INTRODUCING AMQ STREAMS : DATA STREAMING WITH APACHE KAFKA

Ugo Landini
Principal Solution Architect

Paolo Patierno
Principal Software Engineer

What is Apache Kafka?

A publish/subscribe messaging system?

A streaming data platform?

A distributed, horizontally-scalable, fault-tolerant, commit log?

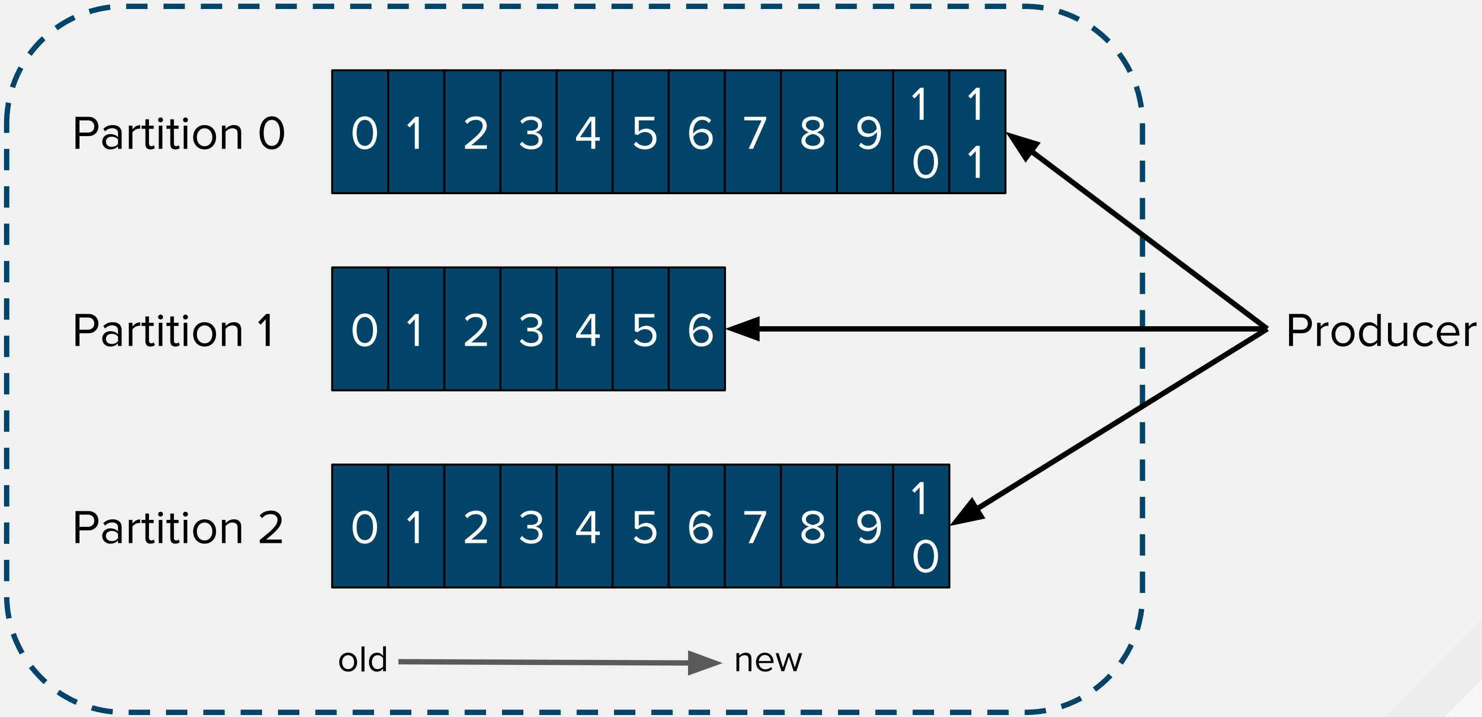
Apache Kafka

Concepts

- Messages are sent to and received from a topic
  - Topics are split into one or more partitions (aka shards)
  - All actual work is done on partition level, topic is just a virtual object
- Each message is written only into a one selected partition
  - Partitioning is usually done based on the message key
  - Message ordering within the partition is fixed
- Retention
  - Based on size / message age
  - Compacted based on message key

