

# IBM Cloud Pak for Integration – MQ and Event Streaming



Shipra Jain  
Shipra\_jain@in.ibm.com

Developer Advocate  
Developer EcoSystem Group

Lab/Demo:

Karan Chaturvedi

Developer Advocate  
Developer EcoSystem Group



# Agenda

- **MQ modernization and containers**
- **Event Streaming**

# MQ modernization and containers

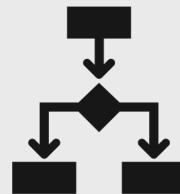


# IBM can help you modernize with *Cloud Pak for Integration*



## People & Process

- ✓ Decentralized ownership
- ✓ Empowering teams
- ✓ Agile methods



## Architecture

- ✓ Fine-grained deployment
- ✓ Cloud Agnostic
- ✓ Scalable and resilient



## Technology

- ✓ Cloud-native infrastructure
- ✓ Unified security, governance, and operations
- ✓ Flexible licensing model

## Development Agility

## Deployment Agility

## Operational Agility

## MQ in Containers, continually evolving

MQ first supported Docker containers in 2015, showing how a stateful solution can run in an often stateless world.



2015

[hub.docker.com/r/ibmcom/mq](https://hub.docker.com/r/ibmcom/mq)



MQ was one of the first certified containers available on IBM's Kubernetes platform, IBM Cloud Private. Showing how to run MQ in a managed container environment.

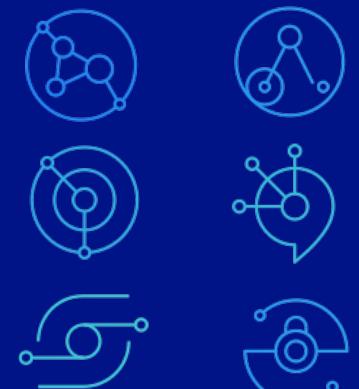
MQ added support for running on Red Hat OpenShift



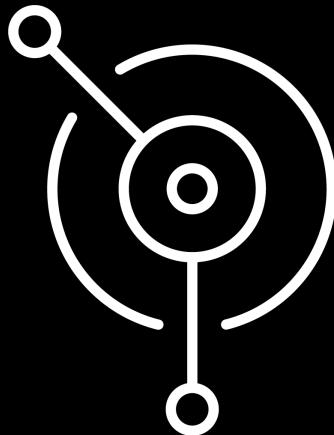
2019

[github.com/ibm-messaging/mq-container](https://github.com/ibm-messaging/mq-container)

MQ is a key component of IBM's Cloud Pak for Integration, providing enterprise messaging for the Integration Platform solution



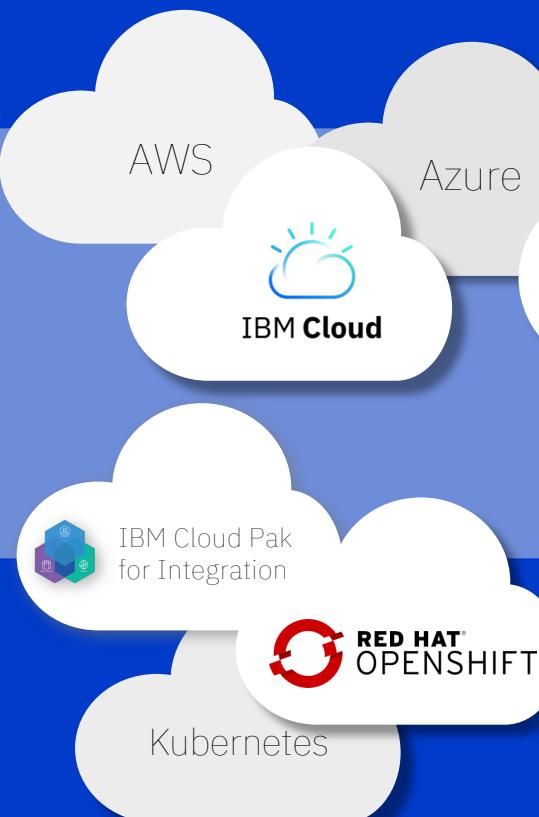
Run IBM MQ in any location or cloud, exactly as you need it



On-premise, software and the MQ Appliance



Run MQ yourself in public or private clouds



Let IBM host MQ for you with its managed SaaS MQ service in public clouds, IBM Cloud and AWS



# On public cloud, choose your preferred level of responsibility

On-premises software MQ	Virtual machine IaaS install	IBM Cloud Pak IaaS install	IBM Cloud Pak on managed RH OCP	MQ on Cloud service
Application	Application	Application	Application	Application
Clustering	Clustering	Clustering	Clustering	Clustering
Q / Msg monitoring	Q / Msg monitoring	Q / Msg monitoring	Q / Msg monitoring	Q / Msg monitoring
Queues, Topics	Queues, Topics	Queues, Topics	Queues, Topics	Queues, Topics
QM failover	QM failover	QM failover	QM failover	QM failover
QM availability/restart	QM availability/restart	QM availability/restart	QM availability/restart	QM availability/restart
MQ patching	MQ patching	MQ patching	MQ patching	MQ patching
OS patching	OS patching	OS patching	OS patching	OS patching
System monitoring	System monitoring	System monitoring	System monitoring	System monitoring
Virtual machine	Virtual machine	Virtual machine	Virtual machine	Virtual machine
Hypervisor	Hypervisor	Hypervisor	Hypervisor	Hypervisor
Storage	Storage	Storage	Storage	Storage
Servers	Servers	Servers	Servers	Servers
Networking	Networking	Networking	Networking	Networking
Data centre	Data centre	Data centre	Data centre	Data centre

Customer

IBM

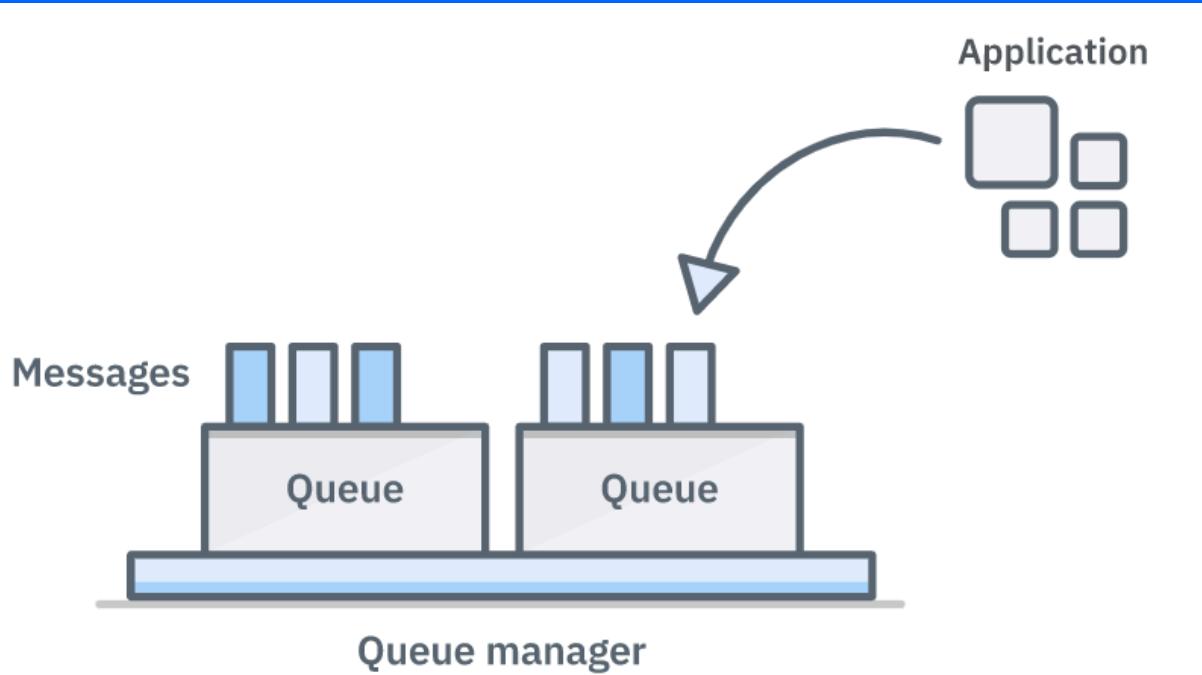
© 2020 IBM Corporation

# MQ Features

## **Salient Features of IBM MQ:**

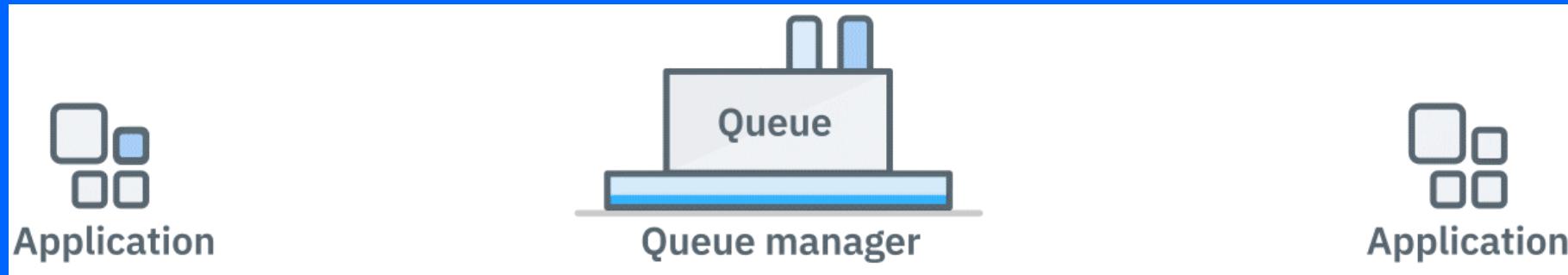
- Once-and-once-only delivery
- End-to-end encryption
- Synchronous/asynchronous messaging
- Comprehensive interfaces and languages
- Replicated data queue managers
- Advanced file transfer, blockchain integration

# MQ Essentials



1. **Messages** – The packets of data produced and consumed by applications.
2. **Queue** – An addressable location where messages are delivered and stored reliably until they need to be consumed.
3. **Queue manager** – This is the actual MQ engine, the server that hosts the queues.

# Message Queues Decouple Applications



Placing a message queue between two or more applications means that the applications are not communicating directly.

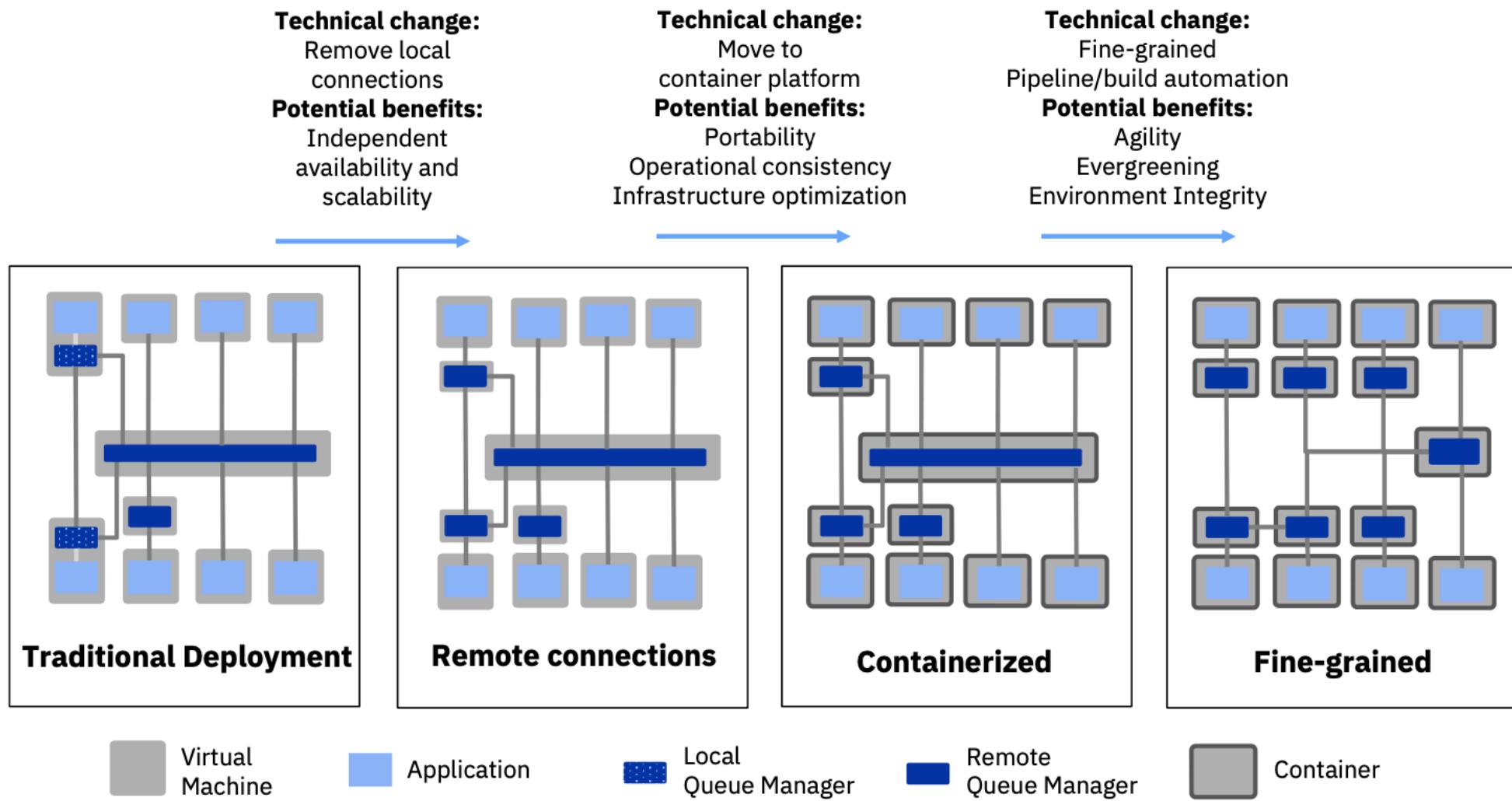
Whether one application goes away, or the consuming app is unable to keep up with a temporary spike in requests, message queuing provides a shock absorber that can deal with any issues on either side.

