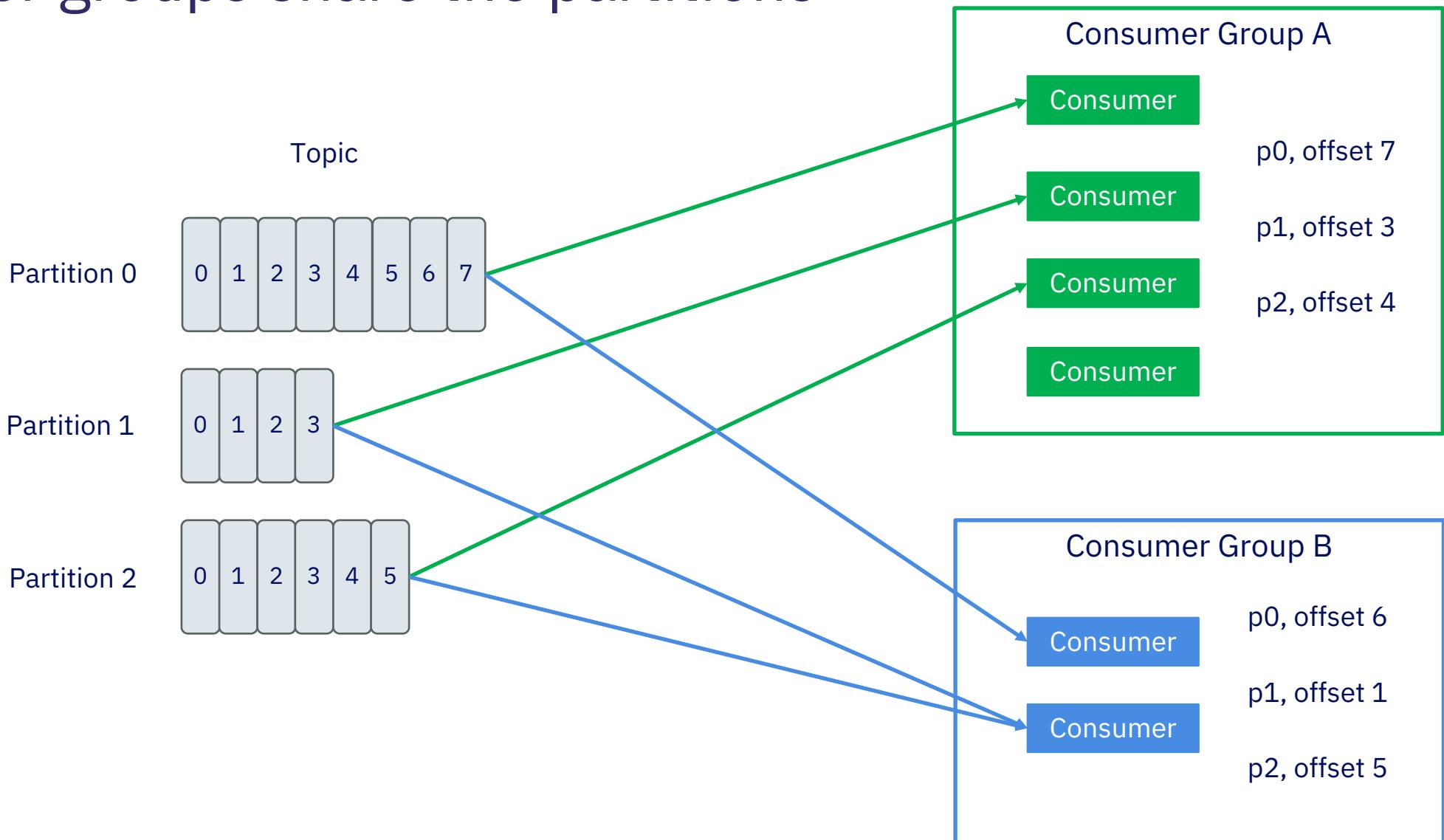
Topics are partitioned for scale



Each partition is an immutable sequence of records



Consumer groups share the partitions



Reliability

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, lose messages on error
- >0 Retry, might result in duplicates on error

Producer can also choose idempotence

Means that retries can be used without risk of duplicates

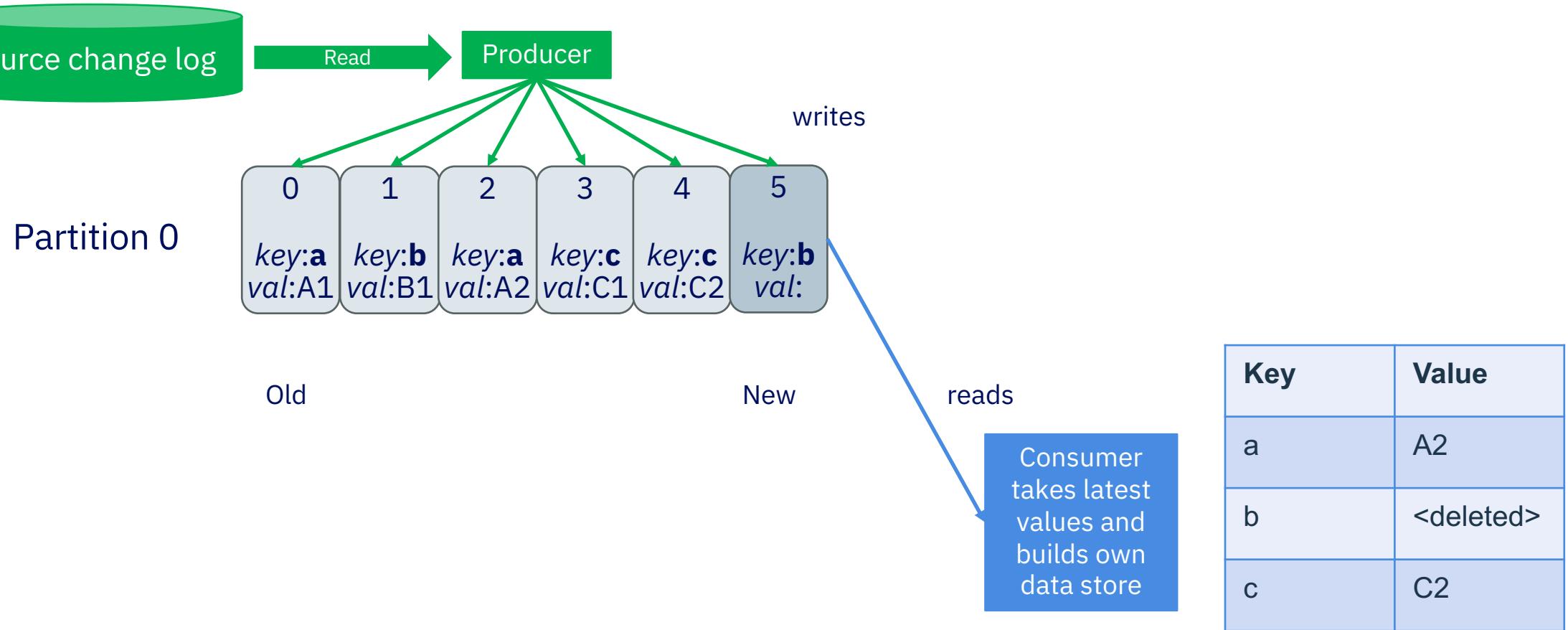
Consumer can choose how to commit offsets