Topic & partitions

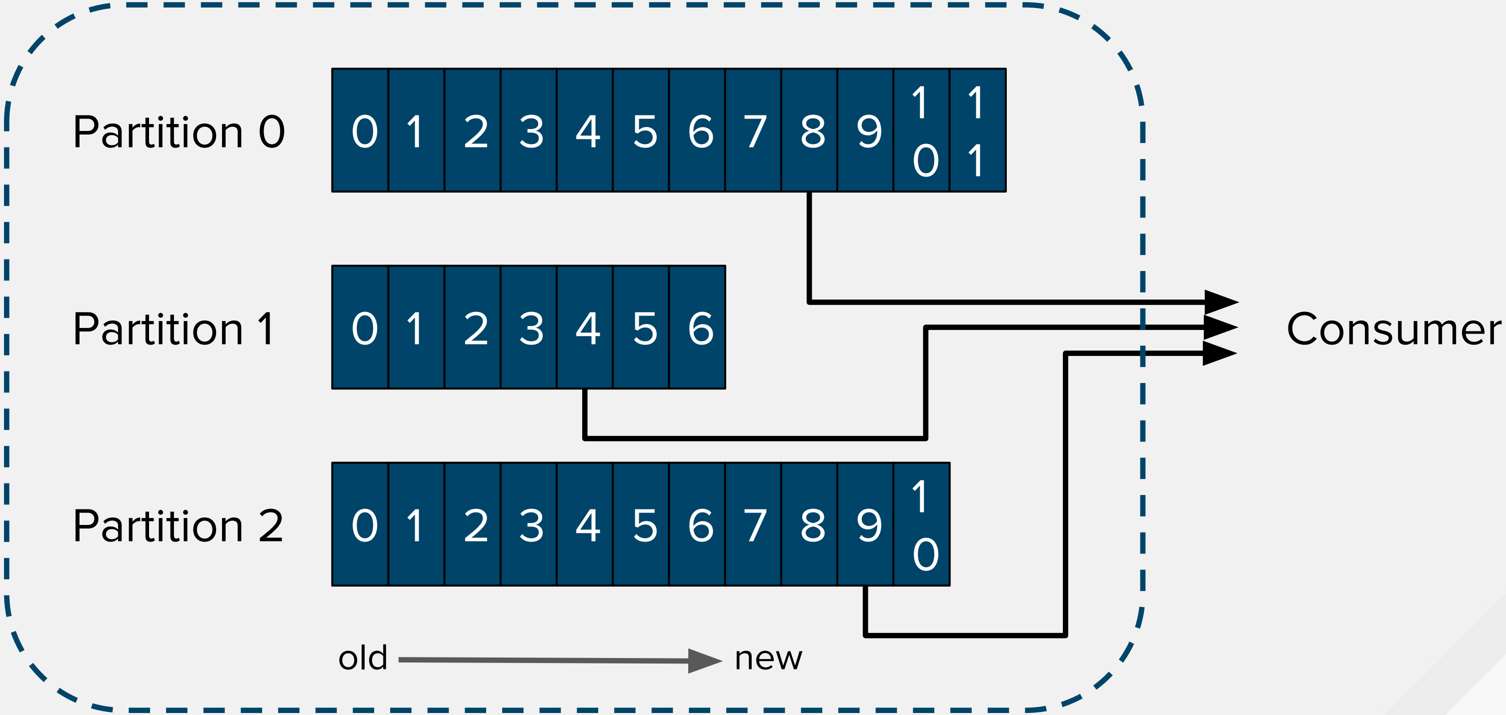
Kafka concepts



Partition 0	0	1	2	3	4	5	6	7	8	9	10	11
Partition 1	0	1	2	3	4	5	6					
Partition 2	0	1	2	3	4	5	6	7	8	9	10	

