STRIMZI

Where Apache Kafka meets OpenShift

kubectl apply -f me.yaml

```yaml
apiVersion: developer/v43
kind: PrincipalMiddlewareArchitect
metadata:
  name: Jose Roman Martin Gil
  namespace: Integration, Messaging, Development
  annotations:
    company: Red Hat
    github: https://github.com/rmarting
    linkedid: https://www.linkedid.com/in/jromanmartin
  labels:
    family: father, husband, friend
    sports: runner, biker
spec:
  replicas: 1
```

@jromanmartin

Spoiler!!

It is a true love story!

kafka ❤️ OPENSHIFT

What is Apache Kafka?

Apache Kafka is a distributed system designed for streams. It is built to be an horizontally-scalable, fault-tolerant, commit log, and allows distributed data streams and stream processing applications.

Main Use Cases

Messaging

Replacement of traditional message broker, has better throughput, built-in partitioning, replication, and fault-tolerance. Provides strong durability.

Website Activity Tracker

Rebuild user activity tracking pipeline as a set of real-time publish-subscribe feeds. Activity is published to central topics with one topic per activity type.

Metrics

Aggregation of statistics from distributed applications to produce centralized feeds of operational data.

Main Use Cases

Stream Processing

Enables continuous, real-time applications built to react to, process, or transform streams.

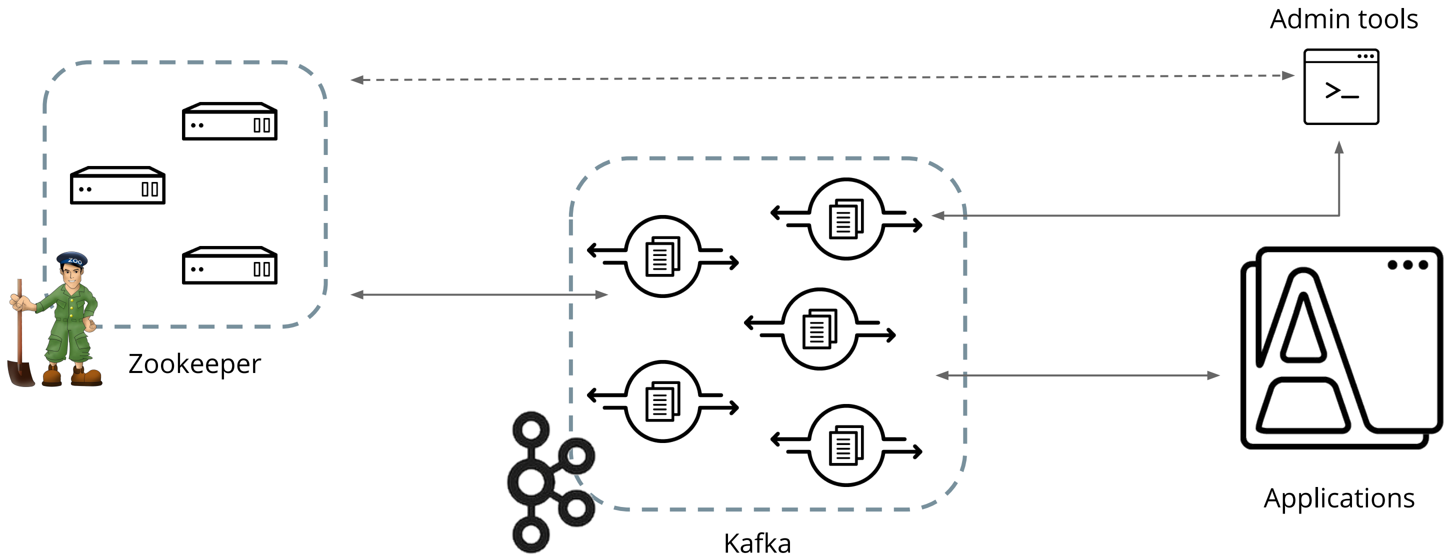
Data Integration

Captures streams of events or data changes and feeds these to other data systems.

Log Aggregation

Abstracts details of files an gives event data as stream of messages. Offers good performance, stronger durability guarantees due to replication.