- 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

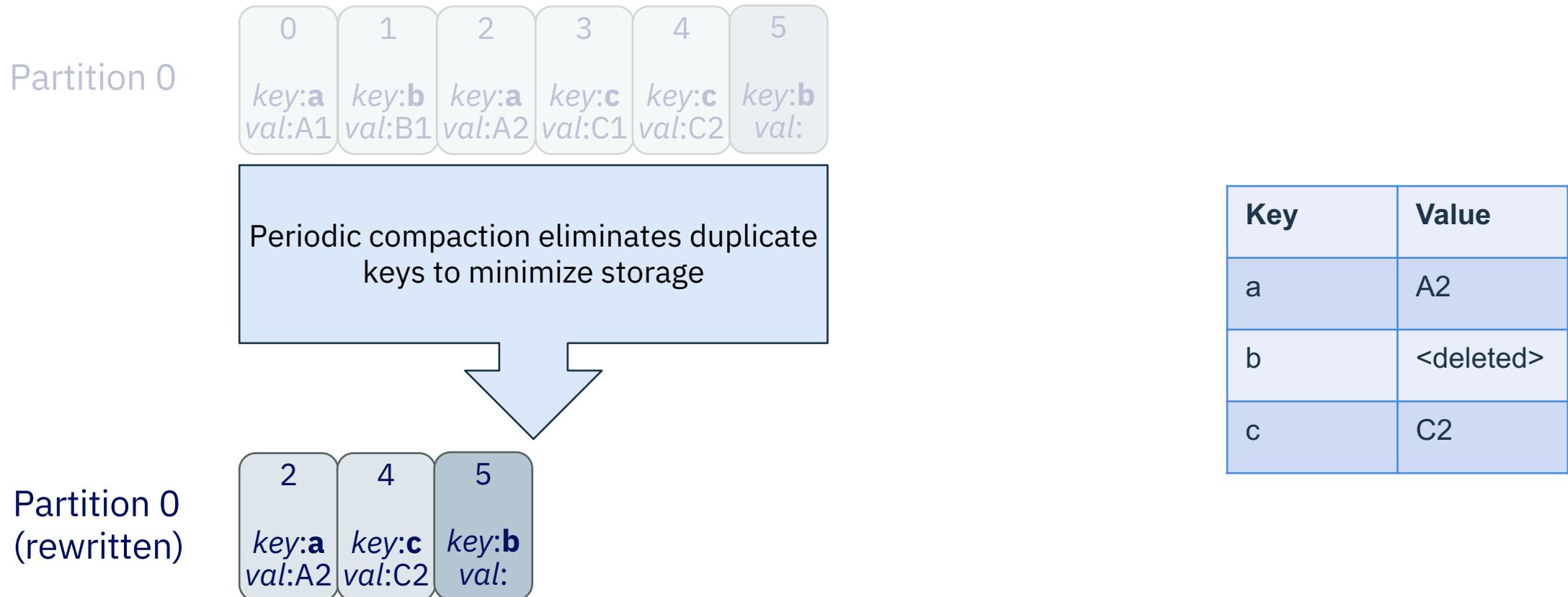
Exactly-once semantics

Can group sending messages and committing offsets into transactions
Primarily aimed at stream-processing applications