Topic & partitions

Kafka concepts



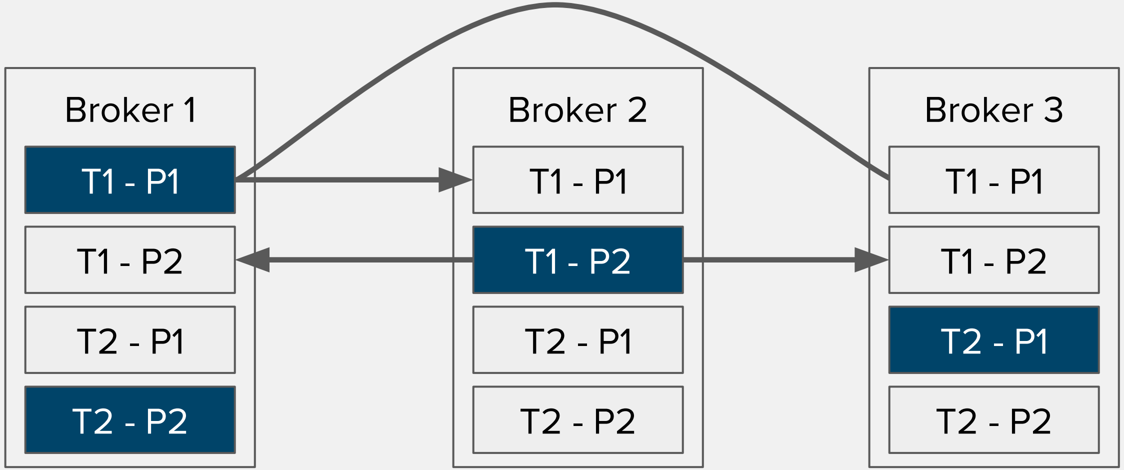
Partition 0: 0 1 2 3 4 5 6 7 8 9 10 11

Partition 1: 0 1 2 3 4 5 6

Partition 2: 0 1 2 3 4 5 6 7 8 9 10

High availability

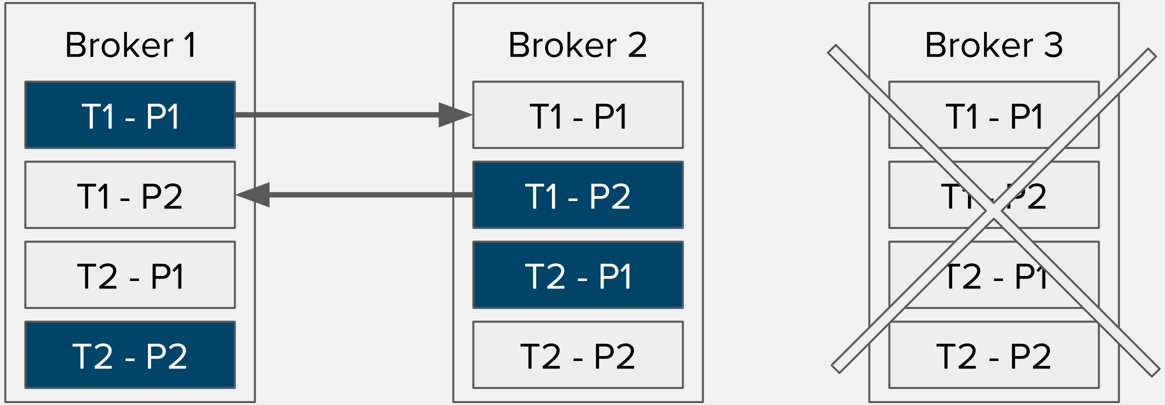
Kafka concepts



Leaders and followers spread across the cluster

High availability

Kafka concepts



If a broker with leader partition goes down, a new leader partition is elected on different node

AMQ Broker & AMQ Streams

Key differences

	AMQ Broker (ActiveMQ Artemis)	AMQ Streams (Kafka)
Model	“Smart broker, dumb clients”	“Dumb broker, smart clients”
Durability	Volatile or durable storage	Durable storage
Storage duration	Temporary storage of messages	Potential long-term storage of messages
Message retention	Retained until consumed	Retained until expired or compacted
Consumer state	Broker managed	Client managed (can be stored in broker)
Selectors	Yes, per consumer	No
Stream replay	No	Yes
High-availability	Replication	Replication
Protocols	AMQP, MQTT, OpenWire, Core, STOMP	Kafka protocol
Delivery guarantees	Best-effort or guaranteed	Best-effort or guaranteed

AMQ Streams

Why should you use AMQ Streams ?

- Scalability and performance
  - Designed for horizontal scalability
- Message ordering guarantee
  - At partition level
- Message rewind/replay
  - “Long term” storage
  - Allows to reconstruct application state by replaying the messages
  - Combined with compacted topics allows to use Kafka as key-value store

AMQ Streams

What’s the catch ?

- Kafka protocol is non-trivial to proxy
  - Clients need access to all brokers in the cluster
  - Producers/consumers might need to maintain large number of TCP connections
  - Proxying via HTTP REST or AMQP could be a solution
- Dumb broker, smart clients
  - Carefully decide the “right” number of partitions for each topic
  - Adding partitions can change destination partition for “keyed” messages
  - Removing partitions is not possible

AMQ Streams on OpenShift

- Based on OSS project called Strimzi
- Provides:
  - Docker images for running Apache Kafka and Zookeeper
  - Tooling for managing and configuring Apache Kafka clusters and topics
- Follows the Kubernetes “operator” model
- OpenShift 3.9 and higher

AMQ Streams on OpenShift

What is Strimzi ?

- Open source project focused on running Apache Kafka on Kubernetes and OpenShift
- Licensed under Apache License 2.0
- Web site: http://strimzi.io/
- GitHub: https://github.com/strimzi
- Slack: strimzi.slack.com
- Mailing list: strimzi@redhat.com
- Twitter: @strimziio

AMQ Streams on OpenShift

The challenges

- Apache Kafka is *stateful* which means we require …
  - … a stable broker identity
  - … a way for the brokers to discover each other on the network
  - … durable broker state (i.e., the messages)
  - … the ability to recover broker state after a failure
- All the above are true for Apache Zookeeper as well
- StatefulSets, PersistentVolumeClaims, Services can help but …

It’s not easy!