Apache Kafka Core Components



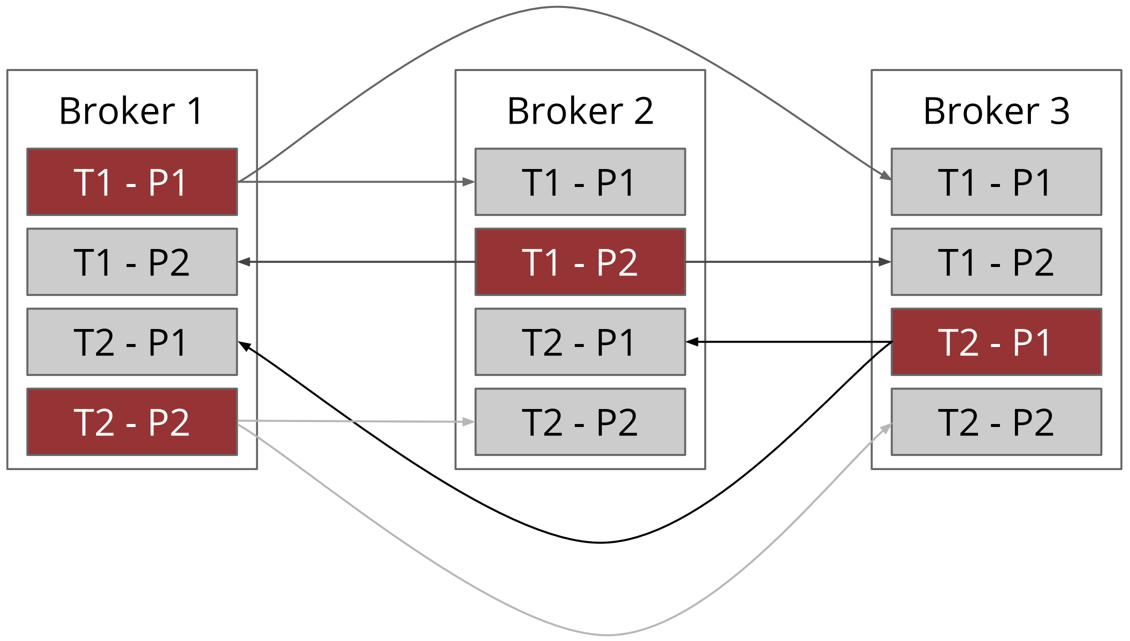
STRIMZI

Main Concepts

- Messages/records are sent to/received from topic
  - Topics are split into one or more partitions
  - All actual work is done on partition level, topic is just a virtual object
- Each message is writing only into one selected partition
  - Partitioning is usually done based on the message key
  - Messaging ordering within the partition is fixed
- Retention policies
  - Based on size/message age
  - Compacted based on message key
- Replication
  - Each partition can exist in one or more backup copies to achieve HA