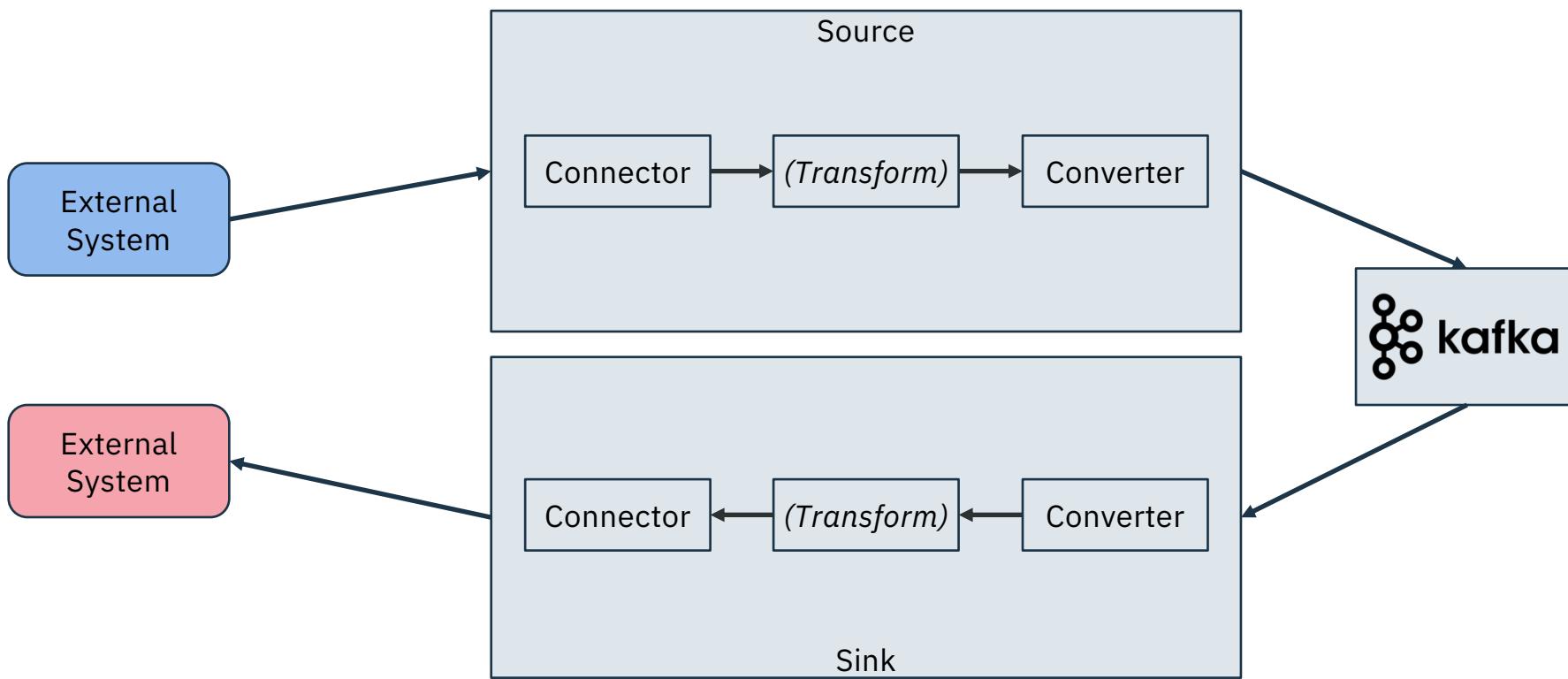
Compacted topics are evolving data stores



Compacted topics are evolving data stores



Kafka Connect



Over 80 connectors

HDFS

Elasticsearch

MySQL

JDBC

IBM MQ

MQTT

CoAP

+ many others

<https://www.confluent.io/product/connectors/>

It's easy to connect IBM MQ to Apache Kafka

IBM has created two open-source connectors available on GitHub

Source Connector

From MQ queue to Kafka topic

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

Sink Connector

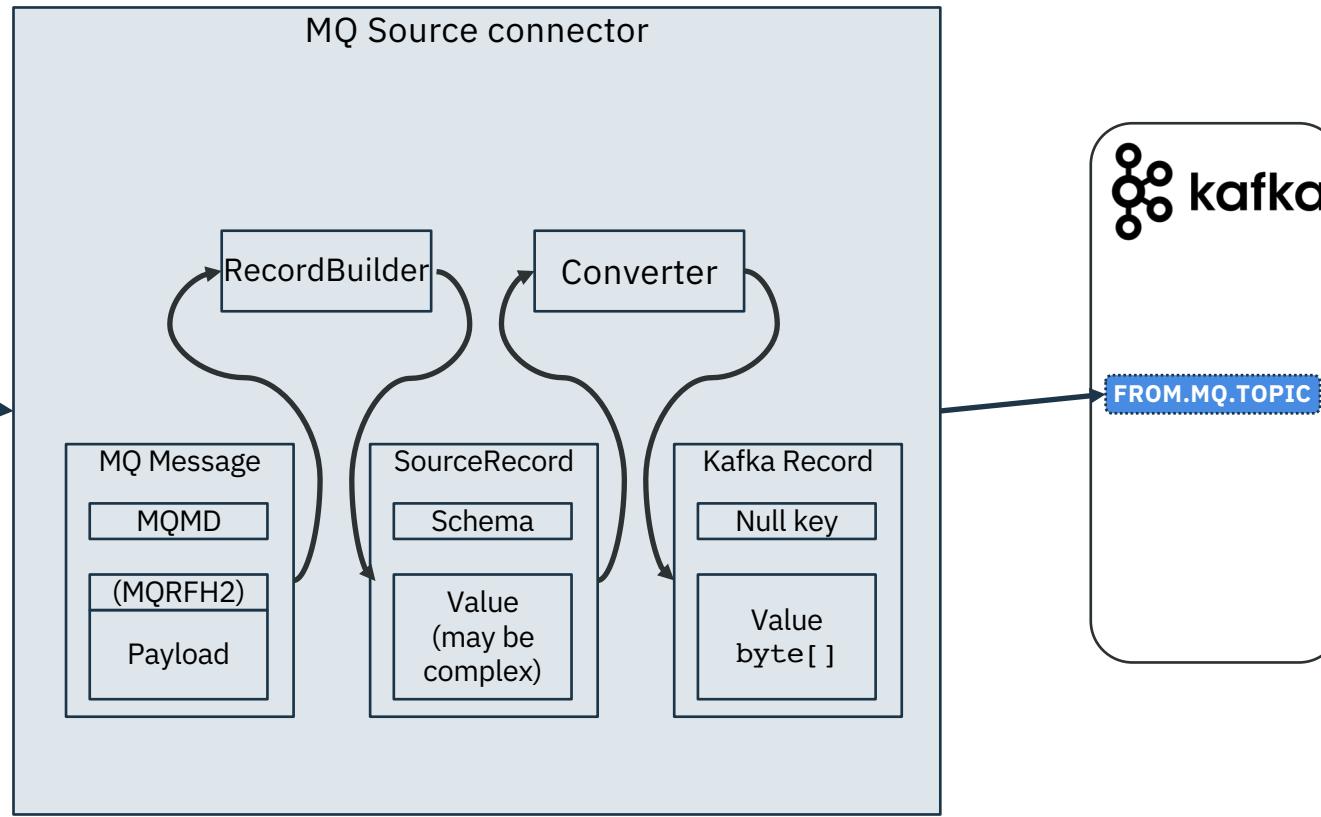
From Kafka topic to MQ queue

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

Detailed instructions for running them:

<https://github.com/ibm-messaging/kafka-connect-mq-source/blob/master/UsingMQwithKafkaConnect.md>