AMQ Streams on OpenShift

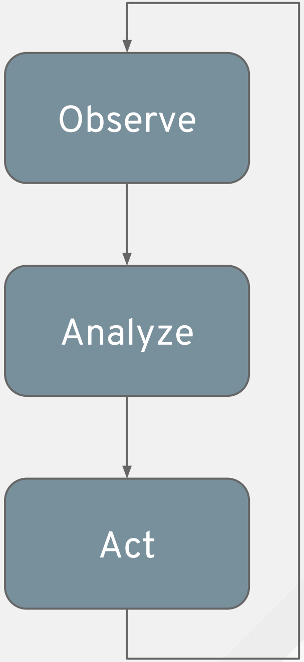
Goals

- Simplifying the Apache Kafka deployment on OpenShift
- Using the OpenShift native mechanisms for…
  - Provisioning the cluster
  - Managing the topics
- … thereby removing the need to use Kafka command-line tools
- Providing a better integration with applications running on OpenShift
  - microservices, data streaming, event-sourcing, etc.

AMQ Streams on OpenShift

The “Operator” model

- An application used to create, configure and manage other complex applications
  - Contains specific domain / application knowledge
- Operator takes as input Config Maps or Custom Resource Definitions
  - User describes the desired state
  - Operator applies this state to the application
- It watches the *desired* state and the *actual* state ...
  - ... taking appropriate actions



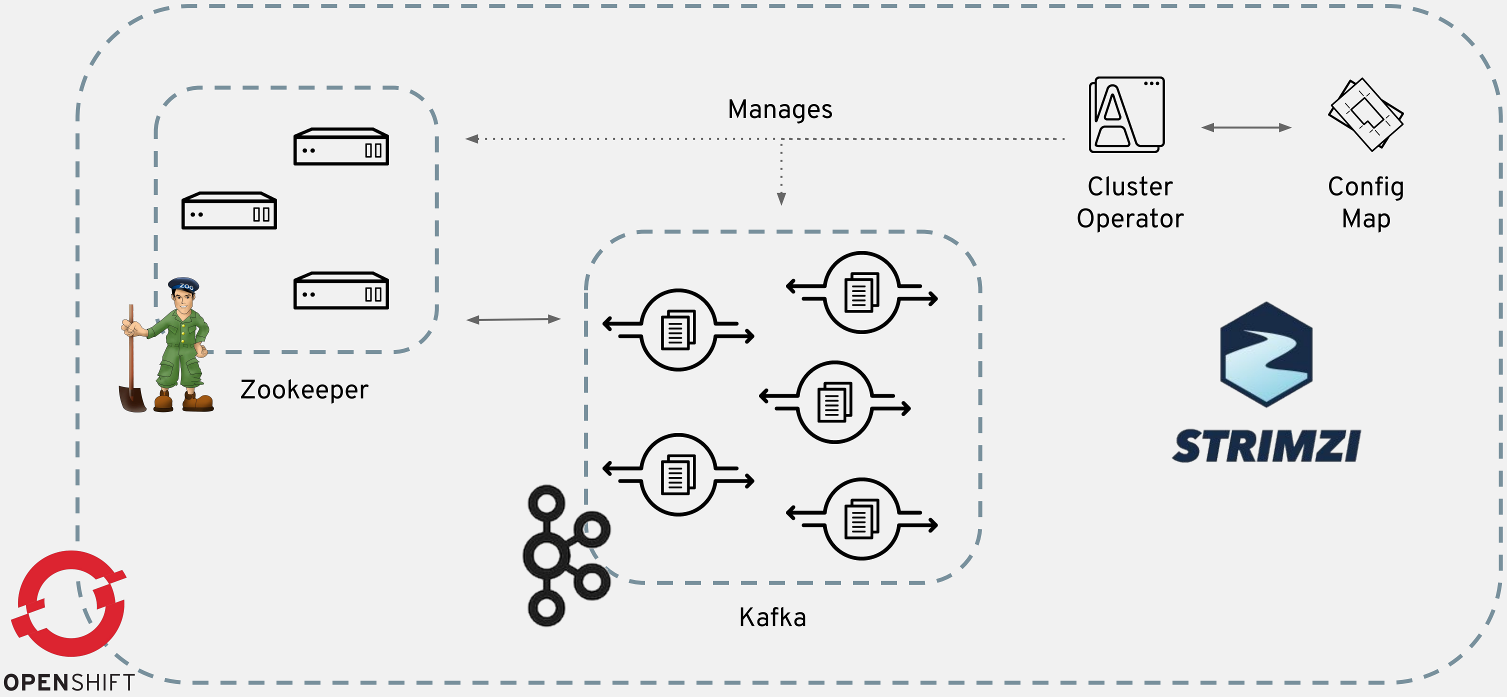
AMQ Streams on OpenShift

Config Map versus Custom Resource Definitions

- Operators are currently using Config Maps for configuration
  - Main advantage of Config Maps is no need for special permissions to install Strimzi/AMQ Streams on OpenShift
- CRDs have some advantages as well
  - Flexible data structure
  - Possibility to set permissions for the CRD resources
- Adding support for CRDs is on backlog for the future