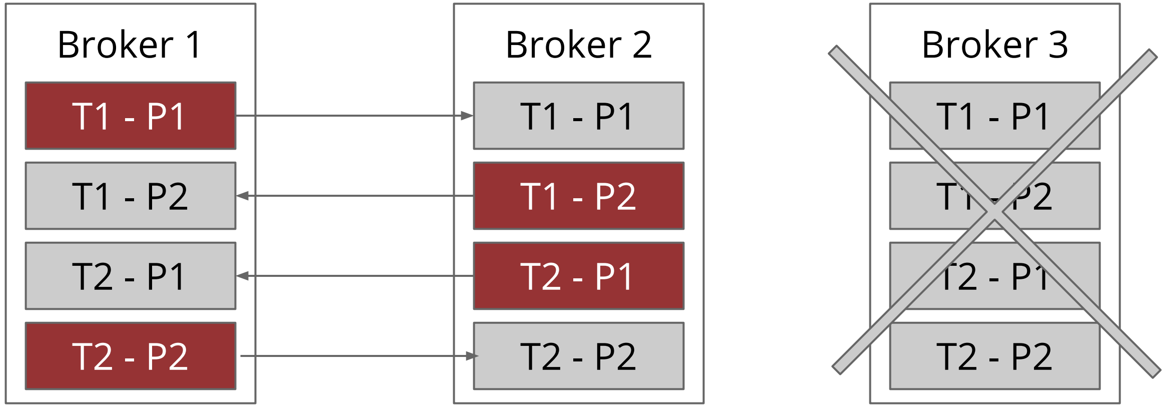
High Availability

Replication from leaders to followers



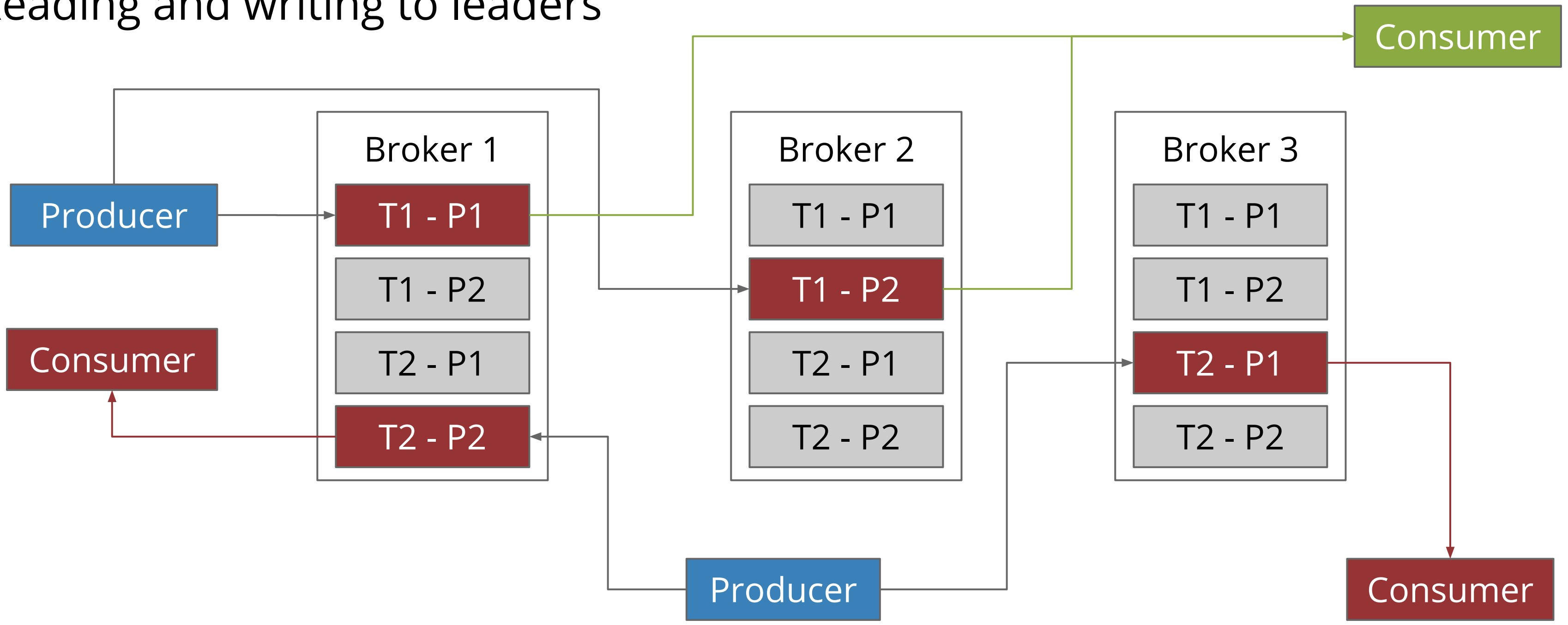
High Availability

New leaders on different brokers



High Availability

Reading and writing to leaders



Applications

- They are really “smart” (unlike “traditional” messaging)
- Configured with a “bootstrap servers” list for fetching first metadata
  - Where are interested topics? Connect to broker which holds partition leaders
  - Producer specifies destination partition
  - Consumer handles messages offsets to read
  - If error happens, refresh metadata (something is changed in the cluster)
- Batching on producing/consuming