Kafka Connect source connector for IBM MQ



- Open-source – build it yourself
- Use any supported MQ release
- Uses JMS client internally
- Client connections
- Supports TLS, authentication
- MQ queue->Kafka topic
- Support for binary, text, JSON
- Easy to extend

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

© 2018 IBM Corporation

Configuration of MQ Source connector

Configuration is provided in a properties file

Required:

- mq.queue.manager – MQ QMgr name
- mq.connection.name.list – MQ client connname
- mq.channel.name – MQ svrconn channel name
- mq.queue – MQ source queue
- topic – Kafka target topic

Optional:

- mq.user.name – MQ user name for client
- mq.password – MQ password for client
- mq.message.body.jms – native MQ or JMS
- mq.ssl.cipher.suite – MQ SSL cipher suite
- mq.ssl.peer.name – MQ SSL peer name

Sample file provided in GitHub

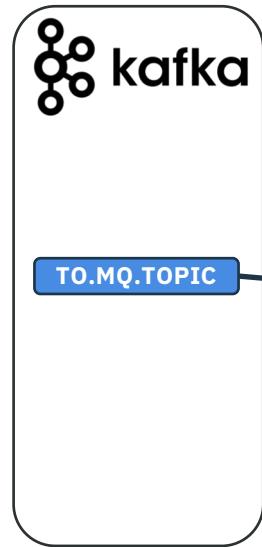
Conversion parameters:

- mq.record.builder
- value.converter

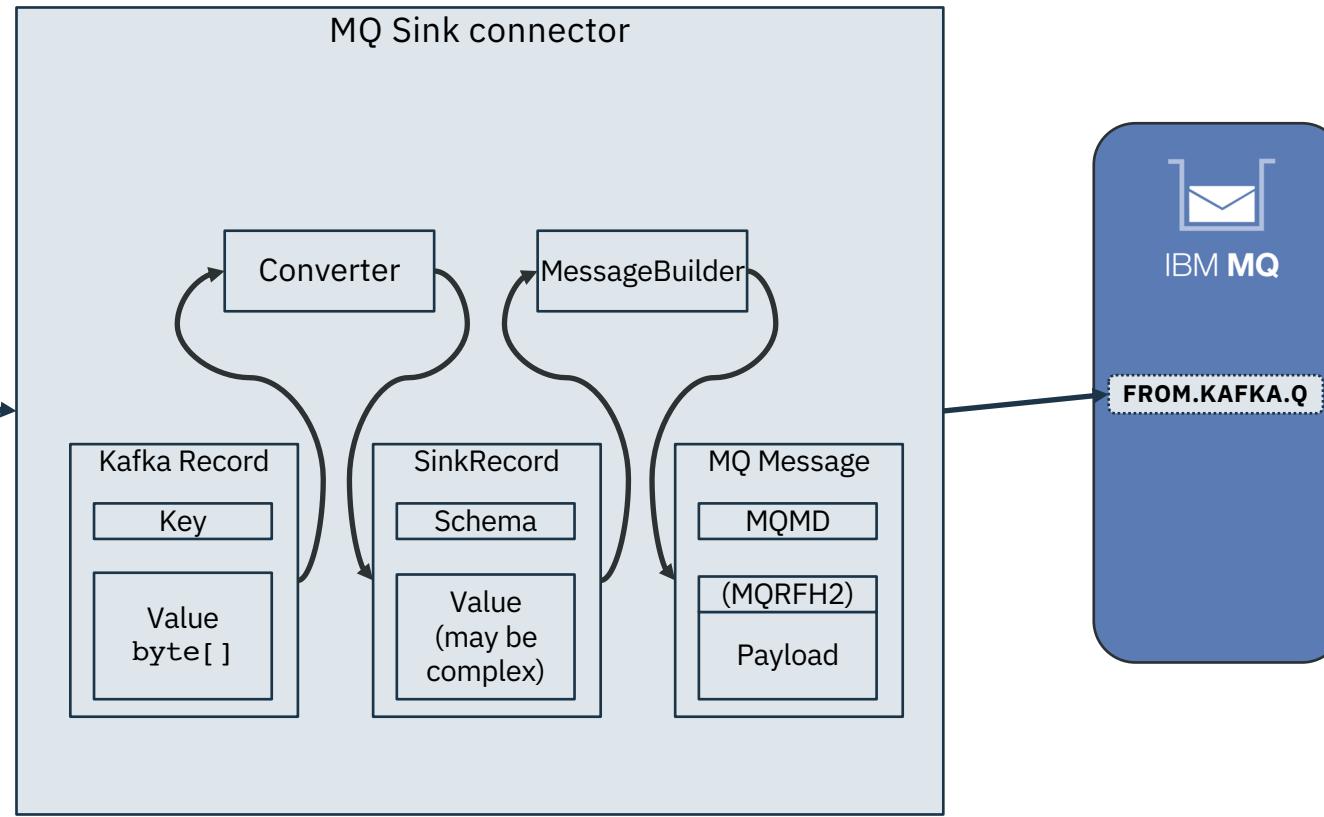
Example:

```
name=mq-source
connector.class=com.ibm.mq.kafkaconnect.MQSourceConnector
tasks.max=1
mq.queue.manager=MYQM
mq.connection.name.list=localhost:1414
mq.channel.name=MYSVRCONN
mq.queue=TO.KAFKA.Q
topic=FROM.MQ.TOPIC
mq.user.name=alice
mq.password=passw0rd
mq.record.builder=com.ibm.mq.kafkaconnect.builders.DefaultRecordBuilder
value.converter=org.apache.kafka.connect.converters.ByteArrayConverter
```

Kafka Connect sink connector for IBM MQ



TO.MQ.TOPIC



- Open-source – build it yourself
- Use any supported MQ release
- Uses JMS client internally
- Client connections
- Supports TLS, authentication
- Kafka topic -> MQ queue
- Support for binary, text, JSON
- Easy to extend

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

© 2018 IBM Corporation

Configuration of MQ Sink connector

Configuration is provided in a properties file

Sample file provided in GitHub

Required:

- topics – Kafka source topic list
- mq.queue.manager – MQ QMgr name
- mq.connection.name.list – MQ client connname
- mq.channel.name – MQ svrconn channel name
- mq.queue – MQ target queue

Optional:

- mq.user.name – MQ user name for client
- mq.password – MQ password for client
- mq.message.body.jms – native MQ or JMS
- mq.ssl.cipher.suite – MQ SSL cipher suite
- mq.ssl.peer.name – MQ SSL peer name
- mq.time.to.live – MQ message expiration
- mq.persistent – MQ message persistence

Conversion parameters:

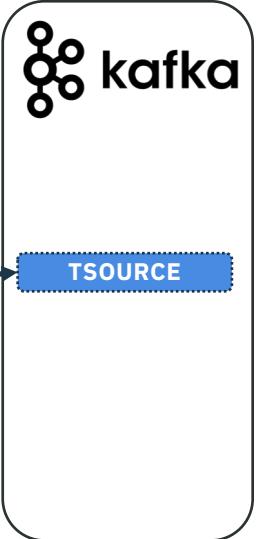
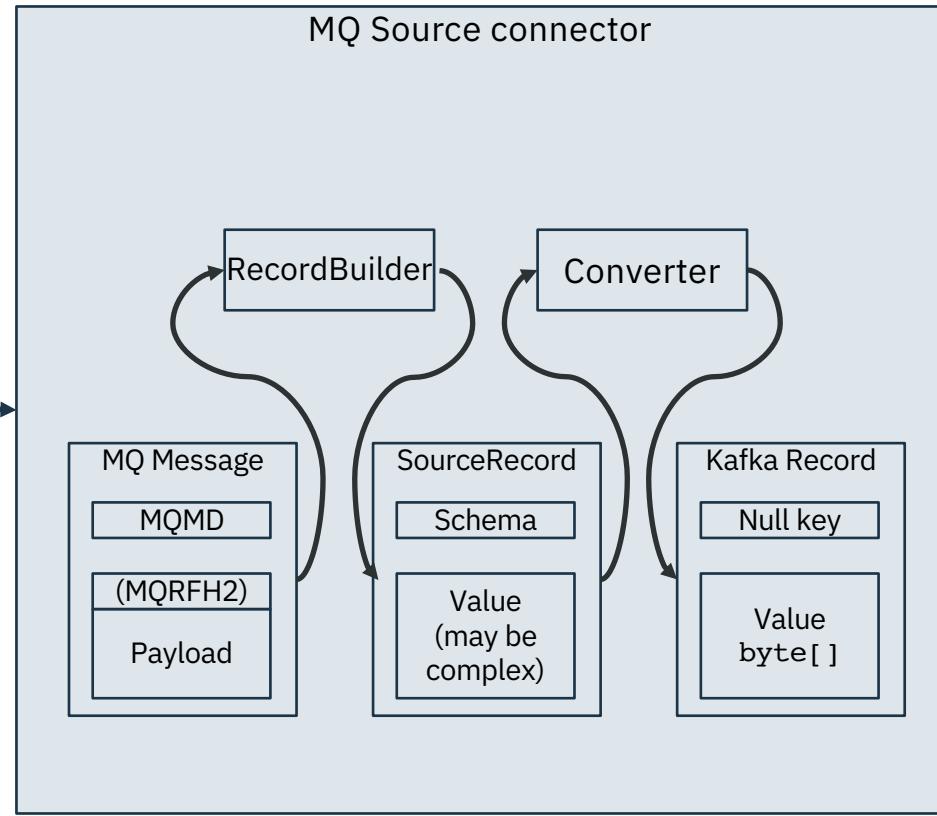
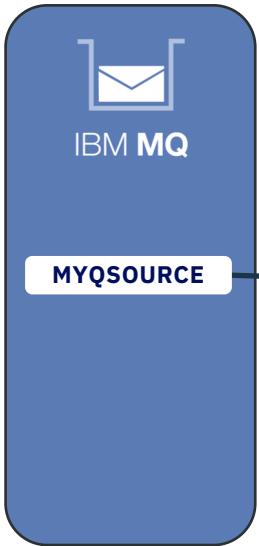
- mq.message.builder
- value.converter

Example:

```
name=mq-sink
connector.class=com.ibm.mq.kafkaconnect.MQSinkConnector
tasks.max=1
topics=TO.MQ.TOPIC
mq.queue.manager=MYQM
mq.connection.name.list=localhost:1414
mq.channel.name=MYSVRCONN
mq.queue=FROM.KAFKA.Q
mq.user.name=alice
mq.password=password
mq.message.builder=com.ibm.mq.kafkaconnect.builders.DefaultMessageBuilder
value.converter=org.apache.kafka.connect.converters.ByteArrayConverter
```

DEMO

Ubuntu 14.04 virtual machine

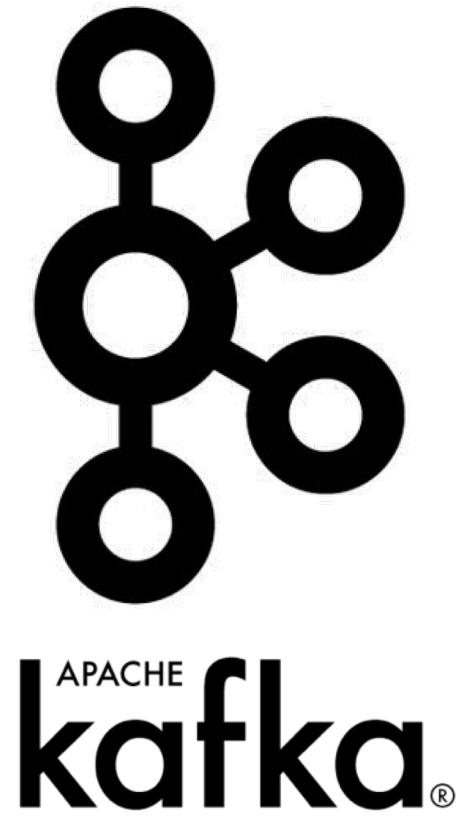


Introducing IBM Event Streams

React to events in real-time to deliver more engaging experiences for your customers