As a result, adopting messaging means that applications do not have to be available at the same time, as the queue provides a neutral place where they can exchange data.

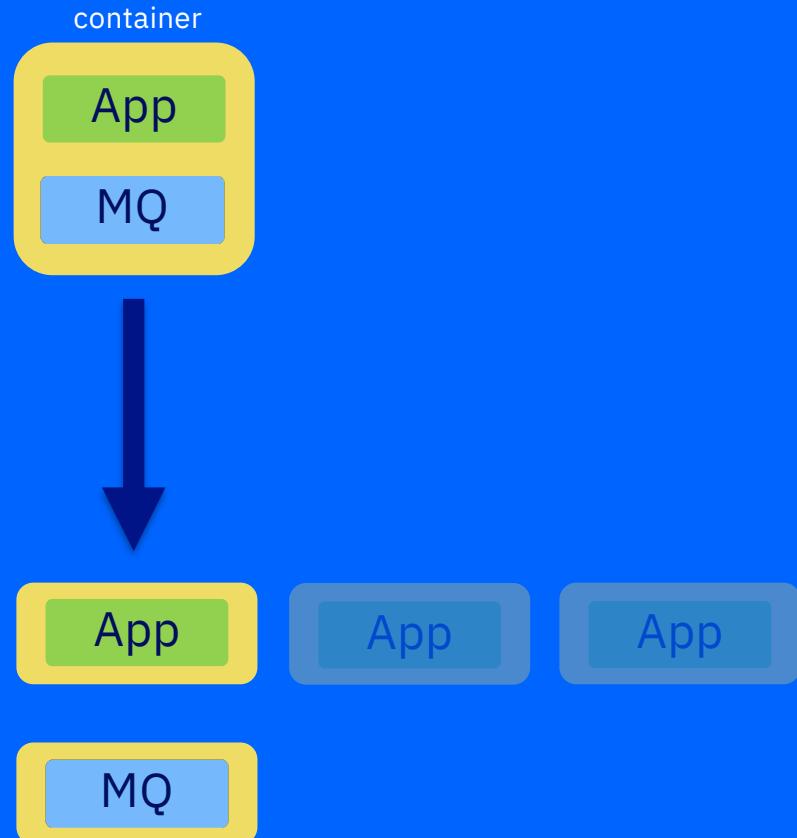
This model is known as **asynchronous messaging**.

<b>Language</b>	<b>API</b>	<b>Protocol</b>
Java	JMS	MQ
Java	MQTT	MQTT (1)
Java	REST (2)	HTTP/HTTPS
Java	MQ Light (3)	AMQP 1.0 (4)
C# (.NET)	XMS	MQ
C# (.NET)	MQ Object Oriented	MQ
Python	MQ Light (3)	AMQP 1.0 (4)
Python	MQI with pymqi (5)	MQ
Python	REST (2)	HTTP/HTTPS
Node.js	MQI	MQ
Node.js	REST (2)	HTTP/HTTPS
Node.js	MQ Light (3)	AMQP 1.0 (4)
Ruby	MQ Light (3)	AMQP 1.0 (4)
C++	XMS	MQ
C++	MQI	MQ
C++	MQ Object Oriented	MQ
C	MQI	MQ
C	XMS	MQ
COBOL	MQI	MQ
HLASM (High Level Assembler)	MQI	MQ
PL/1	MQI	MQ
pTAL	MQI	MQ
RPG – IBMi	MQI	MQ

# Key technical considerations for IBM MQ in Containers



# Single concern per container



## Steps

- Queue manager in one container, application in others
- Only use client connections, not local bindings

## Benefits

- Scale independently
- Patch and update independently
- Simpler, more intuitive health checking
- Promotes reuse of container images
- Easier to retrieve logs, do problem determination

# Small, decentralized queue managers

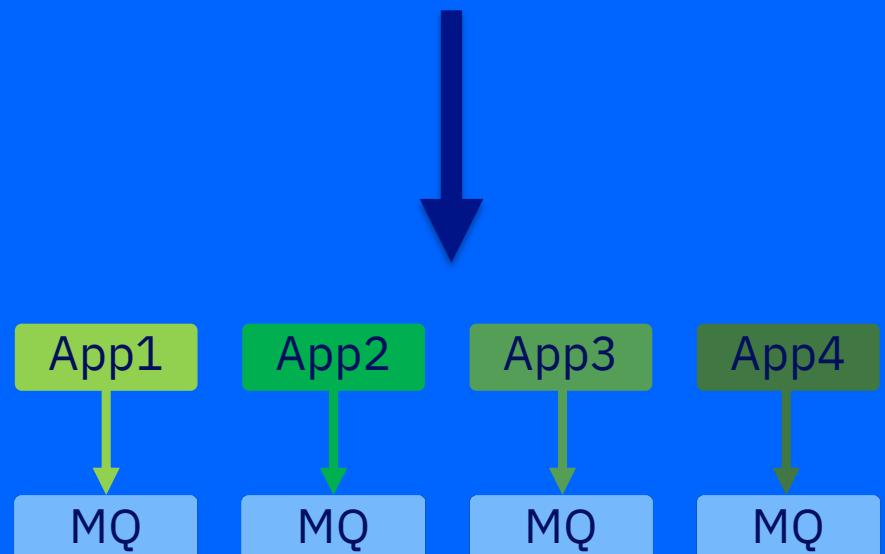
## Steps

- Queue manager per application instead of shared by multiple applications
- Right-size QM to meet the individual application workload

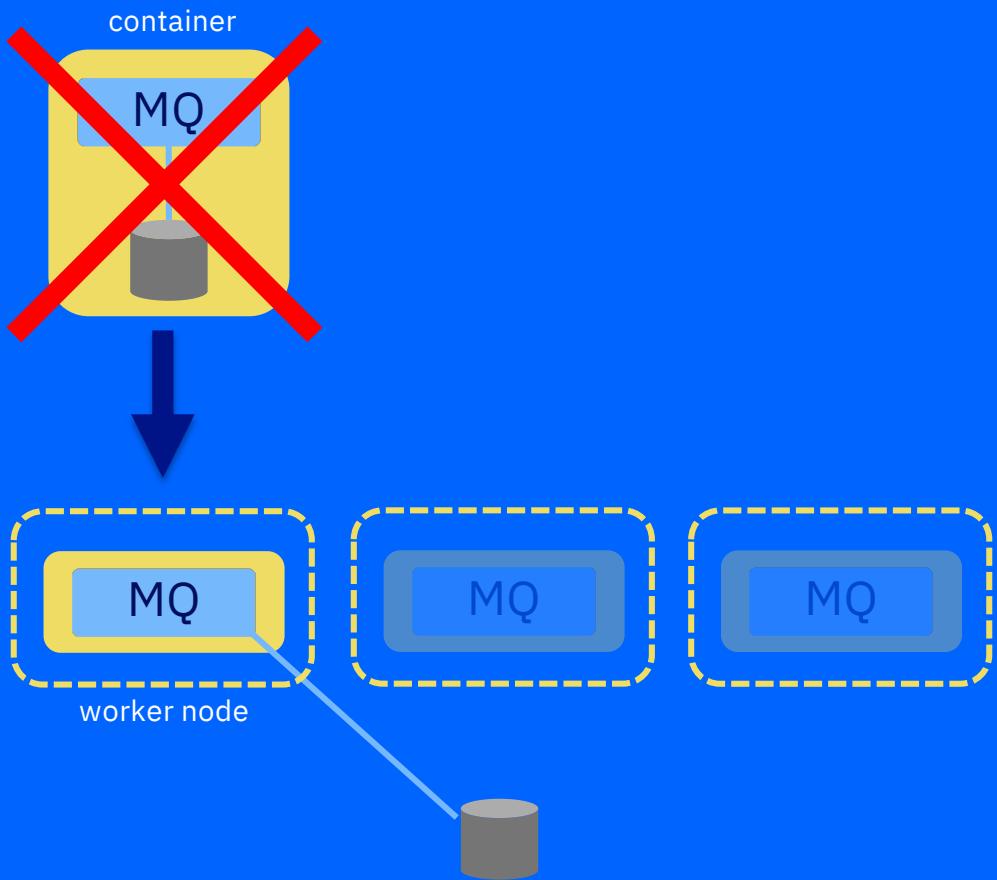


## Benefits

- More agile teams with direct ownership
- Fine grained resilience to failures or outages
- Reduce app/QM version compatibility problems
- Easier to schedule maintenance windows



# Data availability using external storage



## Steps

- Store queue manager data on Persistent Volume outside the container

## Benefits

- Config / message data survives container restart
- Enables failover to other workers in the cluster
- Simplifies applying upgrades or fixes

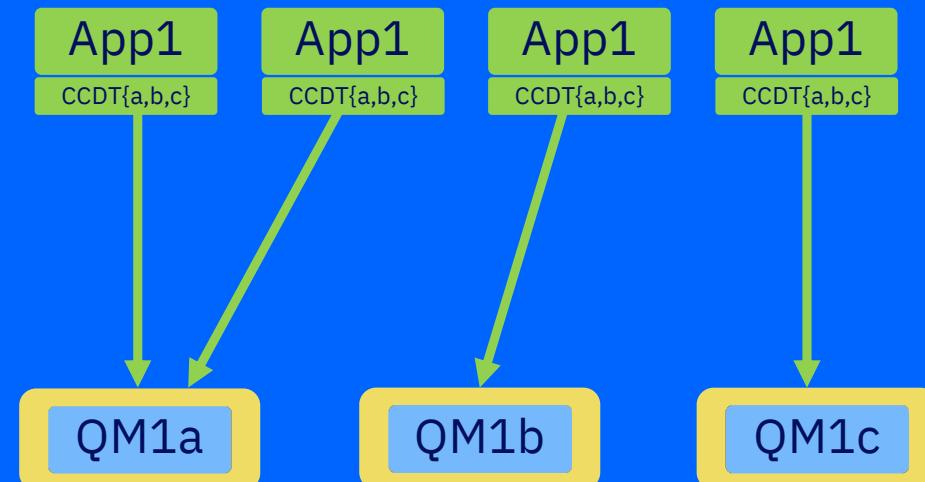
# Continuous availability through horizontal scaling

## Steps

- Deploy two or more equivalent queue managers
- Distribute workload across the available set using connection routing features such as CCDT, ConnectionNameList or external load balancing

## Benefits

- Zero downtime for planned upgrades
- Continue to serve new workload in the event of a failure / failover
- Scale based on workload requirements



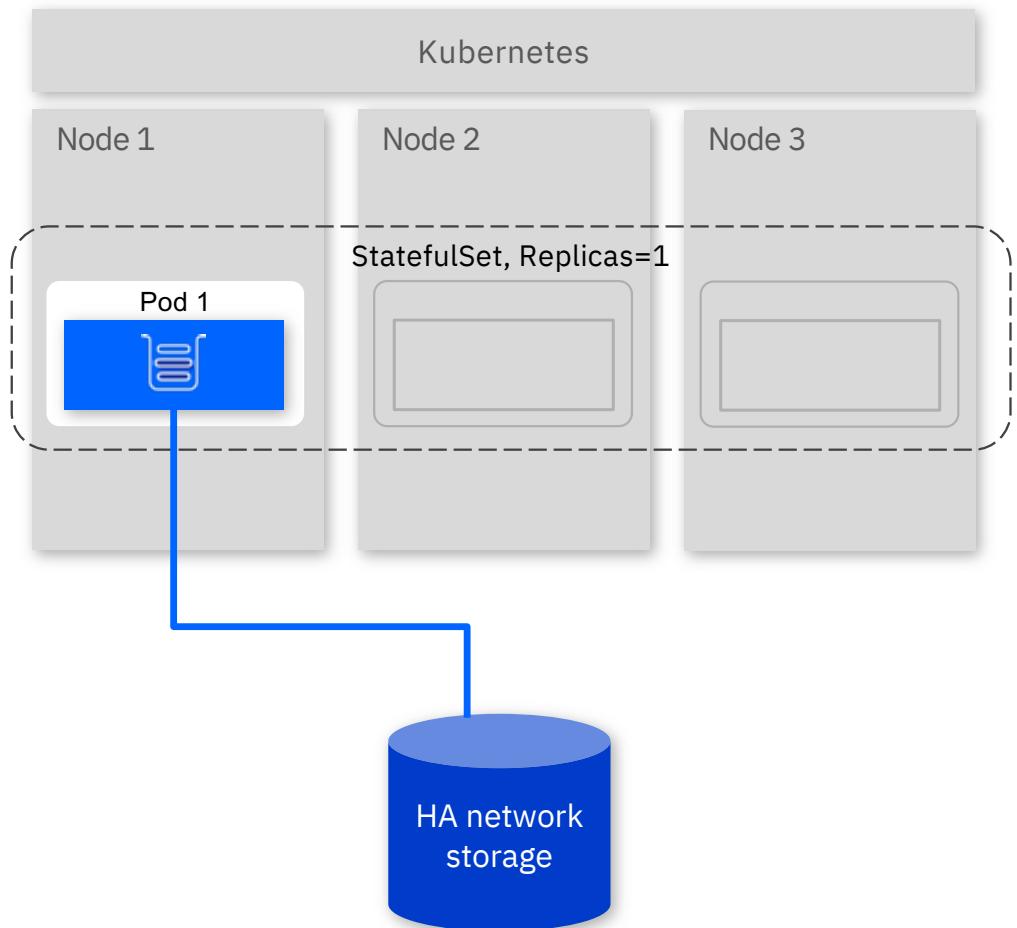
# High availability in containers

## High Availability with Kubernetes

High availability of the MQ data requires highly available replicated storage

Container orchestrators such as Kubernetes handle much of the monitoring and restart responsibilities...