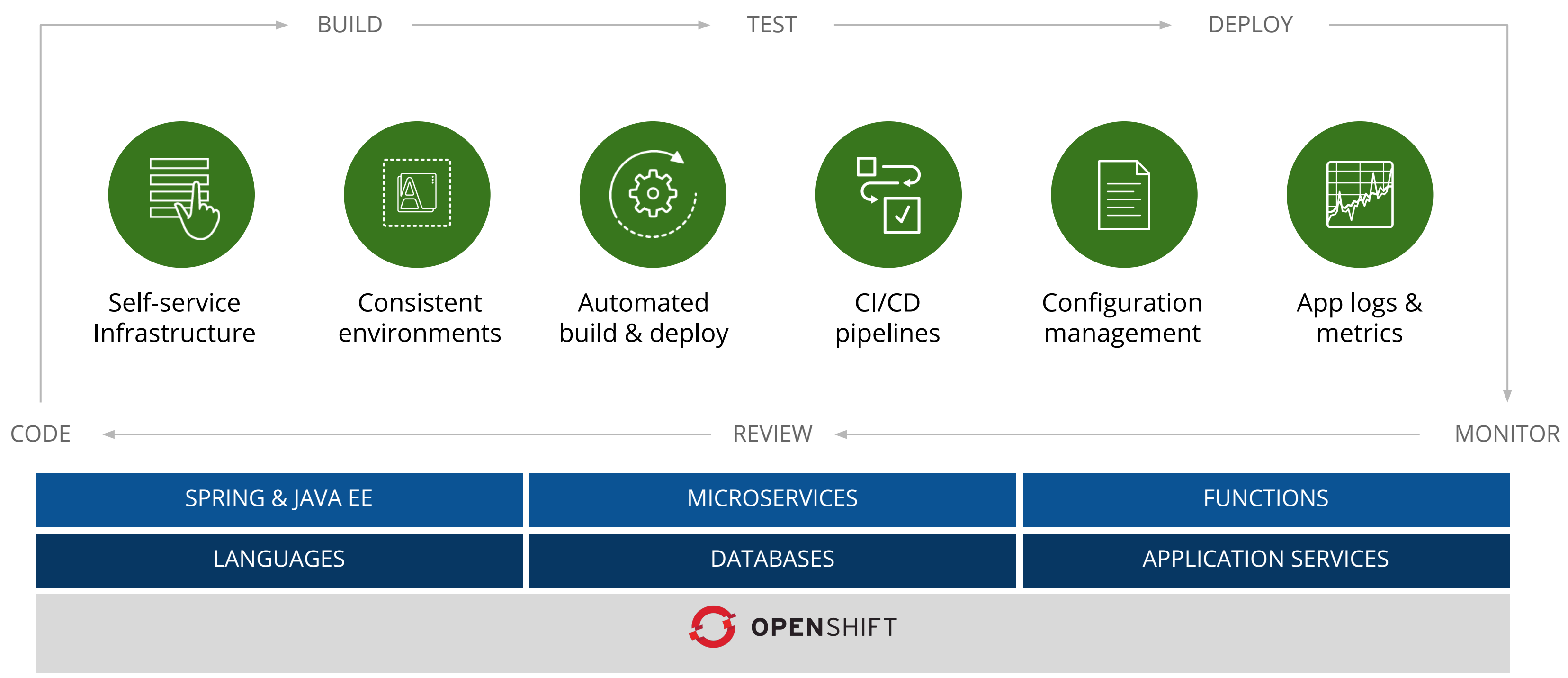
What is OpenShift?

A distribution of Kubernetes optimized for continuous application development and multi-tenant deployment.

What is OpenShift?

- Based in Kubernetes
  - Container Orchestration Manager
  - Started by Google (experience from project “Borg”)
- Extends Kubernetes to make developer lives easier.
- Adds developer and operations-centric tools to enable rapid application development, easy deployment and scaling, and long-term lifecycle maintenance for DevOps teams.
- Might be a little opinionated ...

How OpenShift enables Developer Productivity



But ...