Deploy **production-ready**
Apache Kafka onto
IBM Cloud Private
in minutes

Build intelligent apps on
Kafka with the **confidence**
IBM is supporting you



Rely on disaster recovery &
security designed for
mission-critical use

Exploit existing data to
become a real Event-Driven
Enterprise

IBM Event Streams builds on the open standards of IBM Cloud Private



Executable package
of software that
includes everything
needed to run it

Containers



kubernetes

Automate
deployment,
scaling, and
management of
containerized
applications

Orchestration



Define, install, and
upgrade Kubernetes
applications

Management



HashiCorp
Terraform

Infrastructure as
code to provision
public cloud and on-
premises
environments

Provisioning

Benefit from the Core Services of IBM Cloud Private

Enterprise Content Catalog
Open Source and IBM Middleware, Data, Analytics, and AI Software

Core Operational Services
Log Management, Monitoring, Metering, Security, Alerting

Kubernetes Container Orchestration Platform

Choose your infrastructure:



vmware®

IBM Z



Strategic Value:

Self-service catalog

Agility, scalability, and elasticity

Self-healing

Enterprise security

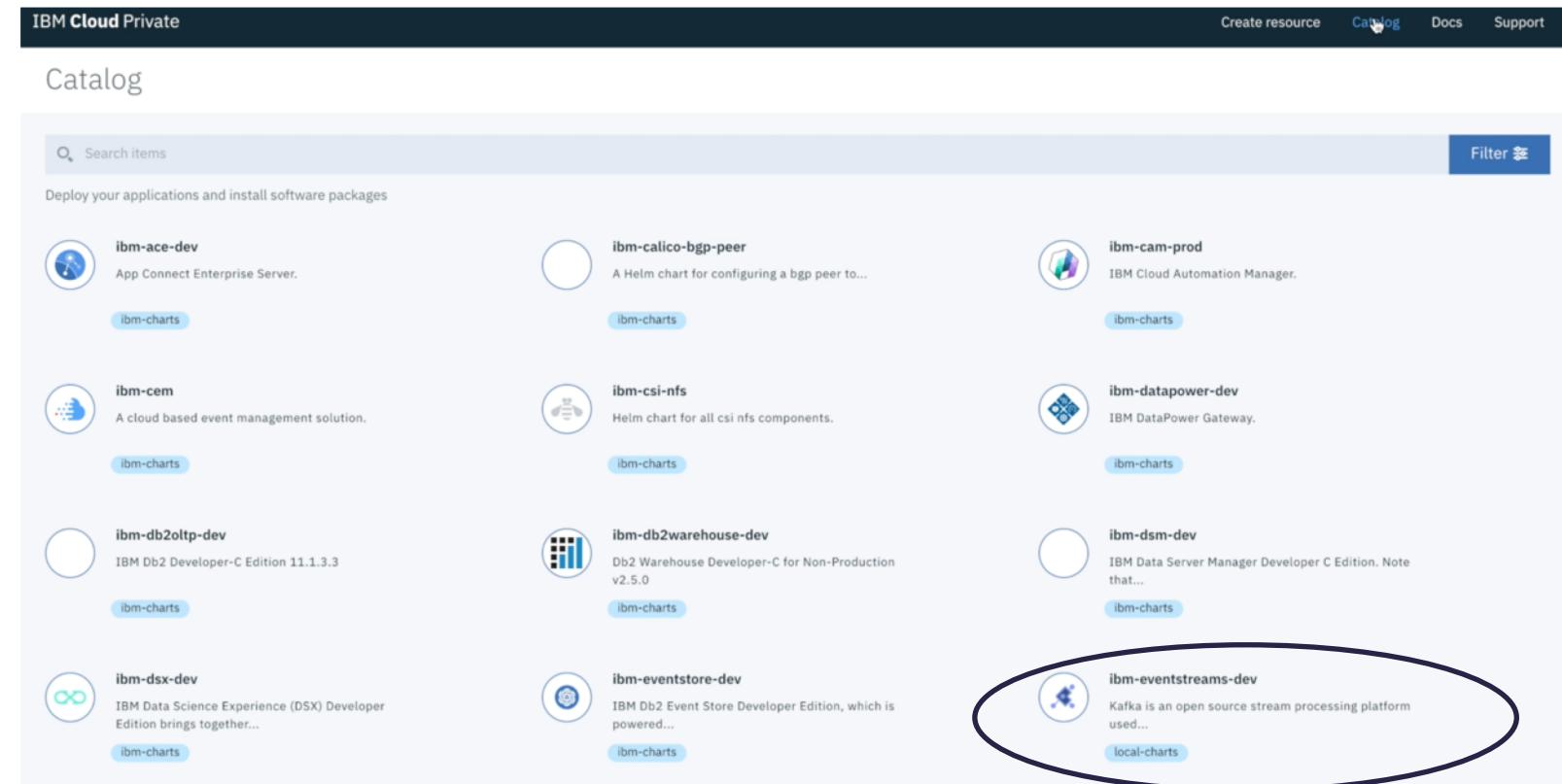
No vendor lock-in



Apache Kafka Orchestrated with Kubernetes and Helm

- IBM Event Streams is packaged as a Helm chart
- A 3-node Kafka cluster, plus ZooKeeper, UI, network proxies and so on is over 20 containers
- Kubernetes and Helm brings this all under control

- Install a Kafka cluster with a few clicks from the IBM Cloud Private catalog
- It comes
 - Highly available
 - Secure
 - Ready for production



High Availability, Scaling and Configuration with Ease

- **Highly available by design**

- Brokers are spread across ICP worker nodes using anti-affinity policies
- Minimizes the risk of down-time in the event of a node outage

- **Scale the Kafka cluster up with one command**

- Safely grows the stateful set, reconfigures the network interfaces and gives you more capacity

- **Roll out Kafka cluster configuration changes easily**

- Make a single configuration change and Event Streams rolls it out across the brokers in the cluster
- Broker availability is managed using health checks to ensure that availability is maintained