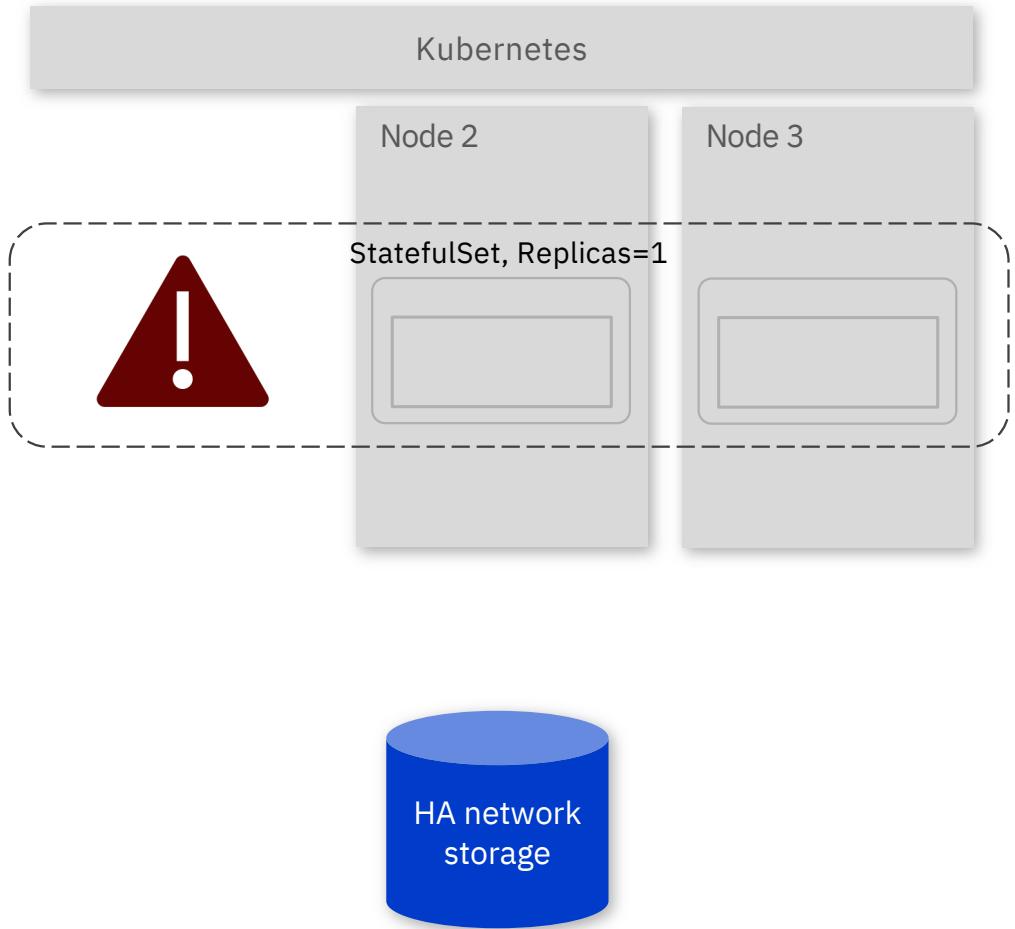
Note: The RDQM solution does not apply to container environments



## High Availability with Kubernetes

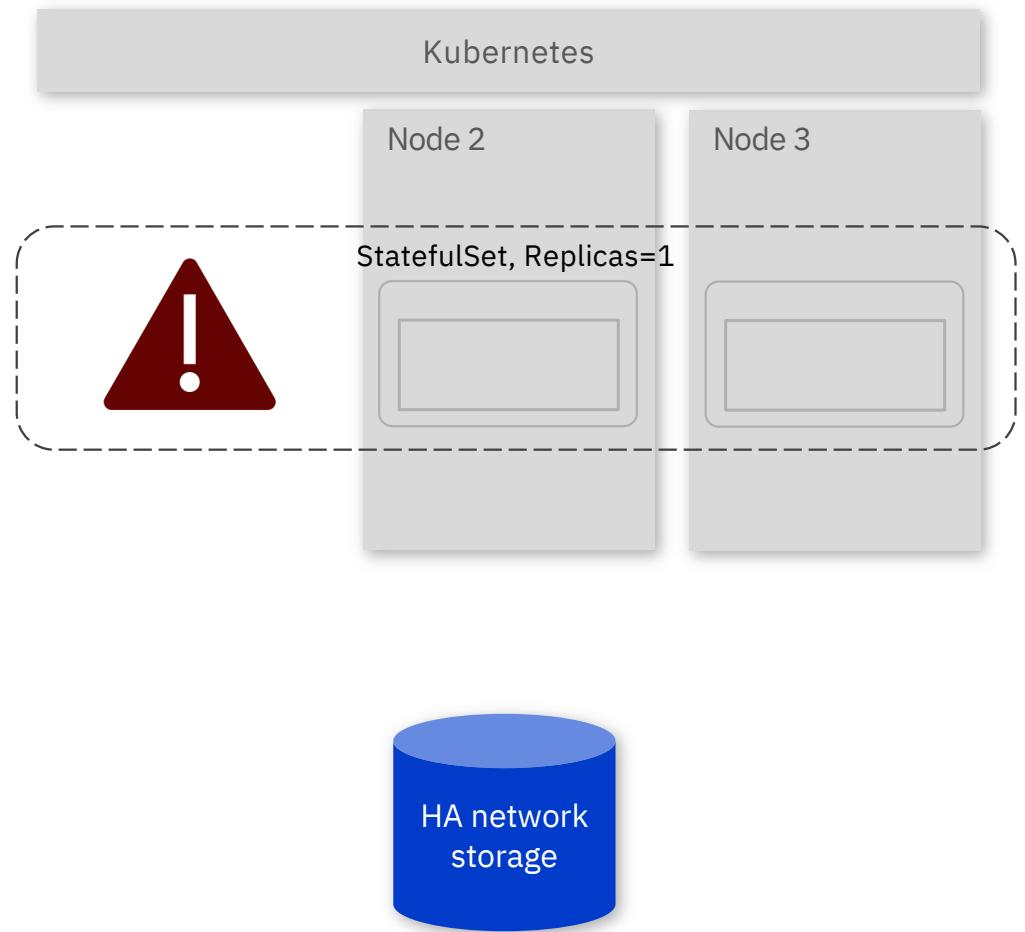
High availability of the MQ data requires highly available replicated storage

Container orchestrators such as Kubernetes handle much of the monitoring and restart responsibilities...



# High Availability with Kubernetes

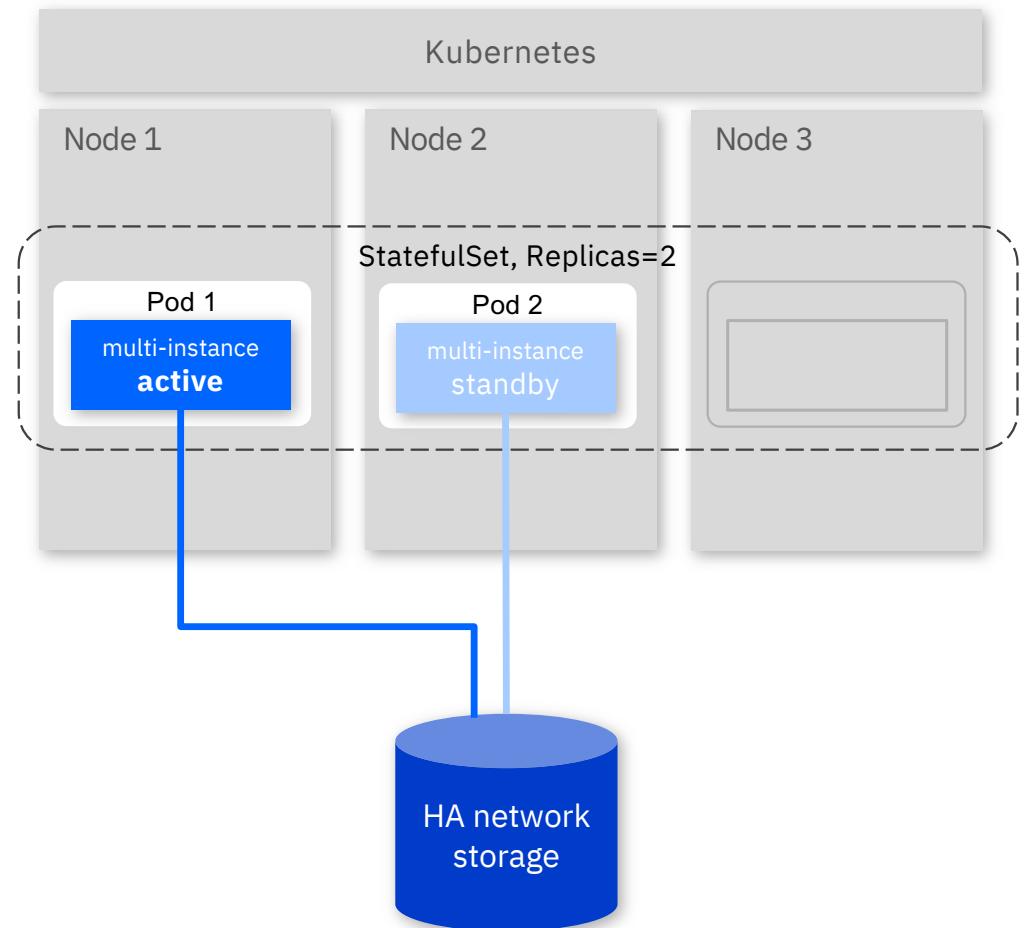
...but not all. StatefulSets such as MQ are not automatically restarted following a Kubernetes node failure



# High Availability with Kubernetes

IBM MQ 9.1.3 CD

The MQ container image and Certified Container now supports a two-replica multi-instance queue manager deployment pattern to handle Kubernetes node failures



## File systems for multi-instance QMs

Multi-instance queue managers rely on shared file systems accessed by both pods concurrently (“ReadWriteMany” in Kubernetes terminology)  
Implies File storage and not Block storage.

Additionally, requires specific file locking behavior in order to work correctly.

## Not all file systems are equal / usable

The following file systems are known not to work as they do not meet IBM MQ's technical requirements:

- Network File System (NFS) version 3 - does not provide lease-based file locking.
- Red Hat Global File System (GFS, or GFS1) - does not provide the correct locking semantics.
- Oracle Cluster File System version 2 (OCFS2) - does not provide the correct locking semantics in version 1.4.
- Oracle ASM Cluster File System (ACFS) - does not provide the correct locking semantics.
- Gluster V3.x, V4.x, and V5.x, using the Gluster Native (FUSE) client

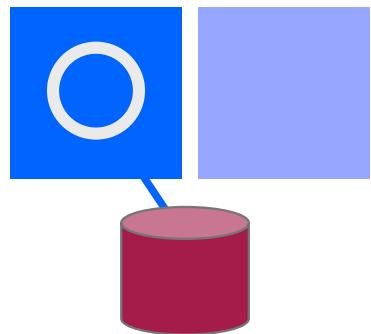
The following file systems have been tested by IBM MQ and have been found to meet IBM MQ's technical requirements:

- IBM AIX 5.3 TL10 NFS v4 server<sup>123457</sup>
- IBM General Parallel File System 3.2.1
- IBM General Parallel File System 3.4.0
- IBM i5/OS NetServer V6R1
- IBM System Storage N series Data ONTAP 7.3.2 NFS v4 server<sup>123457</sup>
- Microsoft Windows 8<sup>6</sup>
- Microsoft Windows Server 2008<sup>6</sup>
- Microsoft Windows Server 2012<sup>6</sup>
- Red Hat Enterprise Linux 5.3 NFS v4 server<sup>123457</sup>
- Red Hat Enterprise Linux 6.5 NFS v4 server<sup>123457</sup>
- Red Hat Global File System 2 (GFS2)
- SUSE Linux Enterprise Server 10 NFS v4 server<sup>123457</sup>
- Veritas Storage Foundation V5.0 MP3 RP3 Cluster File System
- Veritas Storage Foundation V5.1 SP1 Cluster File System

The following systems have not been fully tested, and some important considerations have been identified - see notes for more information:

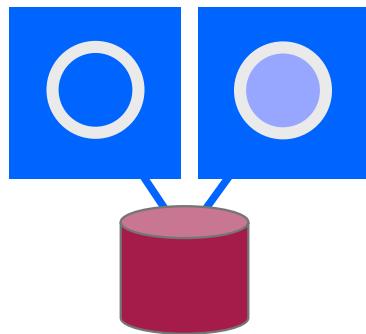
- AWS EFS - Subject to locking considerations<sup>8</sup>

# MQ high availability options



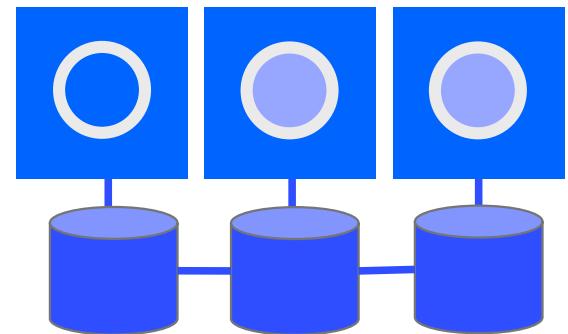
## Single resilient queue manager

- Cloud manages fail-over to somewhere with spare capacity
- Networked storage (block or filesystem), managed by separate subsystem