- 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
  - The ability to recover broker state after a failure
- All the above are true for Zookeeper as well and ...
  - Each node has the configuration of the others
  - To have notes able to communicate each others
- Accessing Kafka is not so simple
- OpenShift provides several services ...

Who is going to help me?

Strimzi is our hero!

Strimzi is a set of enabling services that allow Apache Kafka to work in OpenShift as a first class citizen, be installed easily and configured and managed simply.

What is Strimzi?

- Simplifies Apache Kafka deployment on OpenShift
- Uses the OpenShift native mechanisms for:
  - Provisioning Kafka clusters
  - Managing topics and users
- Provides:
  - Linux Container images for running Apache Kafka and Zookeeper
  - Tooling for managing and configuring Apache Kafka clusters, topics and users
- Follows the Kubernetes Operator Model
- Available from ***OperatorHub.io***

OperatorsHub.io

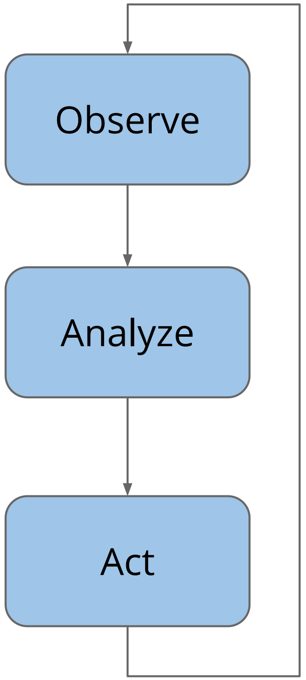
- Home for the Kubernetes community to share Operators
- Many different Operators available ...
  - Big Data, Cloud Provider, Database, Logging & Tracing, Monitoring, ...
- ... And growing up !!!

OperatorHub.io
Strimzi Apache Kafka Operator
Run an Apache Kafka cluster, including Kafka Connect, MirrorMaker and more.

Kubernetes Operators

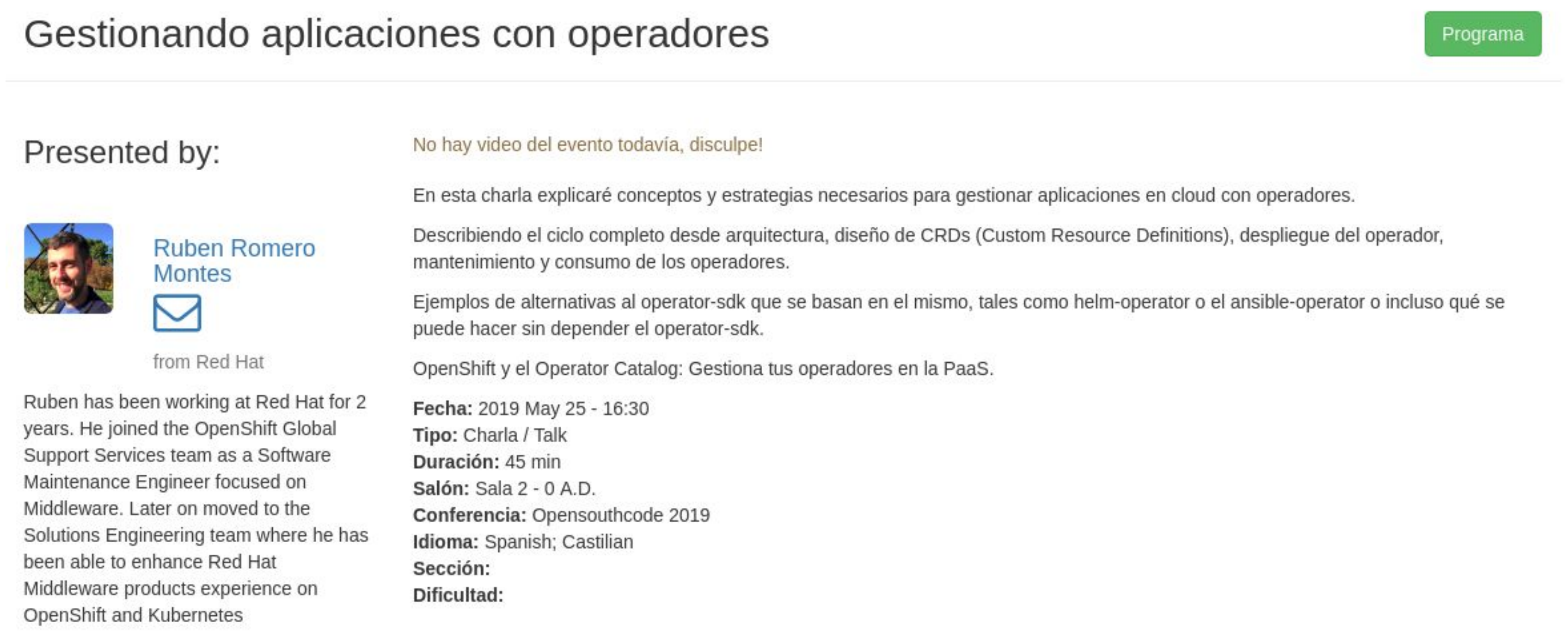
- Application-specific controller is used to create, configure and manage other complex application:
    - The controller contains specific domain/application knowledge
    - Usually used for stateful applications (databases, ...) which are non-trivial to operate on Kubernetes/OpenShift
- Controller operates based on input from Config Maps or Custom Resource Definitions
    - User describes the desired state
    - Controller applies this state to the application