Safe, Planned Upgrade of Apache Kafka

Upgrade Kafka versions safely and without hassle

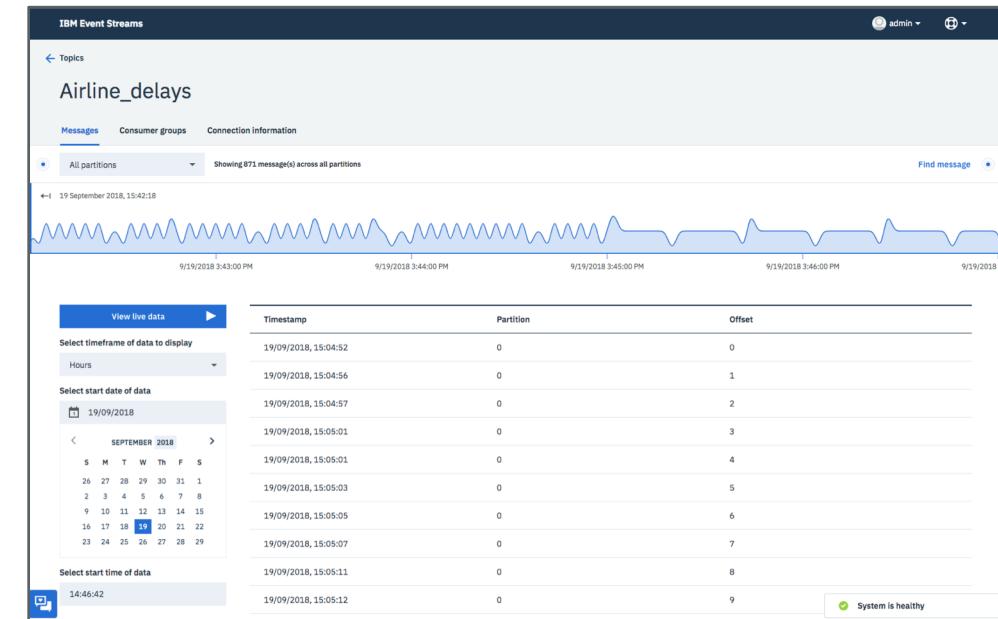
- First, upgrade the Helm chart to a newer version of IBM Event Streams
 - Rolling update of the Kafka brokers minimizes disruption
- As a separate step, upgrade the broker data and protocol version to complete the upgrade
 - Until this point, you can roll back

IBM Event Streams | Making Apache Kafka Intuitive and Easy

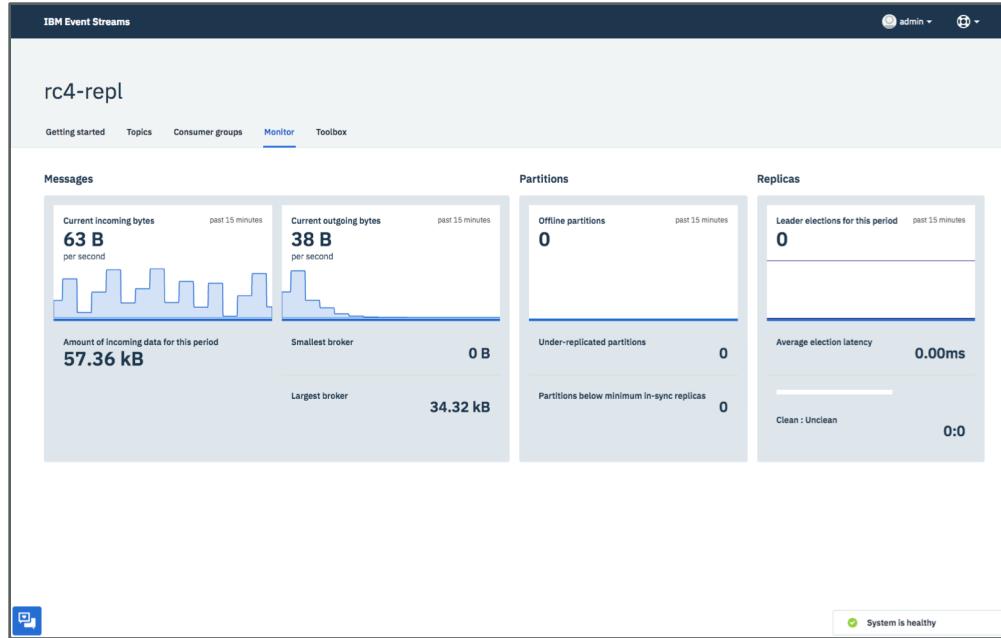
The screenshot shows a browser window titled "IBM Cloud Private" with the URL "Not Secure | https://9.30.247.81:8443/catalog/catalogdetails/ibm-charts/ibm-events...". The page displays the configuration for "Configure ibm-eventstreams-dev V 0.1.1". It includes sections for "Kafka broker configuration" with fields for "CPU limit for Kafka brokers" (1000m) and "Memory limit for Kafka brokers" (1Gi), and "CPU request for Kafka brokers" (1000m) and "Memory request for Kafka brokers" (1Gi). At the bottom are "Cancel" and "Install" buttons.

Simple deploy with just 3 clicks

Visualisation of your topic data



IBM Event Streams | Making Apache Kafka Intuitive and Easy



Monitor status at a glance

Integrated feedback and support

IBM Event Streams support
This is your opportunity to let us know what you think about IBM Event Streams.

Open an issue
Did something go wrong? We are here to help.

Suggest a feature
What would you love to see us add next?

Give us feedback
Let us know how we can make IBM Event Streams even better.

[Existing issues](#) [Existing feature requests](#) [Chat with us on Slack](#)

[View Frequently Asked Questions \(FAQs\)](#)



Security – Authentication and Access Control

- User and group information controlled centrally
 - Integrate with your corporate LDAP through IBM Cloud Private
- Control access to Event Streams resources using role-based access control policies
 - Assign roles to users: Viewer, Editor, Operator, Administrator
 - Optionally, specify access for individual resources, such as topic T
- Application access control using service IDs
 - Same concepts as controlling user access
 - Can restrict application access to exactly the resources they need
 - Prevent accidental or intentional interference between applications

Example policy

Permit Bob to write to topic T

User	bob
Role	Editor
Service	Event Streams instance R
Type	topic
Resource	T

Service action	Roles	Permissions
topic.read	Viewer and above	Read messages or config
topic.write	Editor and above	Write messages
topic.manage	Operator and Administrator	Delete or change config