## Multi-instance queue manager

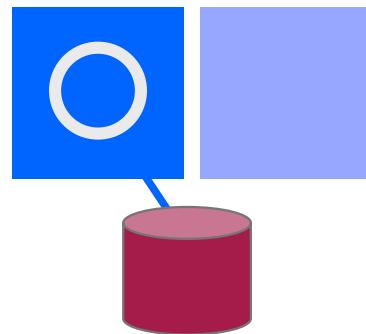
- MQ manages fail-over
- Networked storage (filesystem), managed by separate subsystem



## Replicated data queue manager

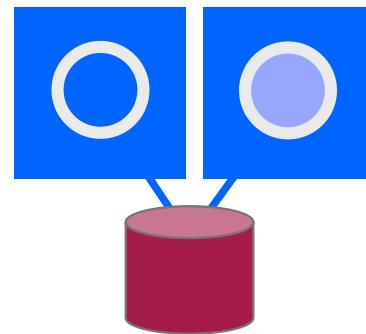
- MQ manages fail-over
- Local block storage, synchronously replicated by MQ

# MQ high availability options - containers



## Single resilient queue manager

- Beware of five minute delay in Kubernetes, in the case of a complete node failure (pod-eviction-timeout feature).
- Don't just rely on Kubernetes Stateful Sets



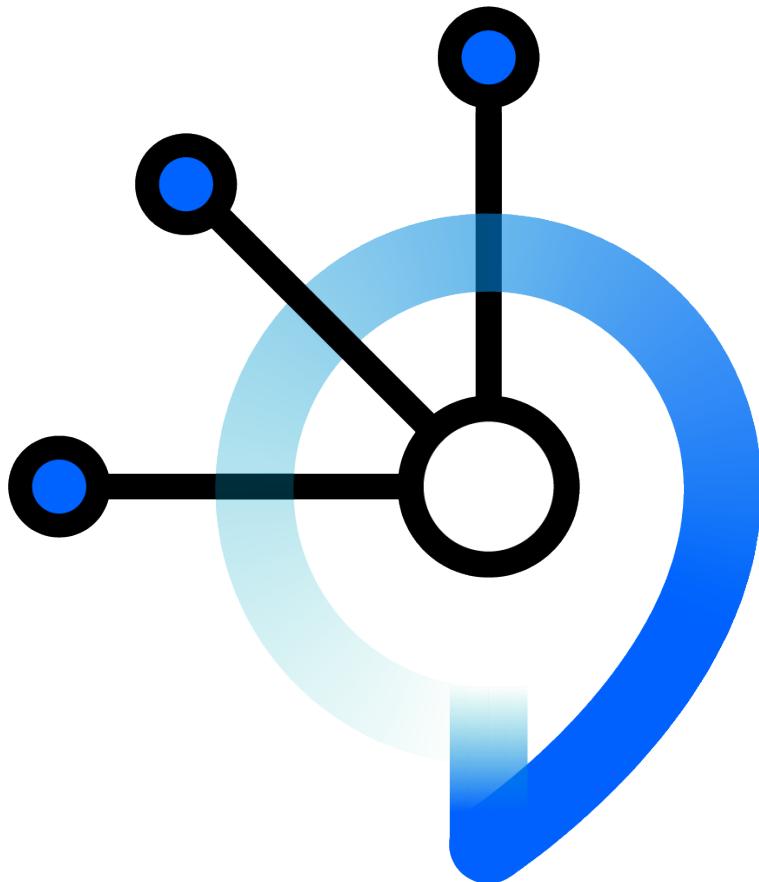
## Multi-instance queue manager

- Not supported by current IBM Helm charts
- Double the compute cost



## Replicated data queue manager

- Unsuitable for most public cloud container services, because it requires changes to container host (kernel module)
- Untested and unsupported in containers



# IBM Event Streams

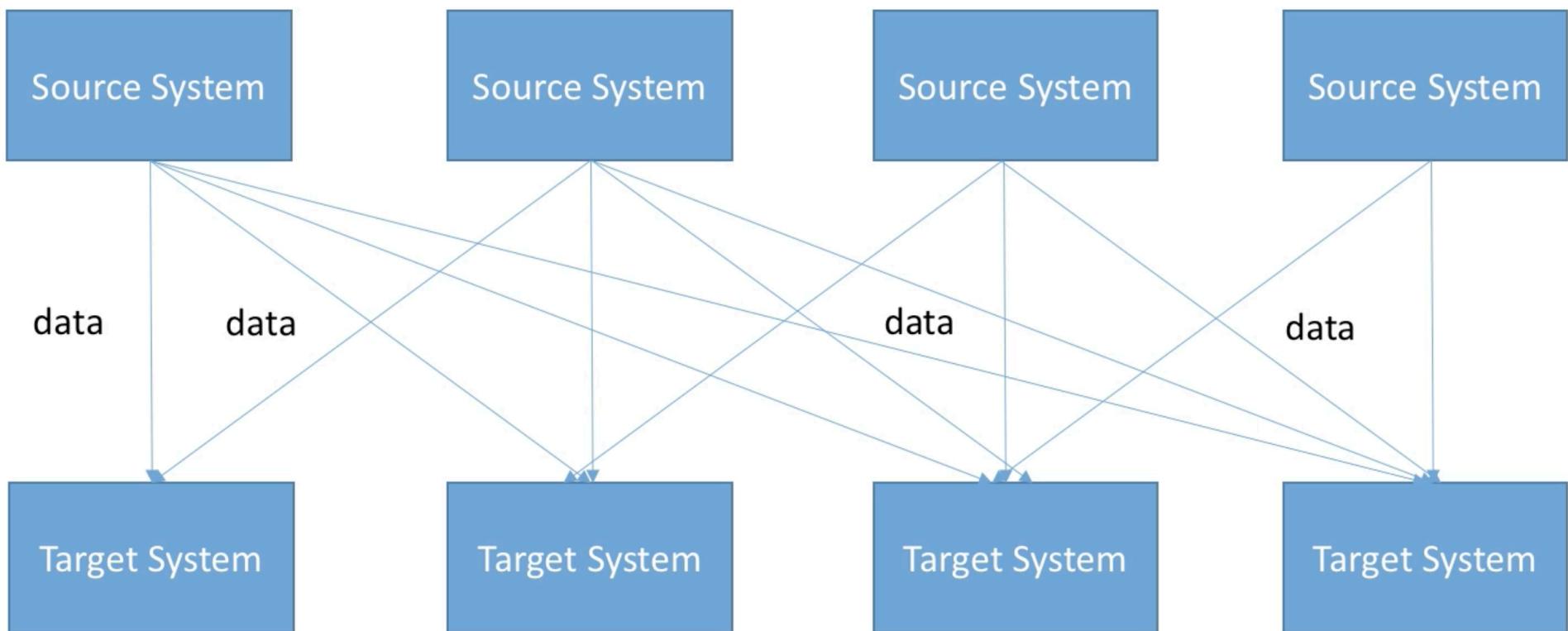
Apache Kafka® for the Enterprise

*A core part of Cloud Pak for Integration*

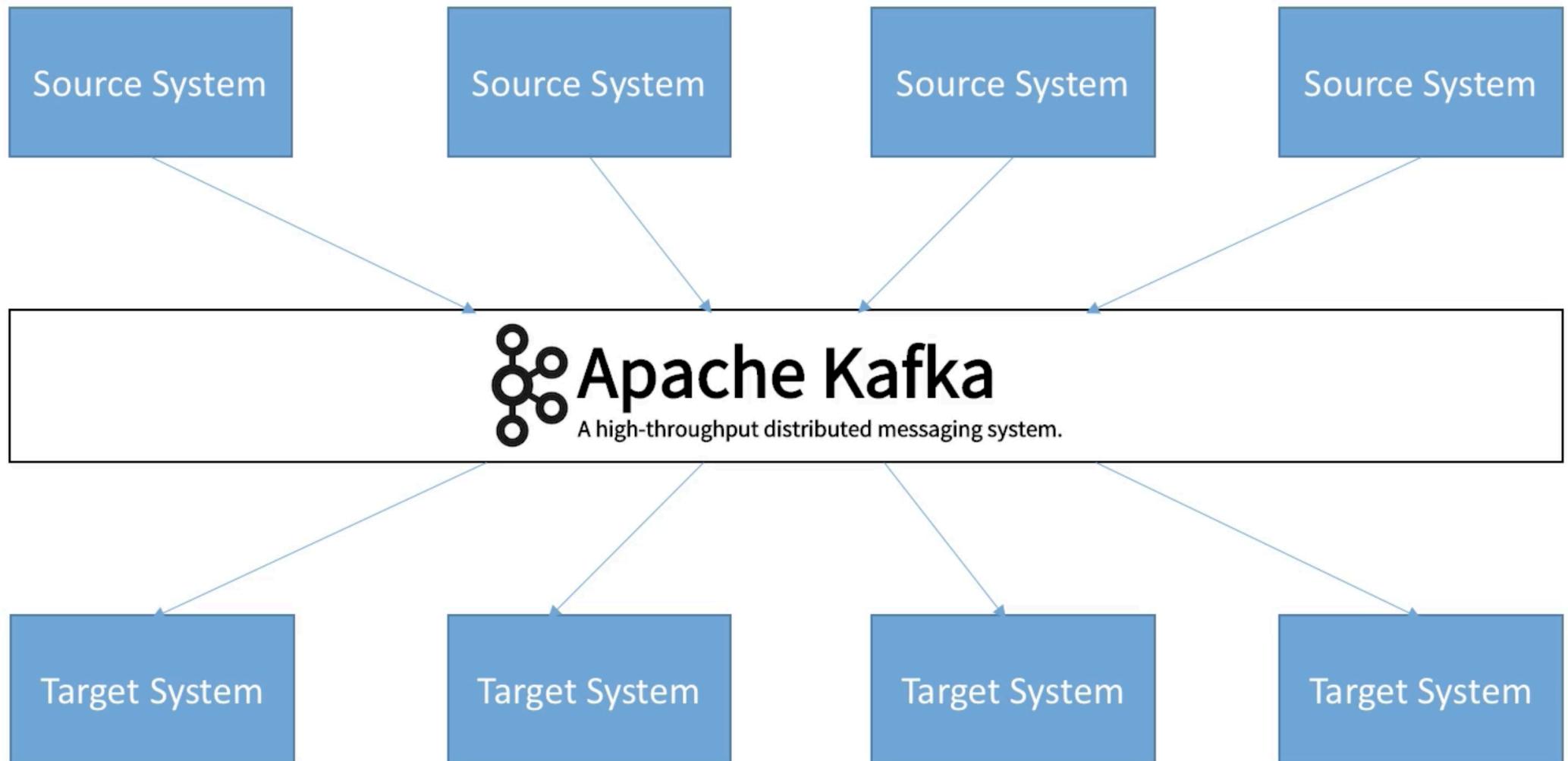
Technical Overview

Developer Advocates  
Tech talk : Shipra Jain

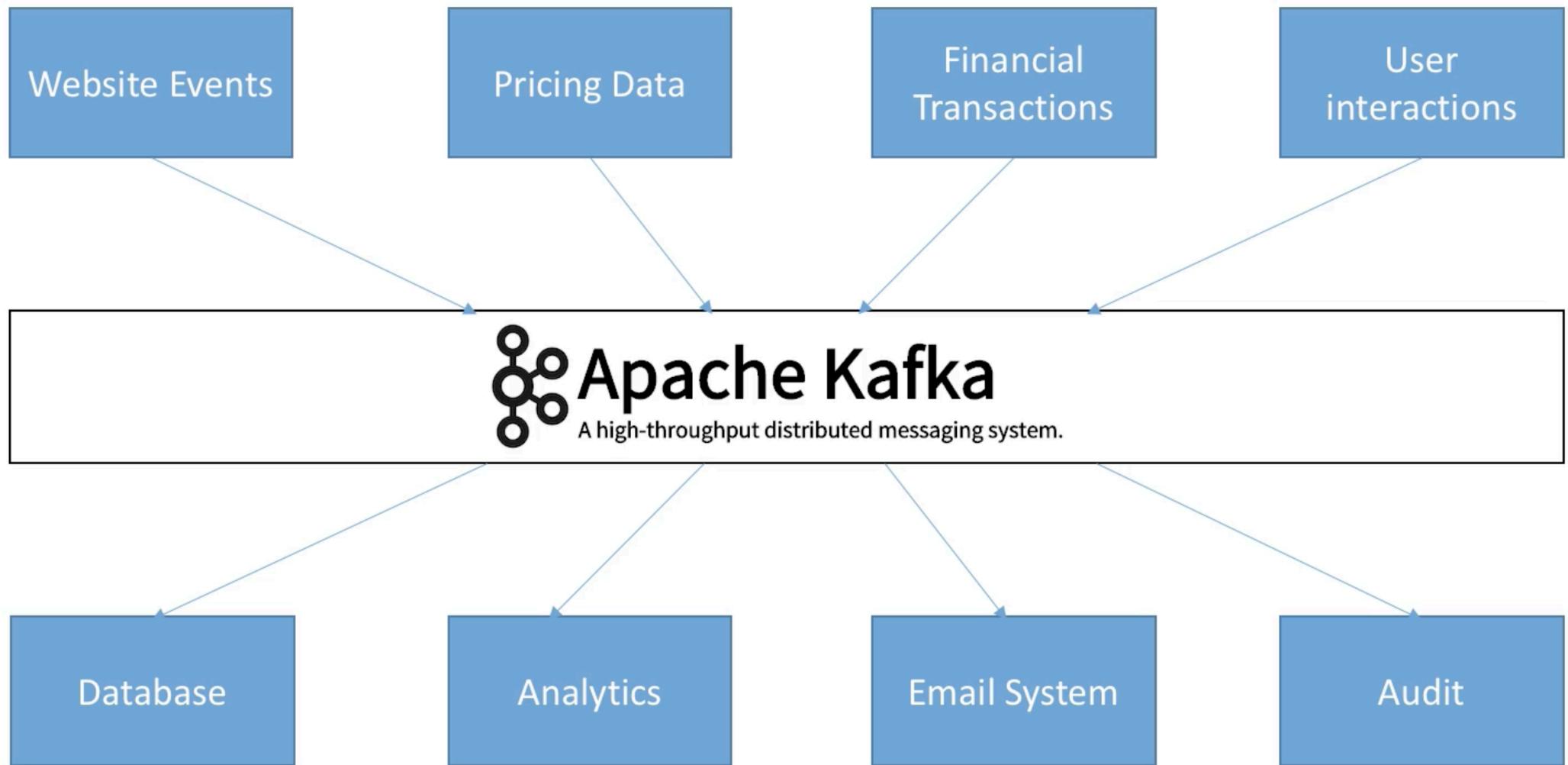
# Integration



## Where Kafka fits



## Examples



## Why Kafka?

- Created by LinkedIn, now Open Source Project mainly maintained by Confluent
- Distributed, resilient architecture, fault tolerant
- Horizontal scalability:
  - Can scale to 100s of brokers
  - Can scale to millions of messages per second
- High performance (latency of less than 10ms) – real time
- Used by the 2000+ firms, 35% of the Fortune 500:



**NETFLIX**

**LinkedIn** TM

**UBER**

**Walmart** The Walmart logo, consisting of the word 'Walmart' in blue and a yellow five-pointed starburst symbol.



## Kafka - Use Cases

- Messaging System
- Activity Tracking
- Gather metrics from many different locations
- Application Logs gathering
- Stream processing (with the Kafka Streams API or Spark for example)
- De-coupling of system dependencies
- Integration with Spark, Flink, Storm, Hadoop, and many other Big Data technologies



## Examples

- **Netflix** uses Kafka to apply recommendations in real-time while you're watching TV shows
- **Uber** uses Kafka to gather user, taxi and trip data in real-time to compute and forecast demand, and compute surge pricing in real-time
- **LinkedIn** uses Kafka to prevent spam, collect user interactions to make better connection recommendations in real time.
- Remember that Kafka is only used as a transportation mechanism!



# IBM – Apache Kafka

- Some vendors provide commercially supported versions of Kafka;
- Our implementation is IBM Event Streams.
- IBM is also a key contributor to the Apache Kafka open source project as committers.





**IBM Event Streams** is fully supported Apache Kafka® with value-add capabilities

### IBM Event Streams

- Award-Winning User Experience
- Powerful Ops Tooling
- Schema Registry
- Geo-replication for DR
- Connector Catalog
- Unrivalled MQ connectivity
- 24 x 7 Support

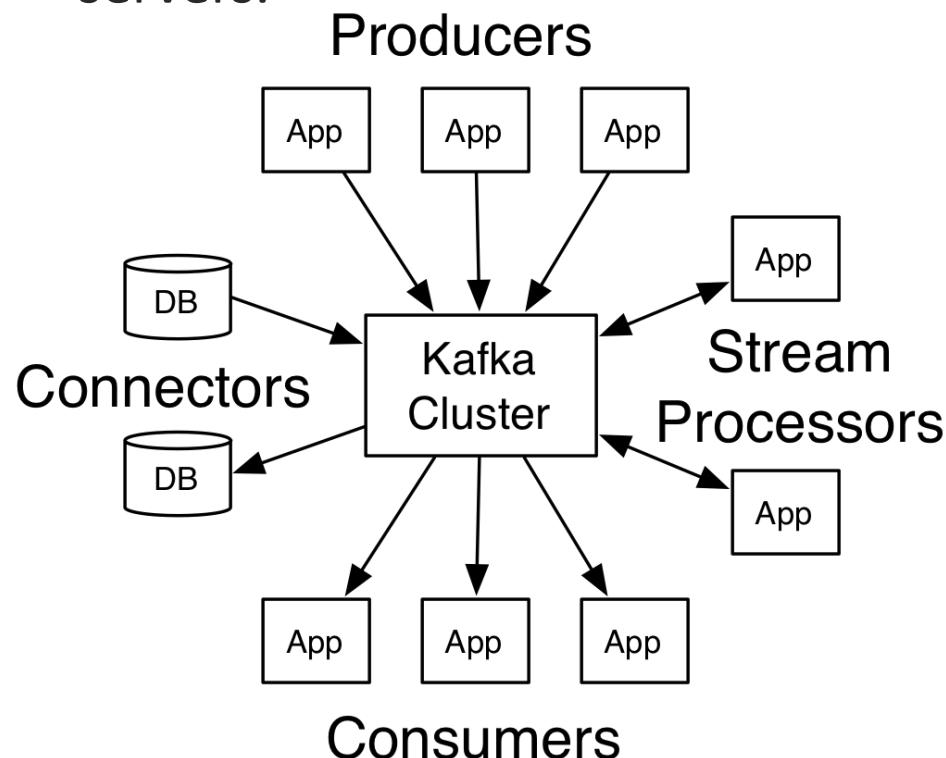


▷ INDIGO

. SILVER WINNER .

# What is Apache Kafka?

Apache Kafka is a ***publish-subscribe*** messaging system which lets you send messages between processes, applications, and servers.



Providers/Producers: Emits and publishes events

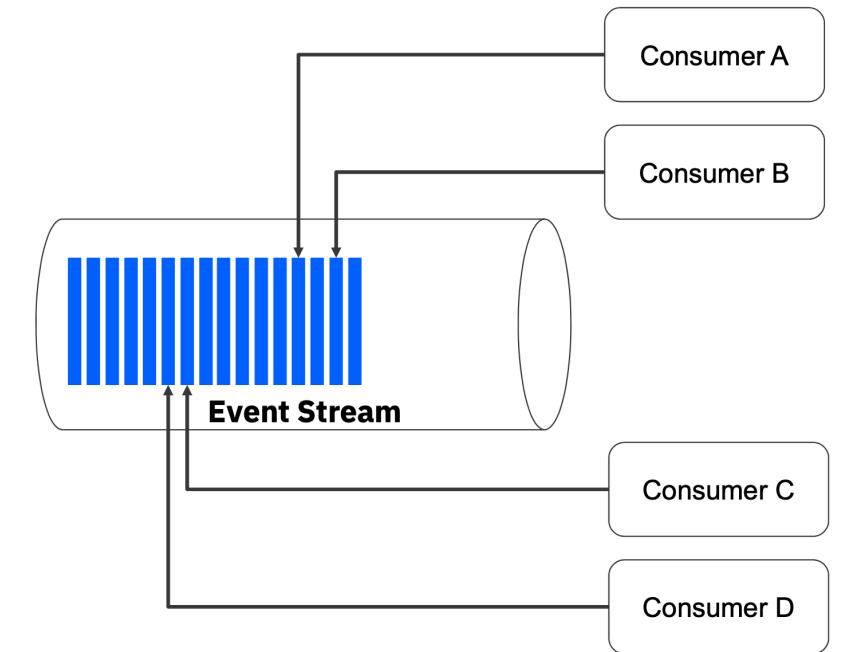
A server that stores events and manages the topics

Client that allows applications to interact as a provider or consumer.

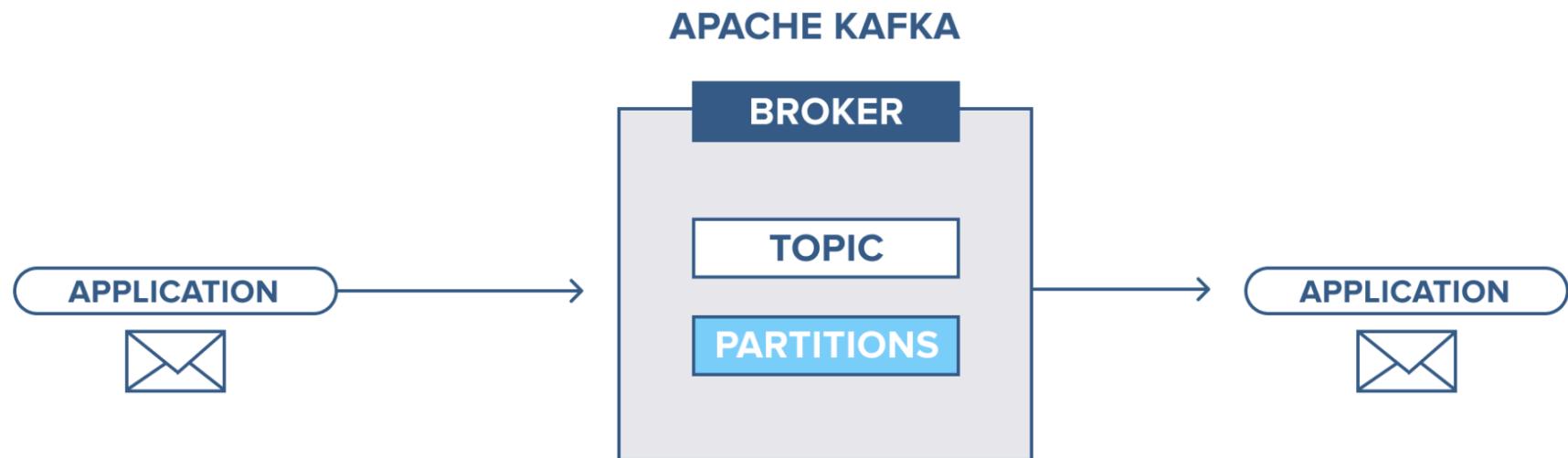
Consumers/Subscribers: Read events

# Capabilities for an event streaming technology

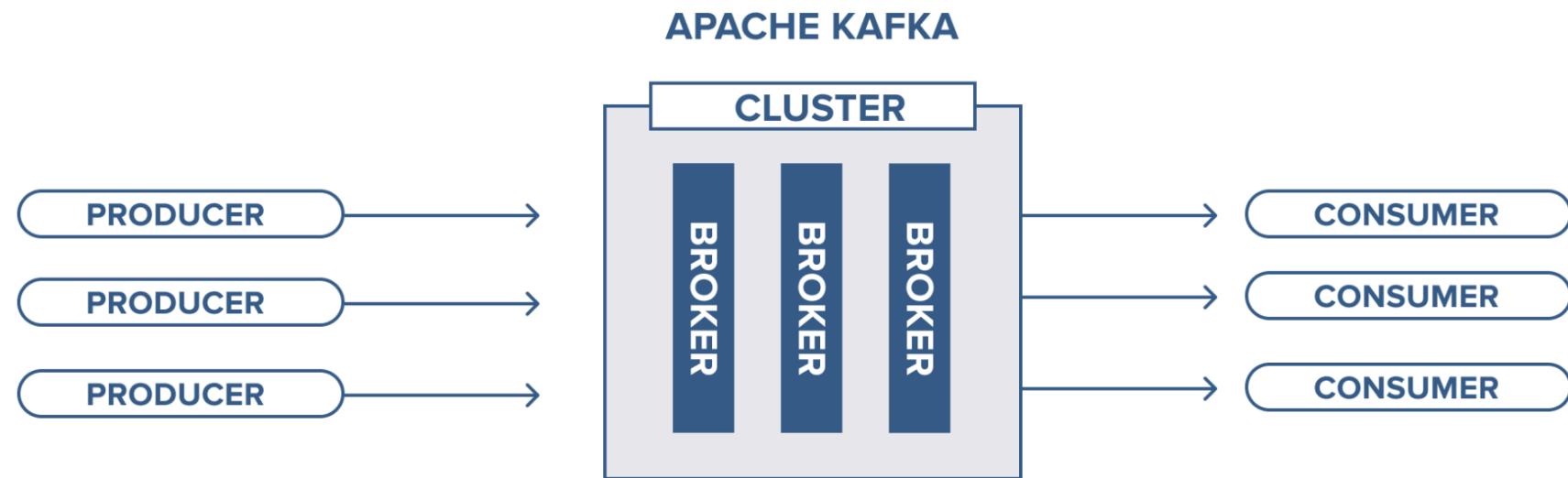
- Stream history : By using Topics The topic can be considered a stream history of all the events that are emitted, and allows subscribers to rewind to different locations of the topic
- High performance : designed to handle millions of events a second
- Scalable subscription: increases in the number of subscribers to a topic has a minimal impact on the resource
- Decoupled communication: provides an intermediary between the two applications, which means that they are decoupled