Operator Pattern



- Monitor the current state of the application
- Compare the actual state to the desire state
- Resolve any differences between actual and desired state

Operators in OpenSouthCode



Gestionando aplicaciones con operadores

Programa

Presented by:

Ruben Romero Montes

from Red Hat

Ruben has been working at Red Hat for 2 years. He joined the OpenShift Global Support Services team as a Software Maintenance Engineer focused on Middleware. Later on moved to the Solutions Engineering team where he has been able to enhance Red Hat Middleware products experience on OpenShift and Kubernetes

No hay video del evento todavía, disculpe!

En esta charla explicaré conceptos y estrategias necesarios para gestionar aplicaciones en cloud con operadores.

Describiendo el ciclo completo desde arquitectura, diseño de CRDs (Custom Resource Definitions), despliegue del operador, mantenimiento y consumo de los operadores.

Ejemplos de alternativas al operator-sdk que se basan en el mismo, tales como helm-operator o el ansible-operator o incluso qué se puede hacer sin depender el operator-sdk.

OpenShift y el Operator Catalog: Gestiona tus operadores en la PaaS.

Fecha: 2019 May 25 - 16:30
Tipo: Charla / Talk
Duración: 45 min
Salón: Sala 2 - 0 A.D.
Conferencia: Opensouthcode 2019
Idioma: Spanish; Castilian
Sección:
Dificultad:

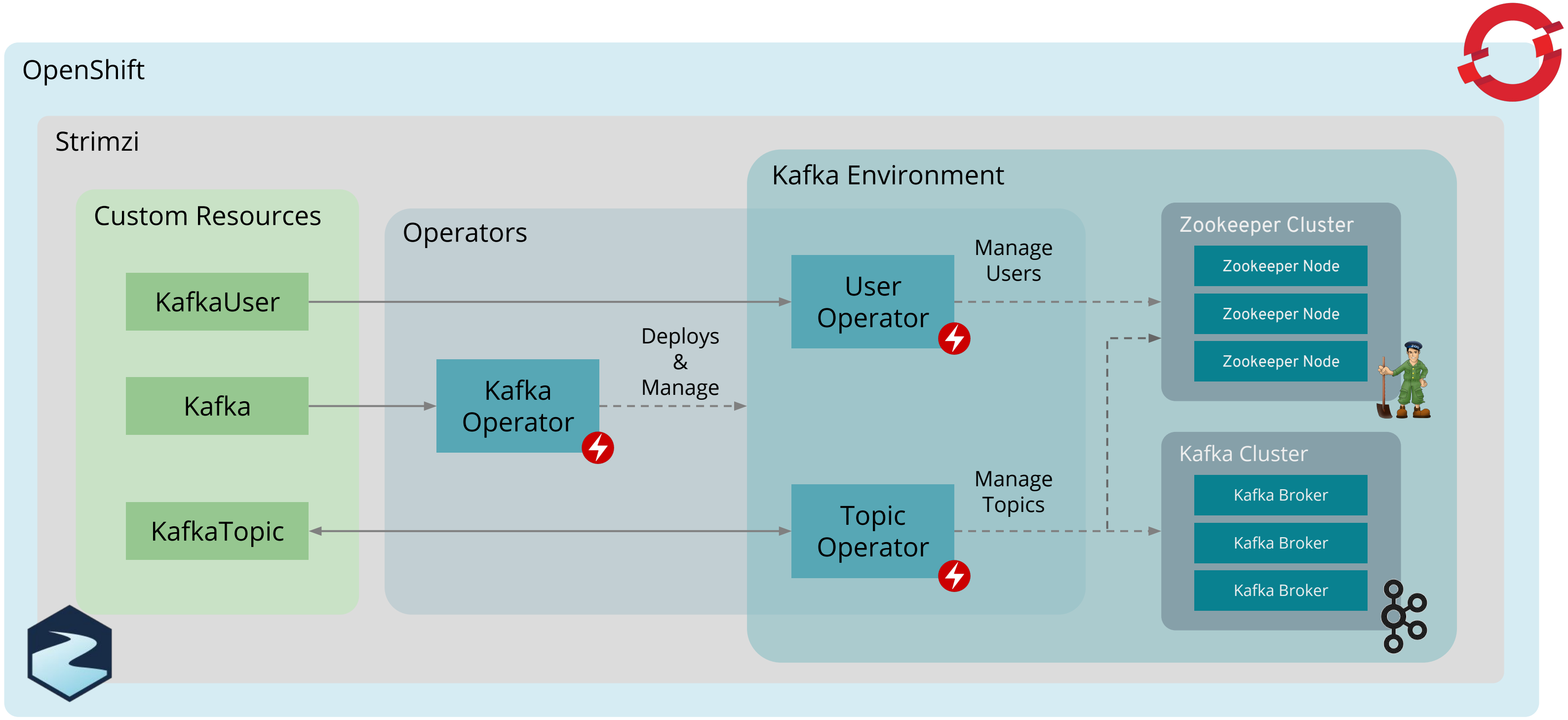
Strimzi Operators

Kafka Operator

Topic Operator

User Operator

Strimzi Operators in a glance



Kafka Operator

- Creating and managing Apache Kafka clusters
  - Number of Zookeeper, Kafka Brokers and Kafka Connect nodes
  - Configuration of Kafka and Kafka Connect
- Responsible for
  - Deployment
  - Storage
  - Scale up / Scale Down
  - Metrics
  - Healthchecks
- One Cluster Operator can manage several clusters in parallel
  - Can cover one or more projects

Topic Operator

- Creating and managing Kafka topics
- Some Kafka components (Streams, Connect) often create their own topics
  - Bi-directional synchronization
  - Changes done directly in Kafka/Zookeeper are applied to Custom Resource Definitions
  - Changes done in Custom Resource Definitions are applied to Kafka topics
- Topic Operator solves this by using 3-way diff
  - Our own Zookeeper-based store
  - Apache Kafka / Zookeeper
  - Custom Resource Definitions

User Operator

- Creating and managing Kafka users
- Unlike Topic Operator, does not sync any changes from the Kafka cluster with OCP
- It is not expected that the users will be managed directly in Kafka cluster in parallel with the User operator
- User Credentials managed in a Secret
- Manages authorization rules

Strimzi Custom Resource Definitions

- Extensions of the Kubernetes API to store a collection of API objects
- Allows to introduce an own API into a cluster
- Strimzi CRD
    - **Kafka**: Definition of a Kafka Cluster
    - **KafkaTopic**: Definition of a Kafka Topic
    - **KafkaUser**: Definition of a User