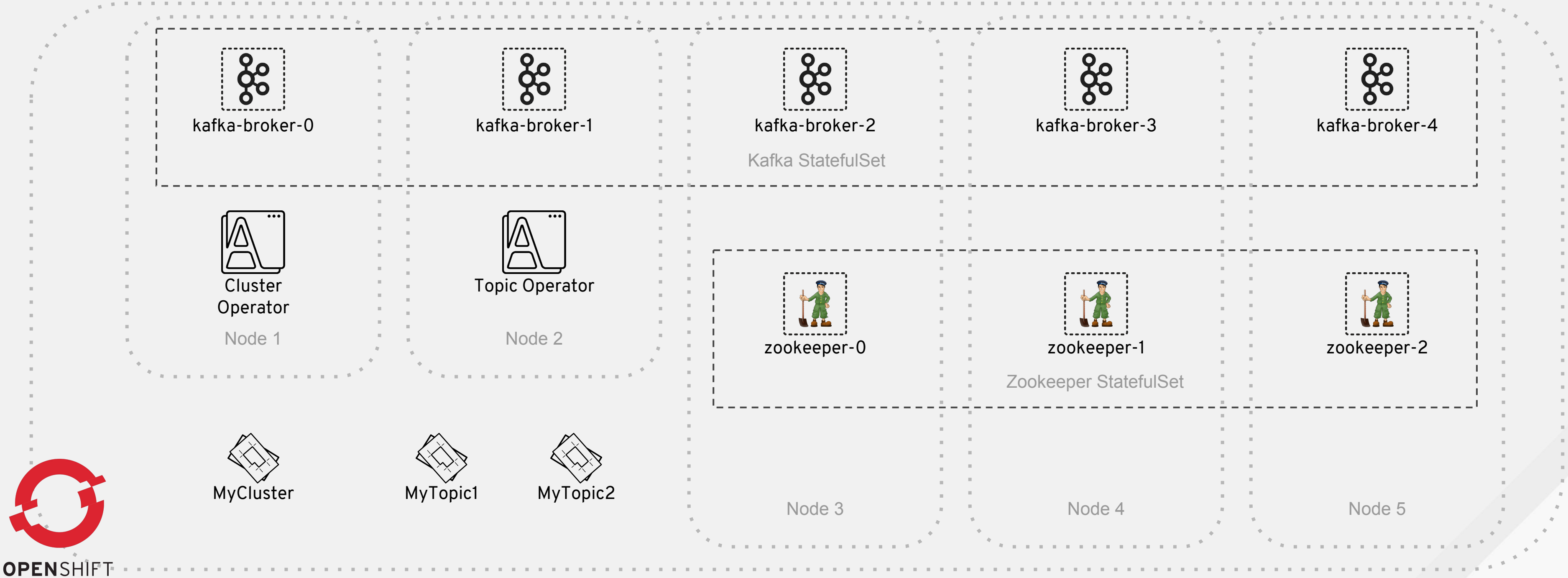
Cluster Operator

Creating and managing Apache Kafka clusters



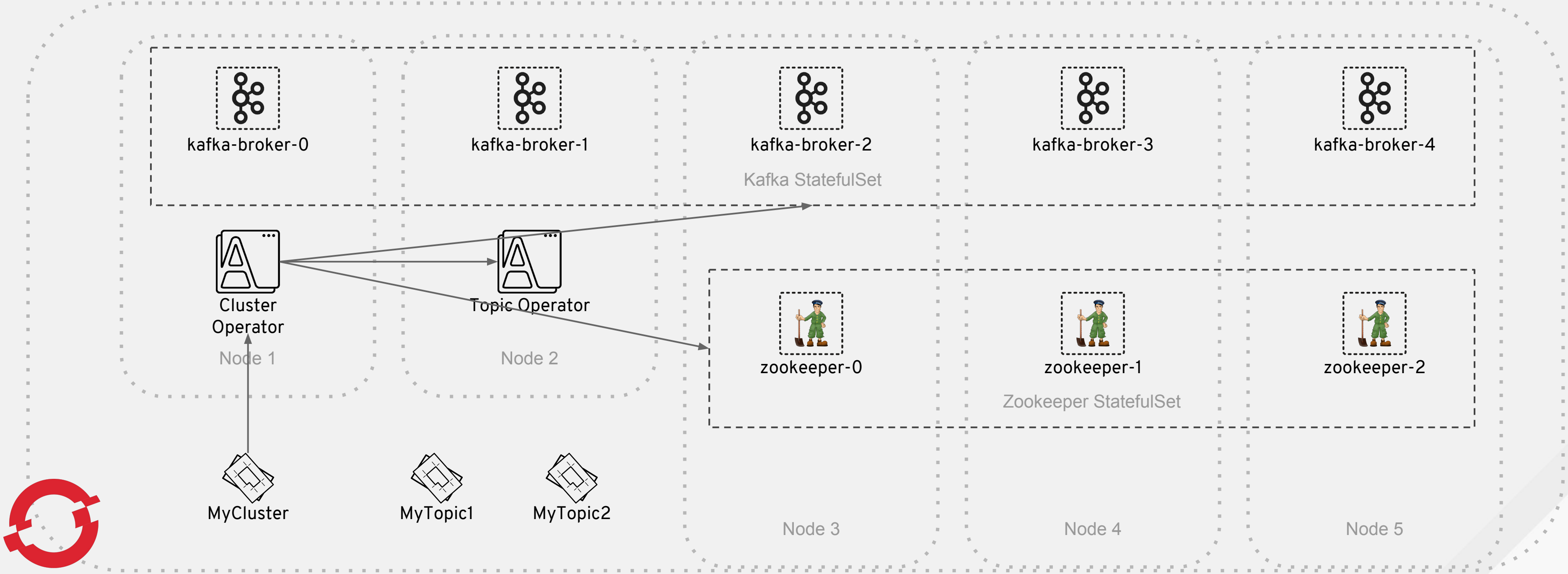
Cluster Architecture

Overview



Cluster Architecture

Deploying a cluster



Cluster Operator

Creating cluster

- Able to deploy two types of clusters
  - Kafka (alongside a Zookeeper ensemble)
  - Kafka Connect (even with S2I support for custom connector plugins)
- The ConfigMap allows to specify