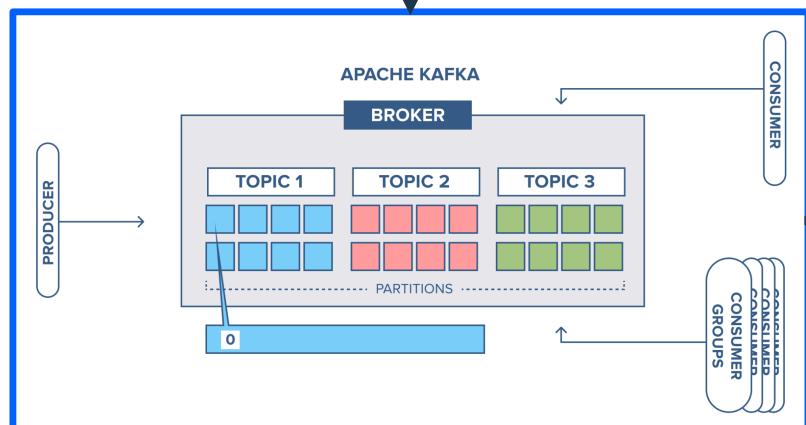
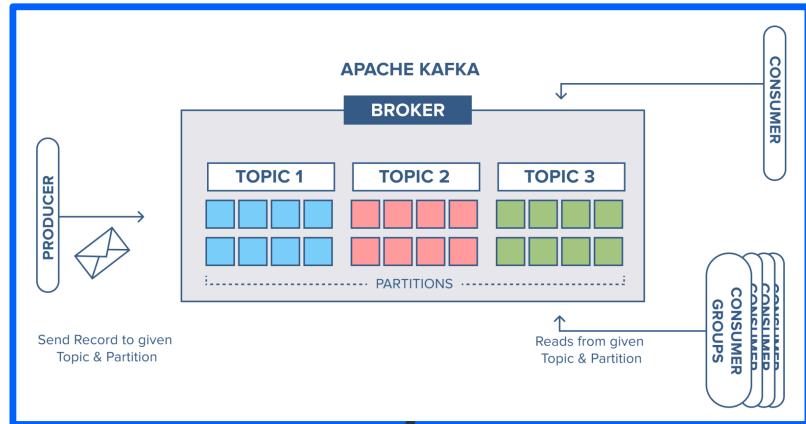
# Publish-subscribe durable messaging system



# Kafka Broker



# Record Flow in Kafka



**Scenario:** 3 topics, where each topic has 8 partitions.

The producer sends a record to partition 1 in topic 1

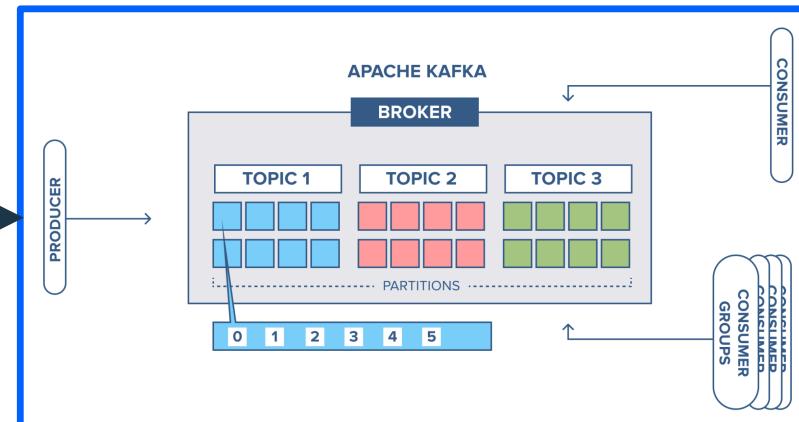
and since the partition is empty the record ends up at offset 0.

Next record is added to partition 1 will end up at offset 1, and the next record at offset 2 and so on.

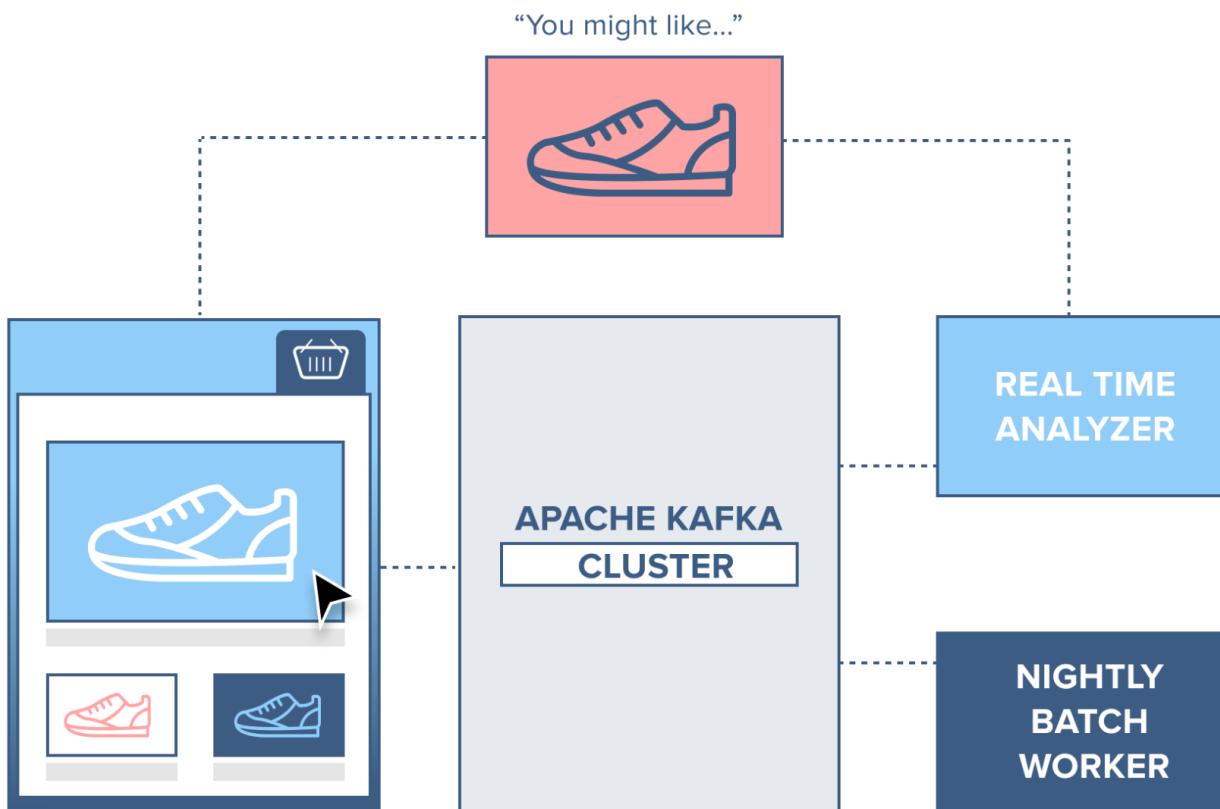
This is referred as Commit Log

Each record is appended to the log and there is no way to change the existing records in the log.

This is also the same offset that the consumer uses to specify where to start reading.



# Web Shop



, each action performed by a consumer is recorded and sent to Kafka.

A separate application comes along and consumes these messages, filtering out the products the consumer has shown an interest in and gathering information on similar products.

This 'similar product' information is then sent back to the webshop for it to display to the consumer in real-time.

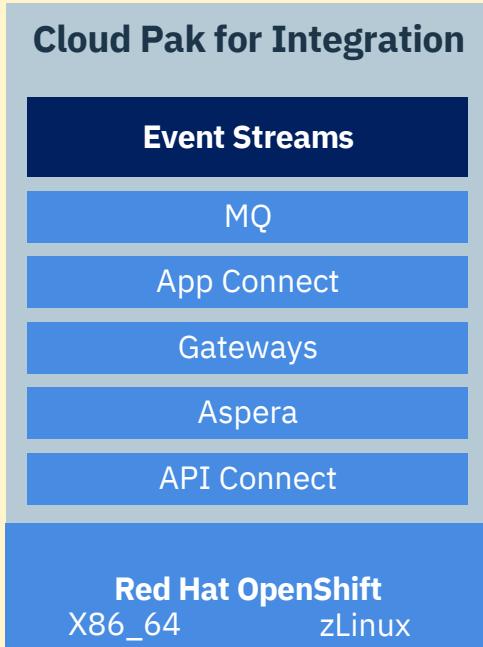
Or

data is persistent in Kafka, a batch job can run overnight on the 'similar product' information gathered by the system, generating an email for the customer with suggestions of products



# Packaging and Deployment Options to Suit Different Needs

## Self Managed Software



## Fully Managed Service



Hosted service on IBM public cloud

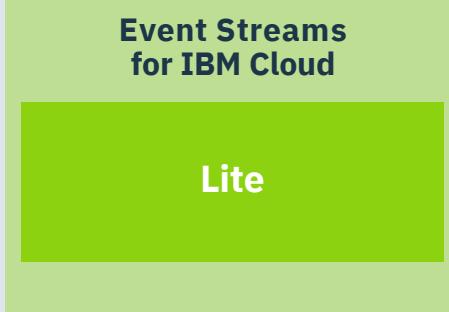
# Getting started

## Self Managed Software



<https://strimzi.io>

## Fully Managed Service



Hosted service on IBM public cloud

**Get Started at:** [ibm.com/cloud/event-streams/get-started](https://ibm.com/cloud/event-streams/get-started)

# Building on the Red Hat OpenShift Container Platform



Develop, deploy, and manage existing and container-based apps seamlessly across physical, virtual, and public cloud infrastructures



# Operators for Kafka



- **Kubernetes** allows you to automatically deploy and run workloads



- **Operators** extends that to be able to control stateful application resources in the same way



*Allows the system to behave like a human operator that has deep knowledge of how the system ought to behave*

Manage  
stateful apps

Reduce  
Complexity

Kubernetes  
native

Failure  
recovery

Scaling

Manage  
updates

- Describe declaratively what the state of the system should be, and the system automatically responds to achieve the desired state
- For Kafka that means you can create topics, control partitioning and replication from a `kubectl` command
- Deploy an end-to-end application with all its resources from a single `kubectl apply`



# Operators for Kafka



- Operator is responsible for managing all Event Streams resources
  - Specialised operator is able to provide much better control and information than a general purpose tool like Helm
  - Operator understands Kafka concepts like users, topics and partitions
- All the well established Kubernetes tools and techniques can now be applied to Kafka
- Put all resources under change control in Kubernetes manifests
  - Infrastructure as code

*Manage  
stateful apps*

*Reduce  
complexity*

*Kubernetes  
native*

*Failure  
recovery*

*Scaling*

*Manage  
updates*



# Apache Kafka orchestrated with Kubernetes

- **Highly available by design**

- Brokers are spread across 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



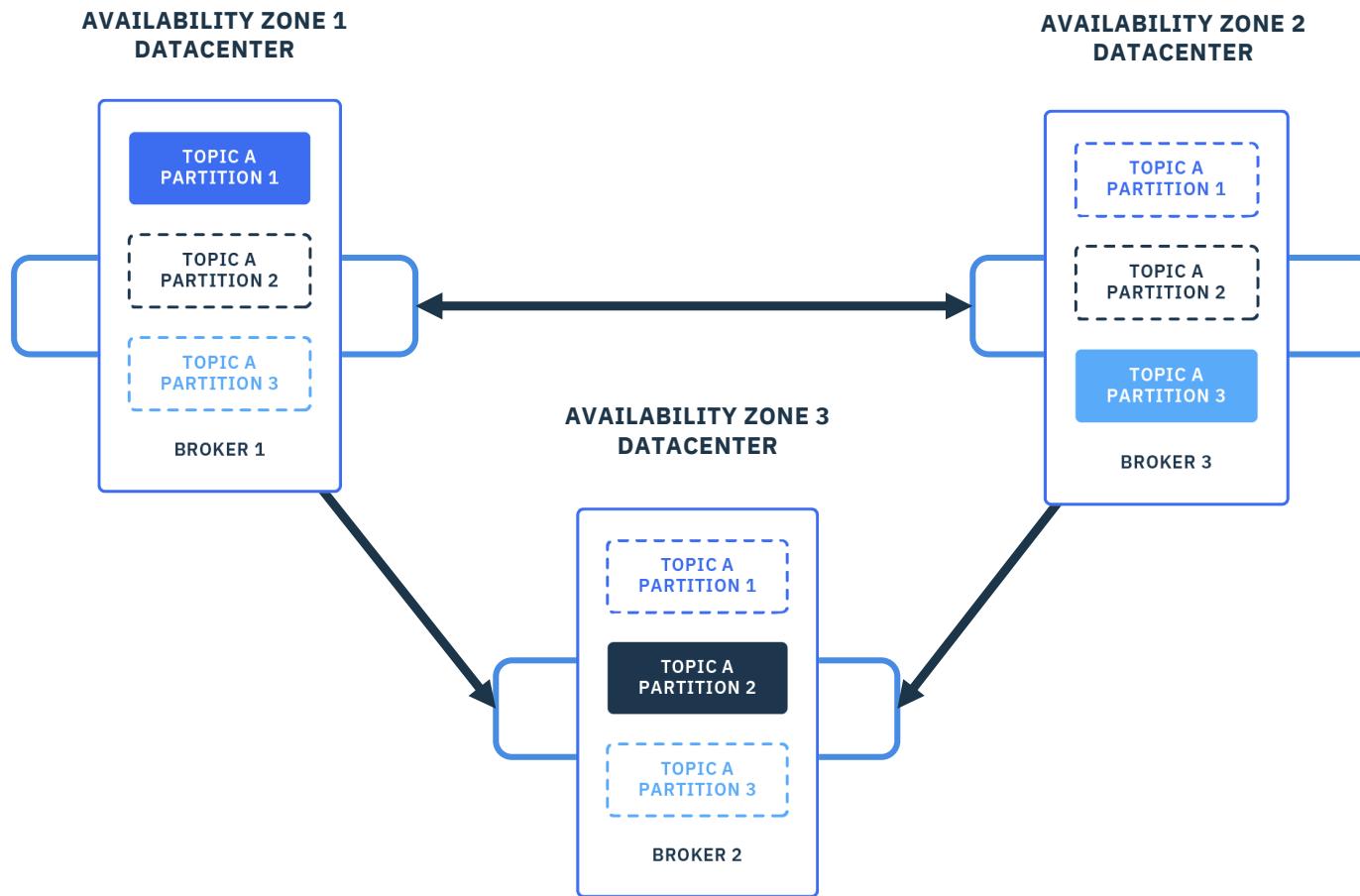
# Safe, Planned Upgrade of Apache Kafka

## **Upgrade Kafka versions safely and without hassle**

- First, upgrade the release 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



# Enhanced resilience with clusters across multiple zones



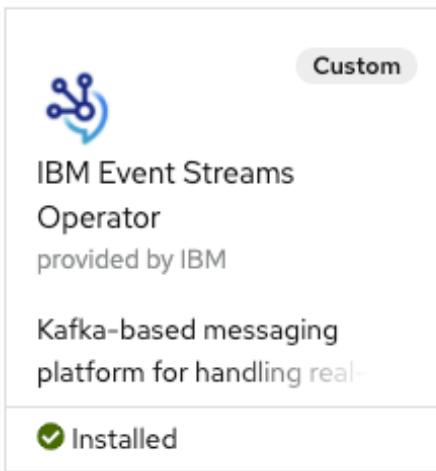
## Multi-Availability Zone

- Must have at least 3 zones
- Kafka brokers and ZooKeeper servers span across zones
- Can tolerate failure of a zone with no service degradation
- High-speed network with low latency between zones required (< 20ms)



# Making Apache Kafka Intuitive and Easy

## *Native deployment on OpenShift with Operators*



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

A screenshot of the OperatorHub interface. The search bar is set to 'event streams op'. The search results list various categories like AI/Machine Learning, Application Runtime, etc., with the IBM Event Streams Operator card highlighted. The card's details match those shown in the first screenshot.