Kafka CRD

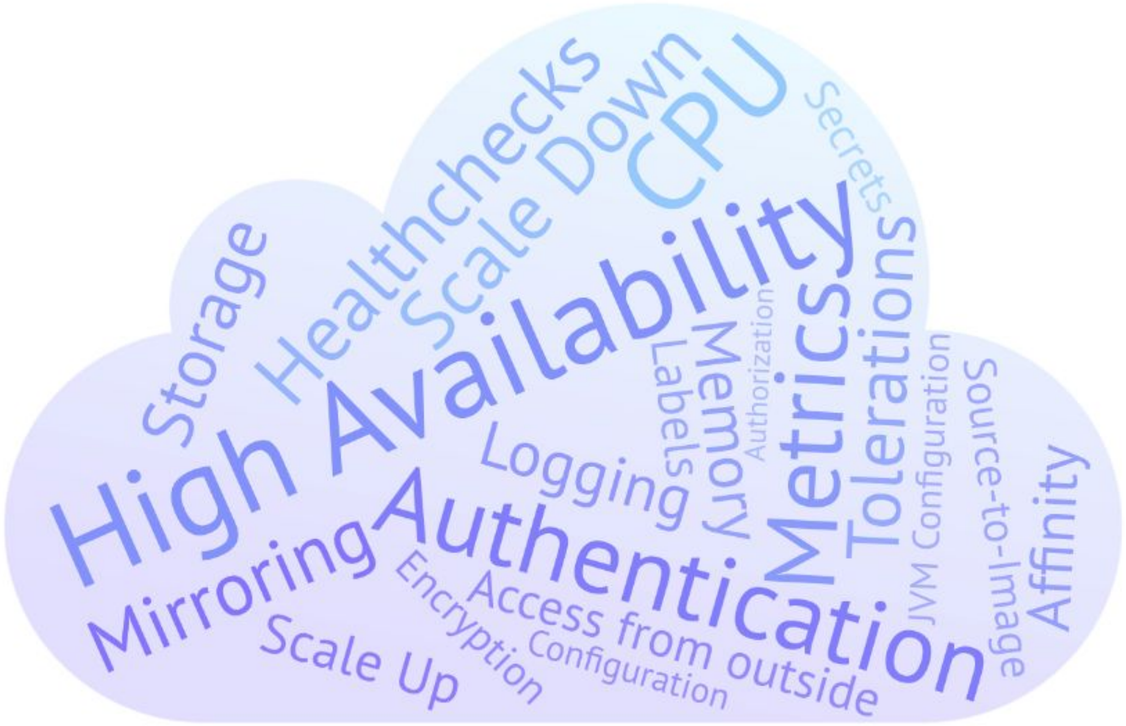
```
apiVersion: kafka.strimzi.io/v1alpha1
kind: Kafka
metadata:
  name: meetup-cluster
spec:
  kafka:
    replicas: 2
    # listeners, config, metrics, jvmOptions, resources
    storage:
      type: persistent-claim
      size: 12Gi
  zookeeper:
    replicas: 3
    # jvmOptions, resources
    storage:
      type: persistent-claim
      size: 5Gi
  entityOperator:
    # topicOperator, userOperator, tlsSidecar
```

KafkaTopic & KafkaUser CRD

```yaml
apiVersion: kafka.strimzi.io/v1alpha1
kind: KafkaTopic
metadata:
  name: meetings
  labels:
    strimzi.io/cluster: meetup-cluster
spec:
  partitions: 10
  replicas: 2
```

```yaml
apiVersion: kafka.strimzi.io/v1alpha1
kind: KafkaUser
metadata:
  name: roman
  labels:
    strimzi.io/cluster: meetup-cluster
spec:
  authentication:
    type: tls
  authorization:
    type: simple
    acls:
      - resource:
          type: topic
          name: meetings
          patternType: literal
        operation: Read
```

Strimzi is much more ...



Demo Time !!!

Get Started !!!

- Strimzi:
  - https://strimzi.io
  - https://github.com/strimzi
- OpenShift:
  - https://www.openshift.com
  - http://learn.openshift.com
- Apache Kafka:
  - https://kafka.apache.org
- OperatorHub.io:
  - https://www.operatorhub.io

open south code

Thank you!