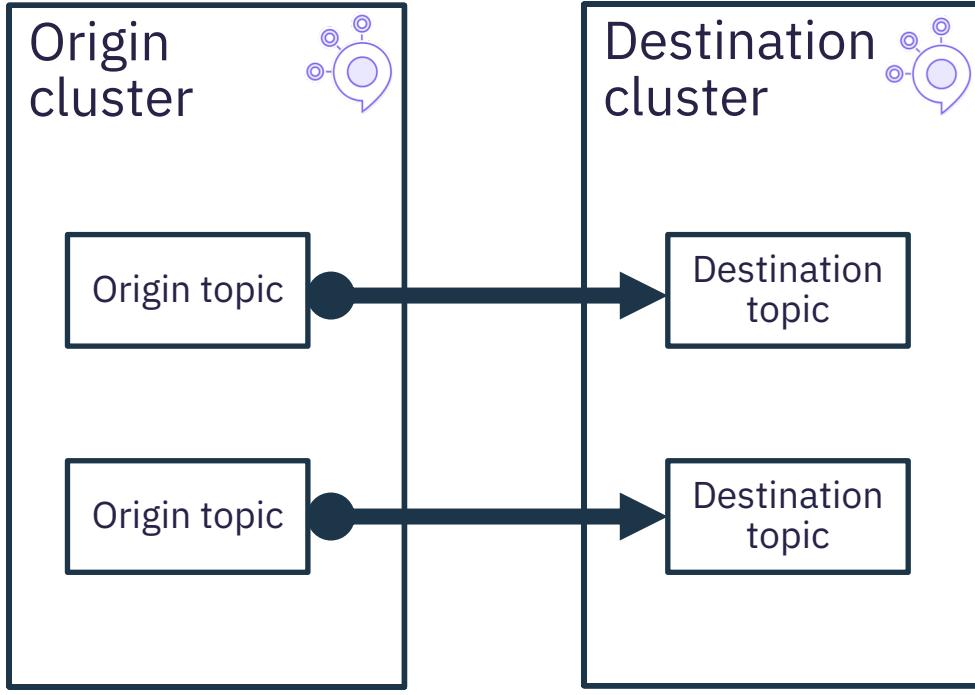


IBM Event Streams | Enterprise-Grade Reliability

Integrated geo-replication for disaster recovery



Geo-Replication makes Disaster Recovery simple



The screenshot shows the 'IBM Event Streams' interface. On the left, under 'rc4-repl', it lists 'Topics' with a search bar and a table:

Name	Replicas	Partitions
Airline_delays	3	1
Passenger_info	3	1
→ kit-ioi	3	2
→ rc4_steve	3	3
steve2	3	2

On the right, the 'Geo-replication' section shows 'Origin locations' (1 cluster geo-replicating topics to this cluster) and 'Destination locations' (You have not connected any destination clusters). A button 'Add destination cluster' is available. A note at the bottom says 'System is healthy'.

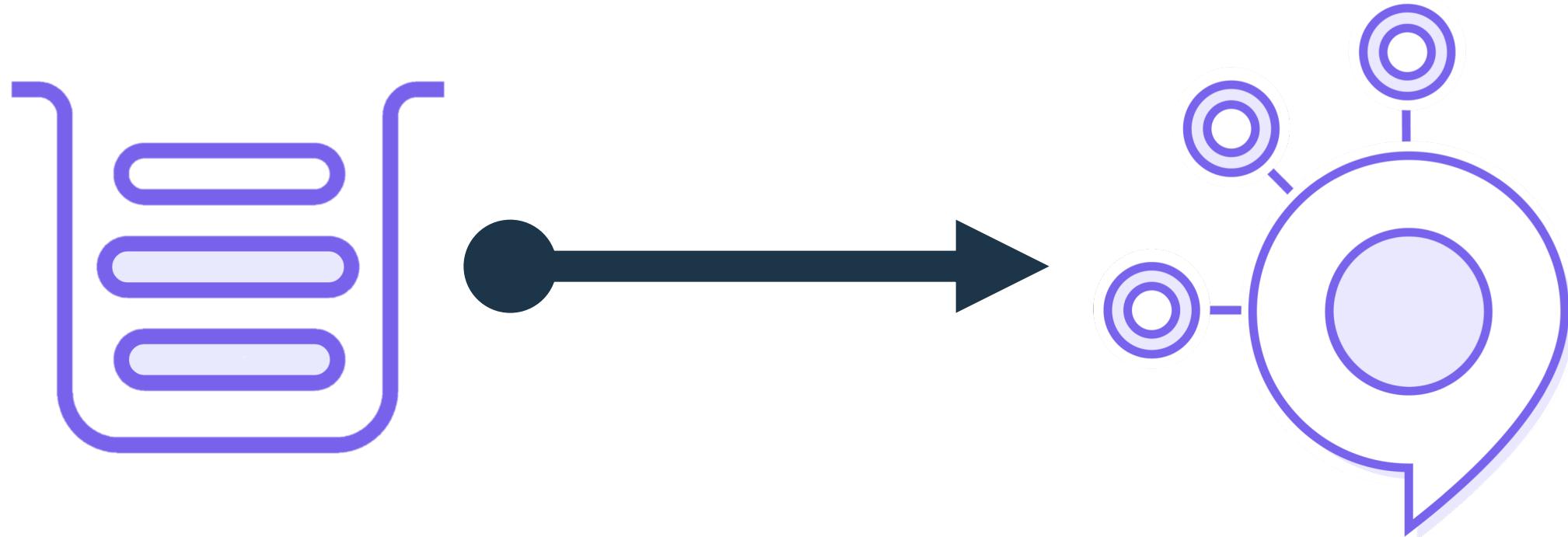
Target is take-over of workload on the destination cluster by business applications within 15 minutes

Easy configuration using the Event Streams UI from the origin cluster sets up the replicator and security credentials

At-least-once reliability so messages are not lost



IBM Event Streams | Connects to existing MQ backbone



Kafka Connect source connector for IBM MQ

Fully supported by IBM for customers with support entitlement for IBM Event Streams

IBM Event Streams | Ready for Mission-Critical Workloads



All with IBM 24x7 worldwide support

IBM has years of experience running Apache Kafka across the globe