  - Number of nodes
  - Brokers configuration
  - Healthchecks
  - Metrics exported for Prometheus
- Ephemeral or persistent storage

Cluster Operator

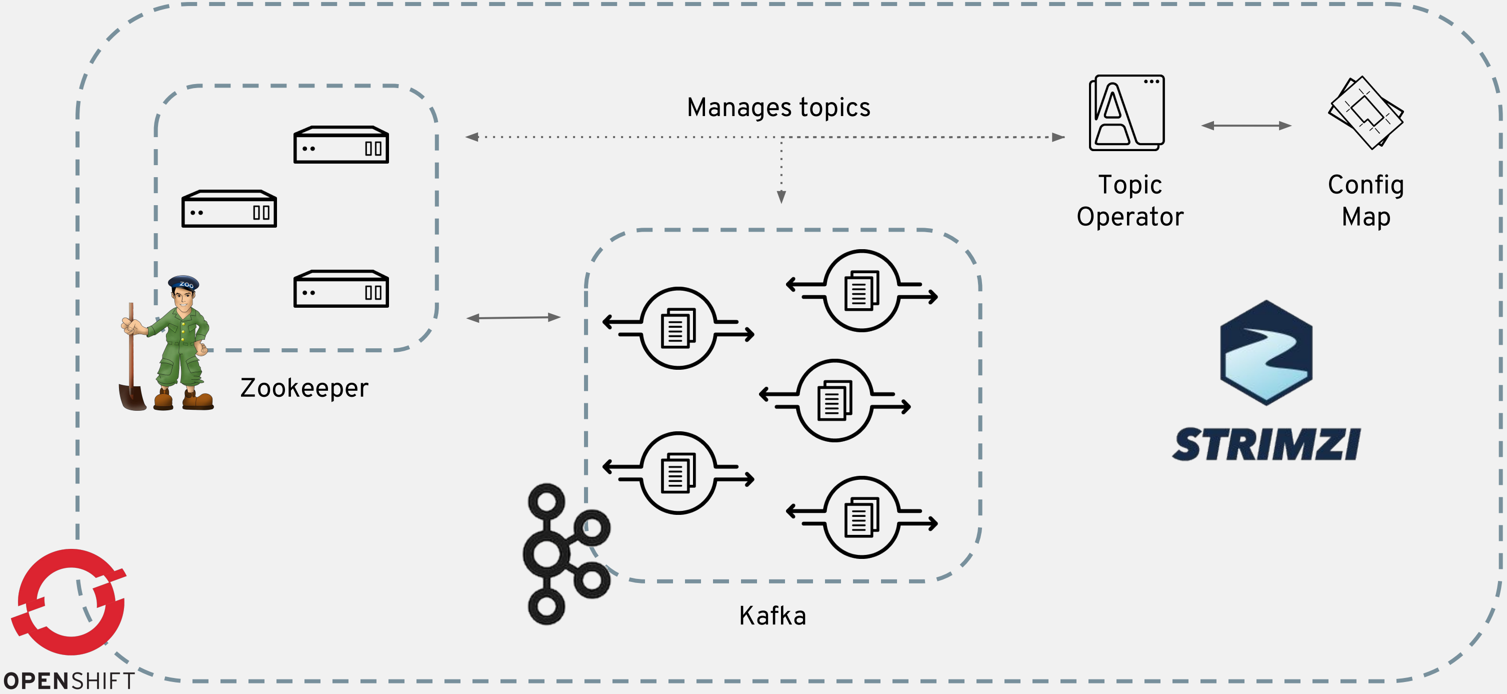
Managing cluster

- Modifying the ConfigMap for updating the cluster
  - Scale up/down
  - Configuration changes (rolling updates)
- Deleting the ConfigMap for de-provisioning the cluster
  - Persisted data will be deleted according to the user configuration