## Simple to deploy and Manage

# Making Apache Kafka Intuitive and Easy

## *Native deployment on OpenShift with Operators*

IBM Event Streams Operator > Create EventStreams

### Create EventStreams

Create by manually entering YAML or JSON definitions, or by dragging and dropping a file into the editor.

```
1  apiVersion: eventstreams.ibm.com/v1beta1
2  kind: EventStreams
3  metadata:
4    name: quickstart
5    namespace: test-event-streams
6  spec:
7    adminApi: {}
8    adminUI: {}
9    collector: {}
10   license:
11     accept: false
12   restProducer: {}
13   schemaRegistry:
14     storage:
15       type: ephemeral
16   strimziOverrides:
17     kafka:
18       config:
19         interceptor.class.names: com.ibm.eventstreams.interceptors.metrics.ProducerMetricsInterceptor
20         offsets.topic.replication.factor: 1
21         transaction.state.log.min_isr: 1
22         transaction.state.log.replication.factor: 1
23     listeners:
24       external:
25         type: route
26       plain: {}
27       tls: {}
28     metrics: {}
```

Create

Cancel

IBM Event Streams Operator > Create EventStreams

### Create EventStreams

Create by completing the form. Default values may be provided by the Operator authors.

Name \*

myEventStreamsDeployment

Labels

app=frontend

Kafka Persistence

Kafka storage

ephemeral ▾

The type of storage used by Kafka brokers

Kafka Storage Class

Select StorageClass

Storage class to use for Kafka brokers

Zookeeper Persistence

- YAML for full control
- Form-based entry for getting started quickly



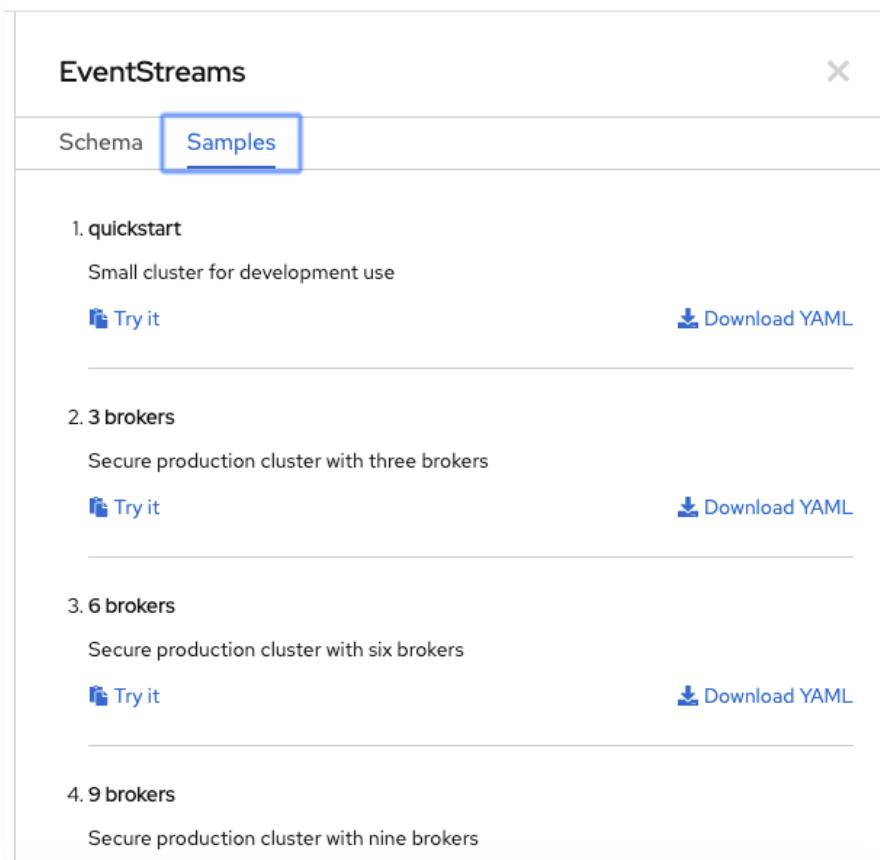
IBM Event Streams

© 2020 IBM Corporation

55

# Making Apache Kafka Intuitive and Easy

## *Native deployment on OpenShift with Operators*



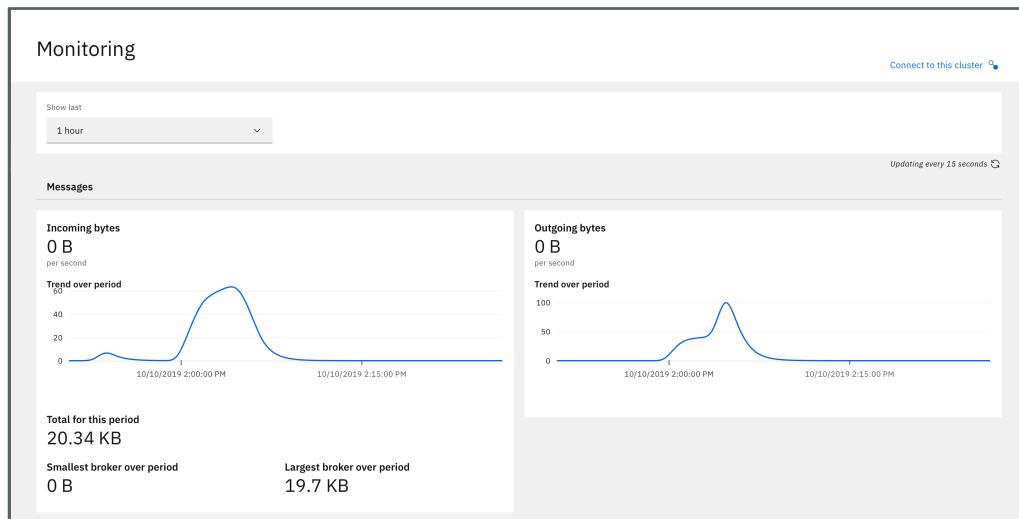
The screenshot shows the EventStreams interface with the "Samples" tab selected. It displays four operator template samples for Kafka clusters:

- 1. quickstart**: Small cluster for development use. Includes "Try it" and "Download YAML" buttons.
- 2. 3 brokers**: Secure production cluster with three brokers. Includes "Try it" and "Download YAML" buttons.
- 3. 6 brokers**: Secure production cluster with six brokers. Includes "Try it" and "Download YAML" buttons.
- 4. 9 brokers**: Secure production cluster with nine brokers. Includes "Try it" and "Download YAML" buttons.

Operator templates to get started quickly with full example configurations



# Making Apache Kafka Intuitive and Easy

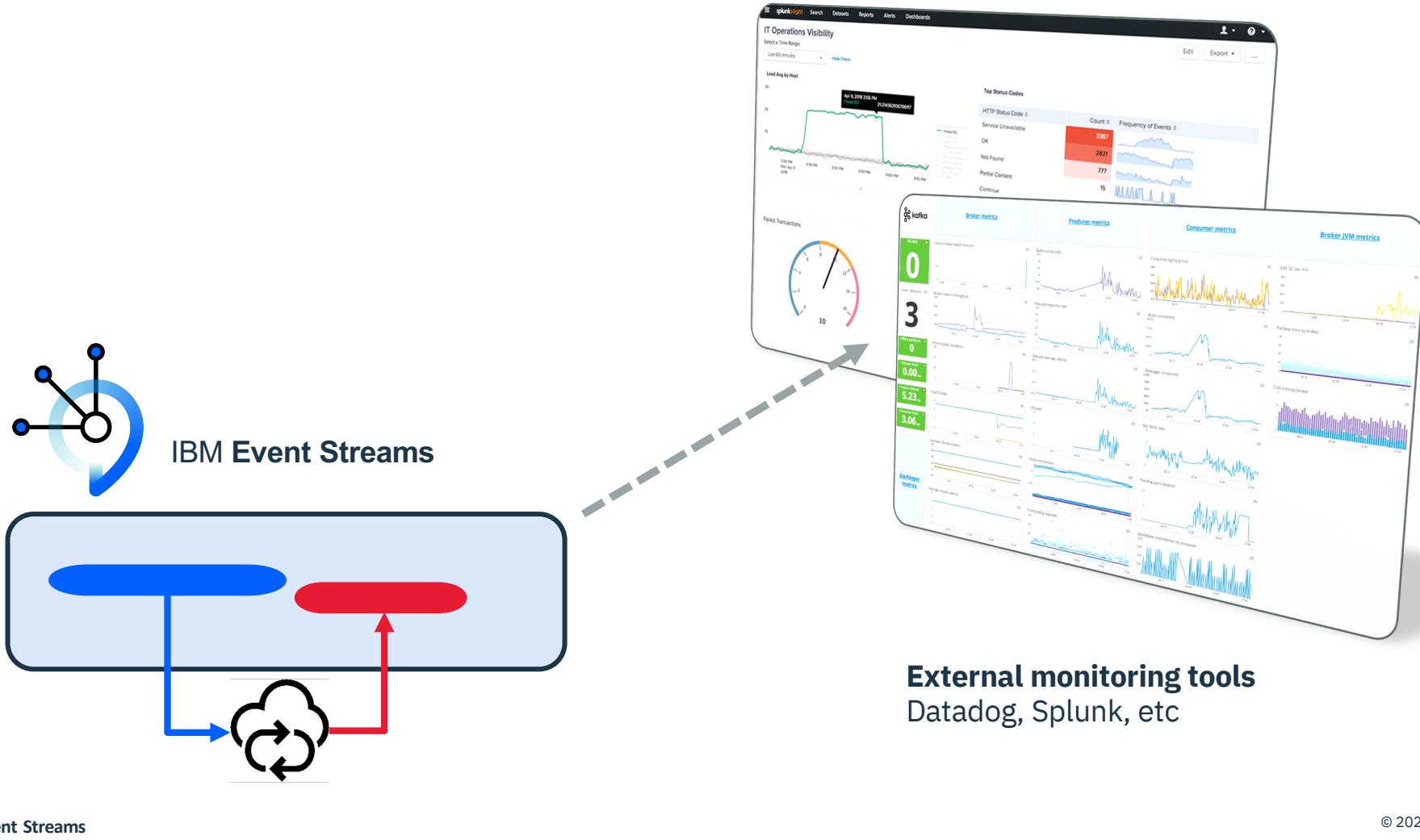


## Monitor status at a glance

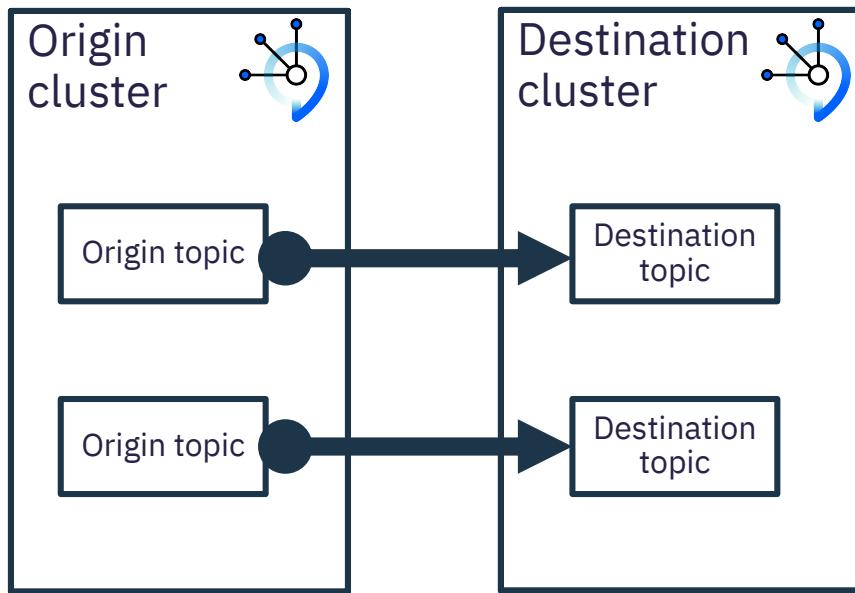
## Integrated feedback and support

This screenshot shows the 'IBM Event Streams support' interface. It features three large, light-blue 3D cube icons with small red or blue highlights. Below each icon is a call-to-action: 'Did something go wrong? Raise an issue', 'What would you like to see? Request a feature', and 'What could we do better? Give us feedback'. At the bottom, there are links for 'View Frequently Asked Questions (FAQs)', 'Existing issues', and 'Existing feature requests'.

# Integrated with Key Monitoring Tools



# Geo-Replication Makes Disaster Recovery Simple



Screenshot of the Event Streams UI under the 'Geo-replication' section. The UI includes a description of geo-replication and tabs for 'Destination locations' and 'Origin locations'. Under 'Destination locations', there are three entries:

Destination cluster ID	Topics	Workers	Action
TestDest1	10	2	>
TestDest2	6	4	>
TestDest3	2	2	>

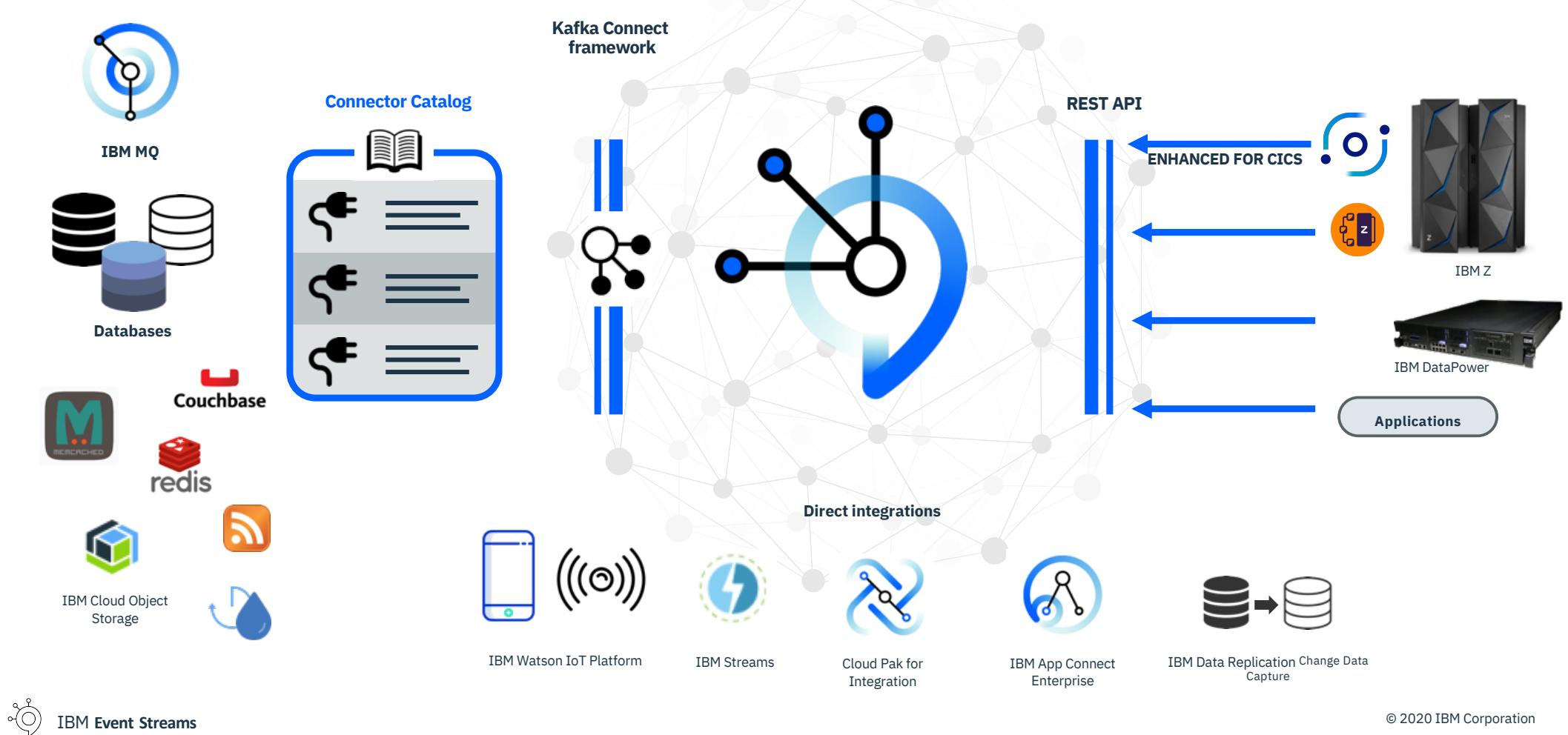
Target is take-over of workload on the destination cluster by business applications within 15 minutes

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

# Use Existing Data in New Ways that Yield Competitive Advantage

## *Unmatched Connectivity to Core Systems*



Welcome to the IBM Event Streams

# Connector catalog

Kafka Connect is a framework for connecting Kafka to external systems. It uses source connectors to move data into Kafka, and sink connectors to move data out of Kafka.

The Event Streams connector catalog contains a list of tried and tested connectors from both the community and IBM.

Find out more about Kafka Connect



All (17)	Source (8)	Sink (9)
 Source connector   Kafka Connect <b>IBM MQ</b>	 IBM supported	 Sink connector   Kafka Connect <b>IBM MQ</b>
 Sink connector   Kafka Connect <b>ArangoDB</b>		 Sink connector   Kafka Connect <b>IBM Cloud Object Storage</b>
 Source connector   Kafka Connect <b>Couchbase</b>		 Sink connector   Kafka Connect <b>Couchbase</b>
 Sink connector   Kafka Connect <b>HTTP</b>		 Sink connector   Kafka Connect <b>Memcached</b>



# 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>

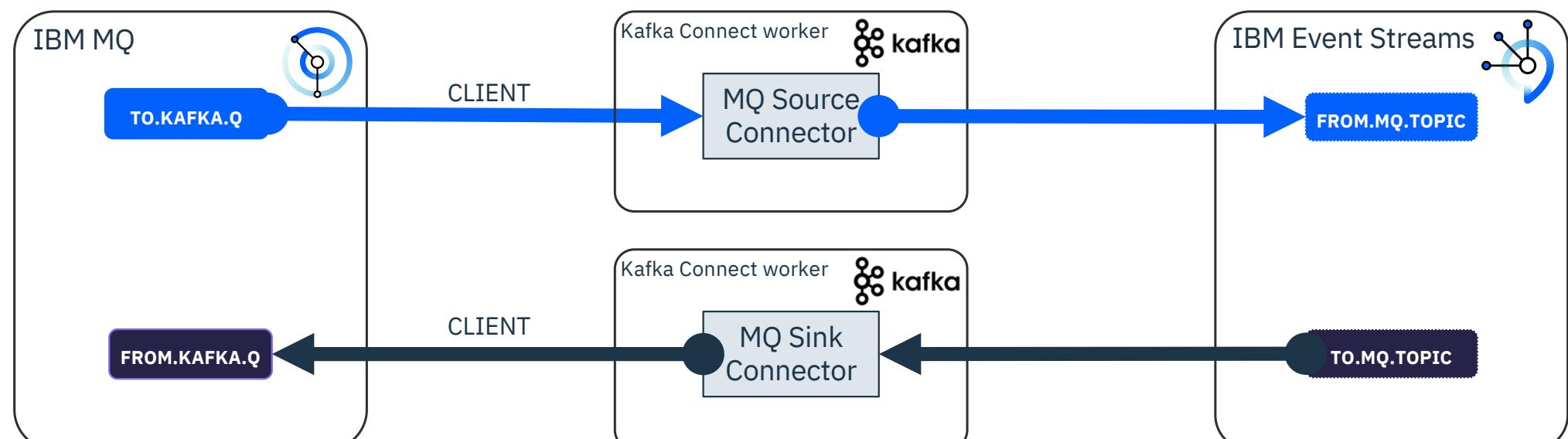
- Copies messages from MQ queues to Event Streams topics and vice versa
- Supports all current MQ versions (MQ v8 or later, all platforms)
- Extend the connector to support any business-specific message format
- Fully supported by IBM for customers with support entitlement for IBM Event Streams



# 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)



Ready for Mission-Critical Workloads



**All with IBM 24x7 worldwide support**

*IBM has years of experience running Apache Kafka across the globe*

## Find Out More

Explore IBM Event Streams at

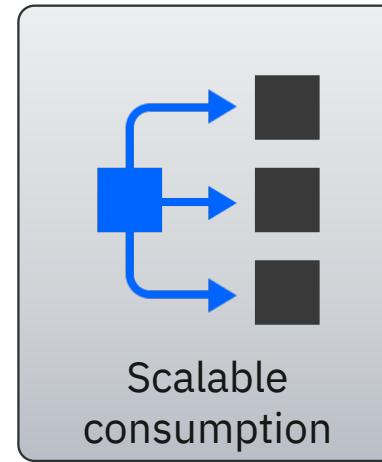
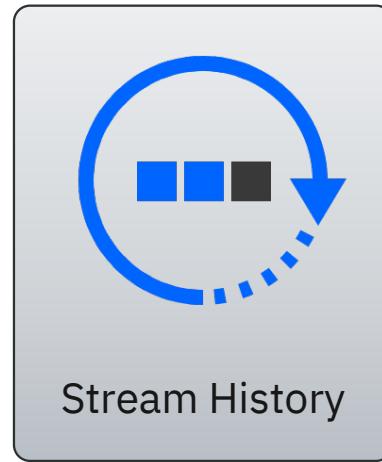
<https://ibm.github.io/event-streams/>

[https://www.tutorialspoint.com/apache\\_kafka/apache\\_kafka\\_introduction.htm](https://www.tutorialspoint.com/apache_kafka/apache_kafka_introduction.htm)

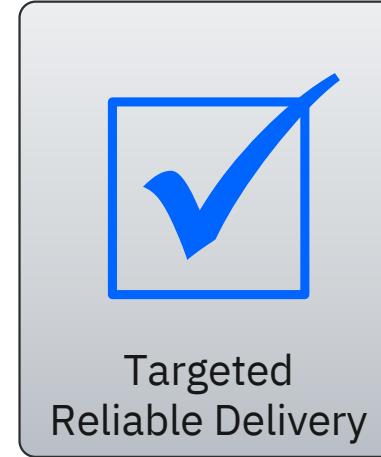
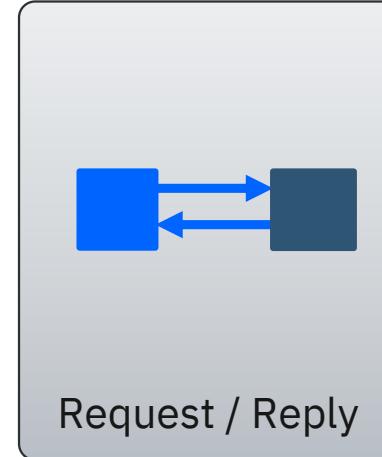
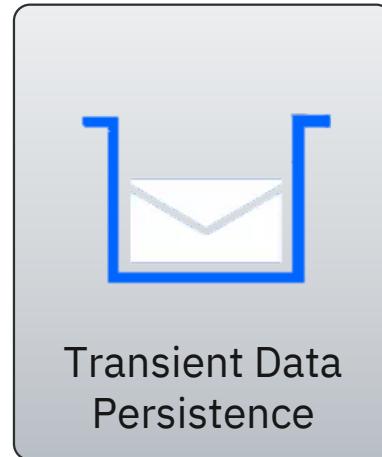
Contact us: [eventstreams@uk.ibm.com](mailto:eventstreams@uk.ibm.com)

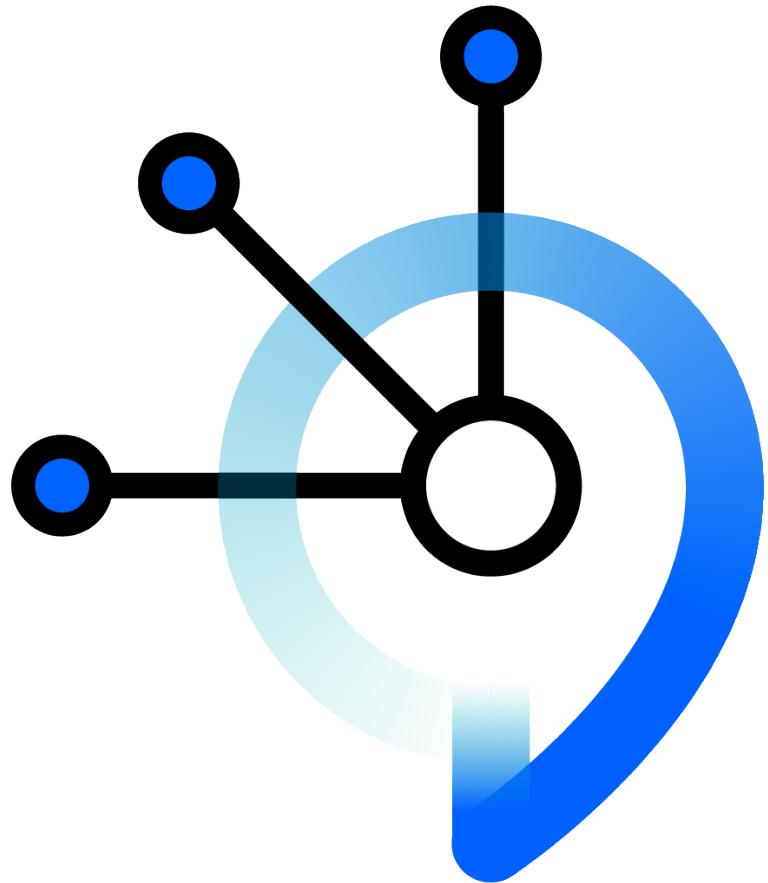
# Event Streaming & Message Queuing Need Different Capabilities

## Event Streaming



## Message Queueing





# Thank You



IBM Event Streams

© 2020 IBM Corporation