DEMO : KAFKA CLUSTER DEPLOYMENT

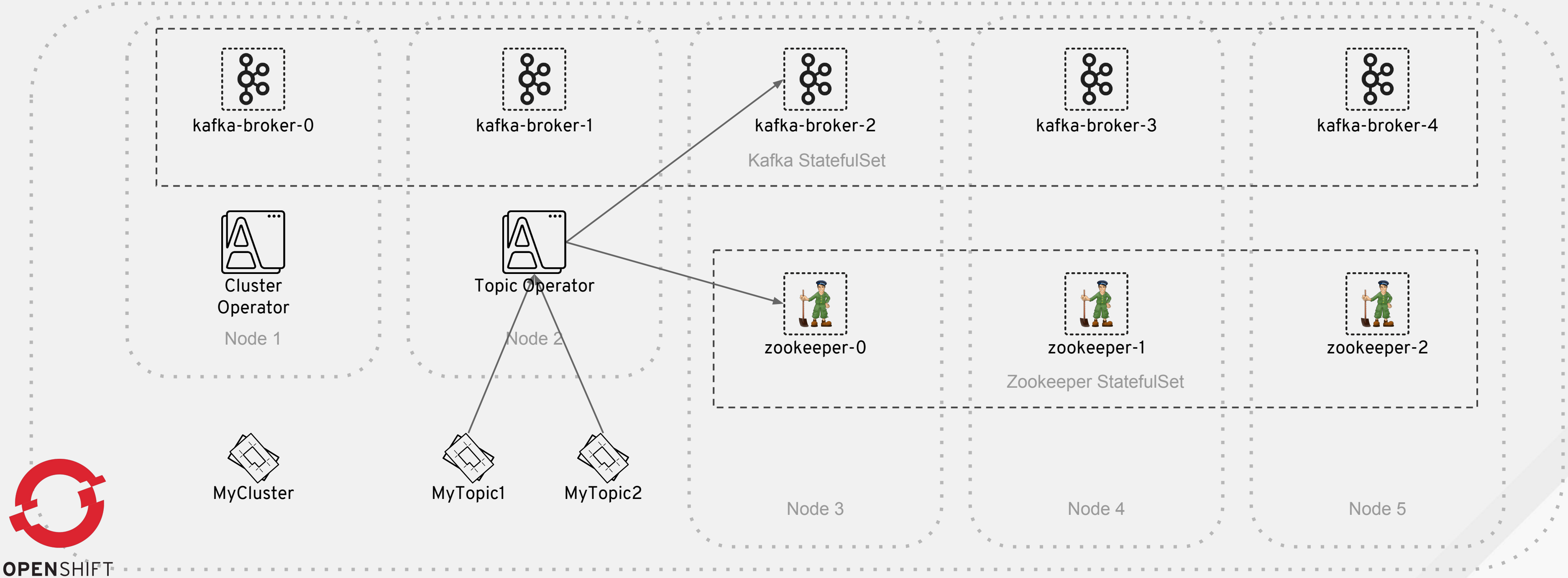
Topic Operator

Creating and managing Kafka topics



Cluster Architecture

Managing topics



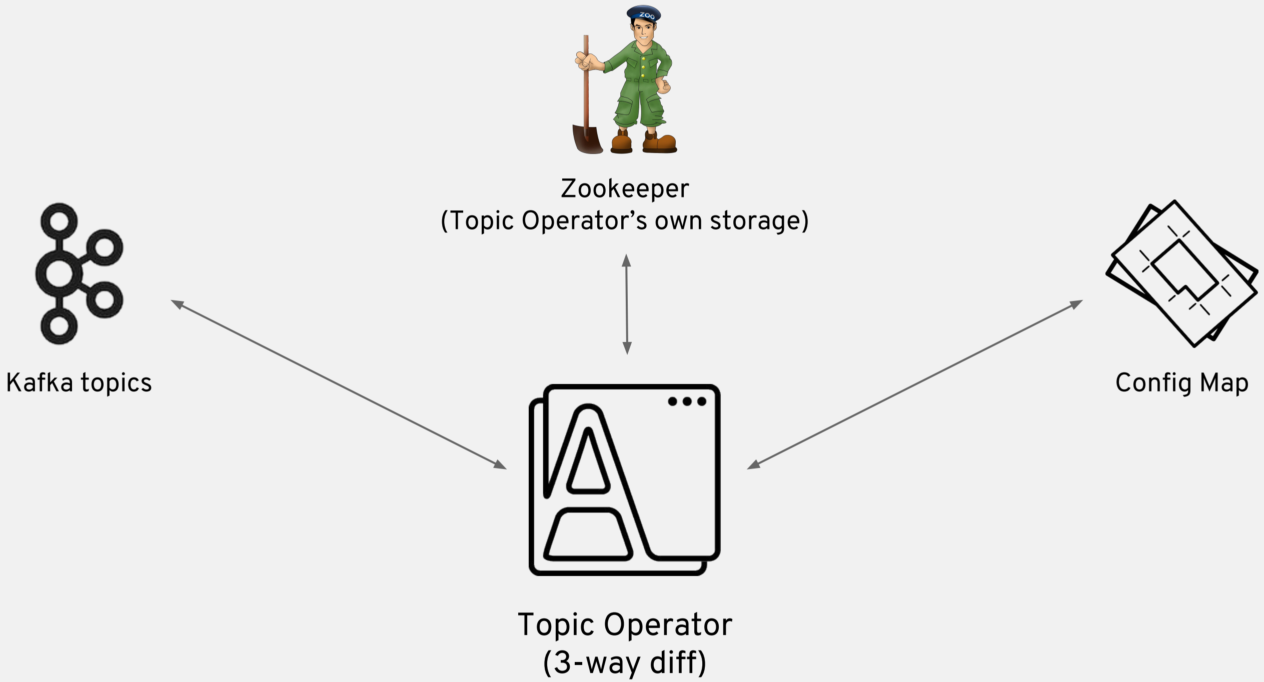
Topic Operator

Creating and managing Kafka topics

- Topics can be created by…
  - Writing a ConfigMap
  - Interacting directly with Kafka cluster
  - Automatically by others (Kafka Connect, Kafka Streams)
- Consistency is handled by using 3-way diff
  - Our own Zookeeper store
  - Apache Kafka/Zookeeper
  - ConfigMaps

Topic Operator

Creating and managing Kafka topics



Zookeeper
(Topic Operator’s own storage)

Kafka topics

Config Map

Topic Operator
(3-way diff)

DEMO : TOPICS MANAGEMENT

AMQ Streams on OpenShift

Planned for 1.0

- Detailed Kafka configuration (buffers, topic defaults, etc.)
- TLS encryption and authentication
- Authentication options
  - TLS Client Authentication
  - SASL-SCRAM mechanism with credentials stored in Zookeeper
  - SASL-PLAIN mechanisms with credentials stored in OpenShift secret
- Authorization using ACL rules stored in Zookeeper
- Resources configuration (memory and CPU limits, ...)
- Scaling (with manual partition reassignment)
- Managing topics

AMQ Streams on OpenShift

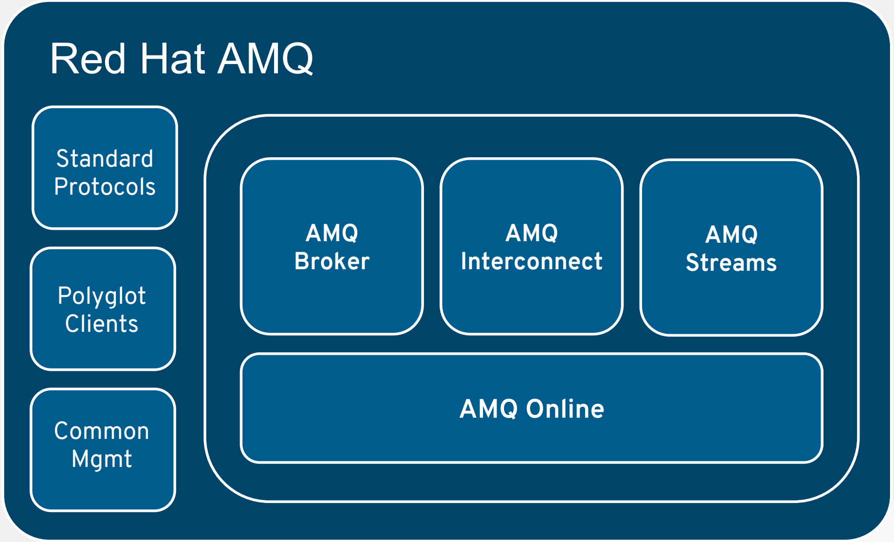
Planned after 1.0

- Kafka updates
- Automated partition balancing and automated scaling
- Additional authentication options
  - Using Red Hat SSO, LDAP, Kubernetes tokens
- Exposing Kafka cluster outside of OpenShift
- Service broker integration
- Integrated AMQP, MQTT and HTTP bridge
- Integrated Schema registry
- MirrorMaker Operator

Summary

Click to add subtitle

- AMQ Streams is distribution of Apache Kafka included as part of the AMQ product
- Simplifies the deployment, management and monitoring of Kafka on OpenShift using the Operator approach
- Fully open source based on Strimzi
- Available now as a Developer Preview Signup: http://amq.io/amqstreams-signup
- Beta tentatively planned for Summer 2018 with GA in late Fall 2018



Resources

- Strimzi : http://strimzi.io/
- Apache Kafka : https://kafka.apache.org/
- AMQ Streams Dev Preview : http://amq.io/amqstreams-signup
- Demo : https://github.com/ppatierno/ocp-roadshow-2018

redhat.

THANK YOU

plus.google.com/+RedHat

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHatNews