

Apache Kafka on Kubernetes with Strimzi

Paolo Patierno, Strimzi maintainer, @ppatierno
Lukas Kral, Strimzi maintainer

Who are we?



- Senior Principal Software Engineer @ Red Hat
- CNCF Ambassador
- Strimzi maintainer
- Formula 1 & MotoGP addicted



- Senior Software Quality Engineer @ Red Hat
- Focusing on testing and automation
- Strimzi maintainer

Kubernetes

" A system for ..."

" ... automating deployment ..."

" ... scaling ..."

" ... management ..."

" ... of containerized applications ..."

Kubernetes

" A system for ..."

" ... automating deployment ..."

" ... scaling ..."

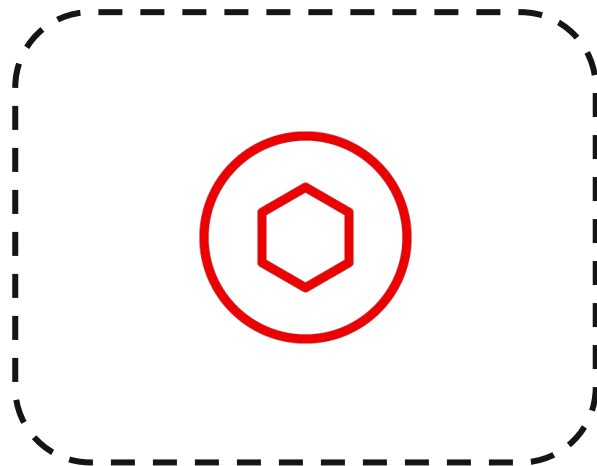
" ... management ..."

" ... of containerized applications ..."

" It's like a Linux kernel ... but for distributed systems"

It's declarative!

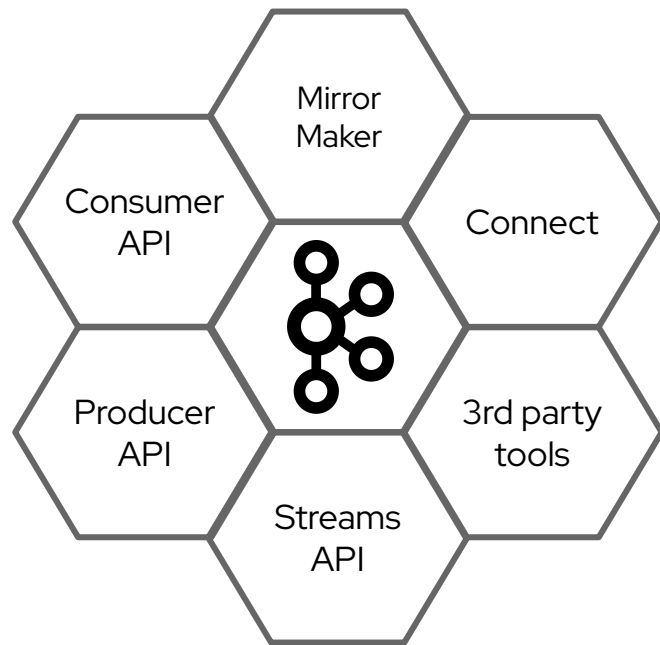
```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```



Kubernetes cluster

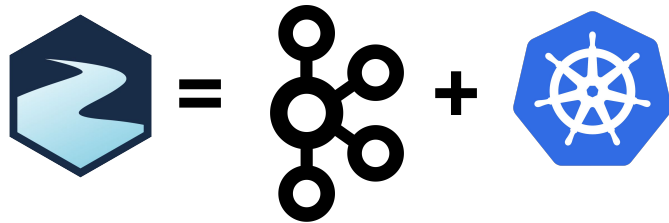
Apache Kafka

- Open Source project originally created by LinkedIn
 - publish/subscribe messaging system
 - data-streaming platform
 - distributed commit log
- Broader ecosystem more than just the broker
- Details on <http://kafka.apache.org>



Why running on Kubernetes

- Apache Kafka
 - distributed by nature
 - workloads using Apache Kafka are also distributed and scalable
- Kubernetes
 - great abstraction for running software everywhere
 - enables cloud-native development
- Why not using the Kubernetes knowledge to run Apache Kafka?



Strimzi

- Open source project licensed under Apache License 2.0
- Focuses on running Apache Kafka on Kubernetes
 - Container images for Apache Kafka, Apache ZooKeeper and other components
 - Operators for deploying, managing and configuring Kafka clusters
- Provides a Kubernetes-native experience
 - Not only Kafka clusters, but also users, topics and the rest of Kafka ecosystem
- CNCF incubating project since February 2024

Let's make Apache Kafka to be Kube-native ...

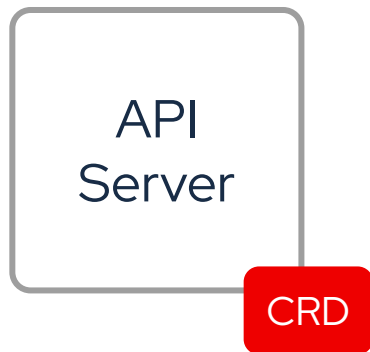
... by using the
operator pattern!

... in two steps!

- Extend the Kubernetes API ...
 - ... by making Kubernetes aware of Kafka as a Kube-native “thing”.
- Putting the “human” Kafka knowledge ...
 - ... within an application, a “non human” operator

Extending the Kubernetes API

```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: kafkas.kafka.strimzi.io
spec:
  group: kafka.strimzi.io
  names:
    kind: Kafka
    listKind: KafkaList
  #...
  versions:
  - name: v1beta2
    schema:
      openAPIV3Schema:
        type: object
        properties:
          spec:
            # spec definition for the custom resource
            kafka:
              #...
          status:
            # status definition reported back
            # in the custom resource
```



```
GET /apis/kafka.strimzi.io/v1beta2/kafkas/
kubectl get kafka
```

Extending the Kubernetes API

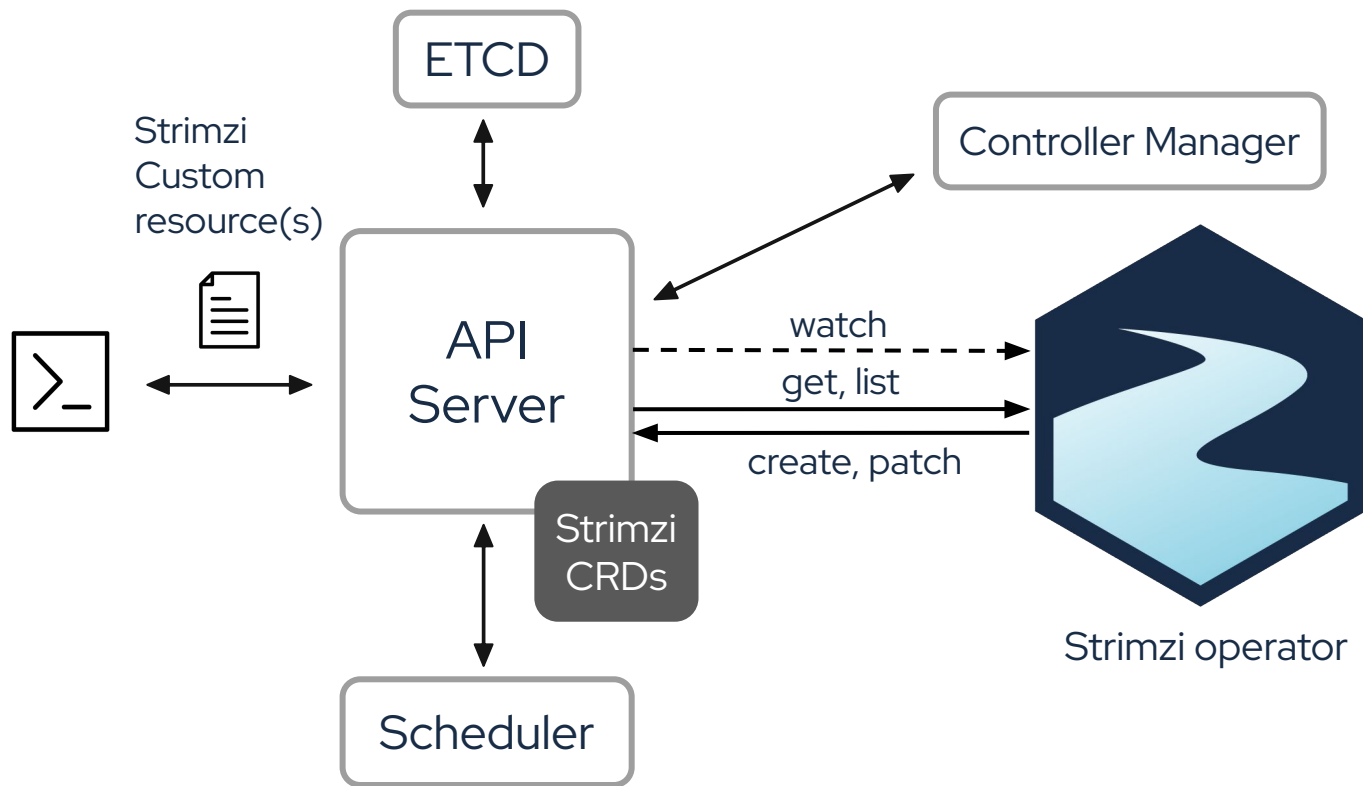
```
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: kafkas.kafka.strimzi.io
spec:
  group: kafka.strimzi.io
  names:
    kind: Kafka
    listKind: KafkaList
    #...
```

```
versions:
- name: v1beta2
  schema:
    openAPIV3Schema:
      type: object
```

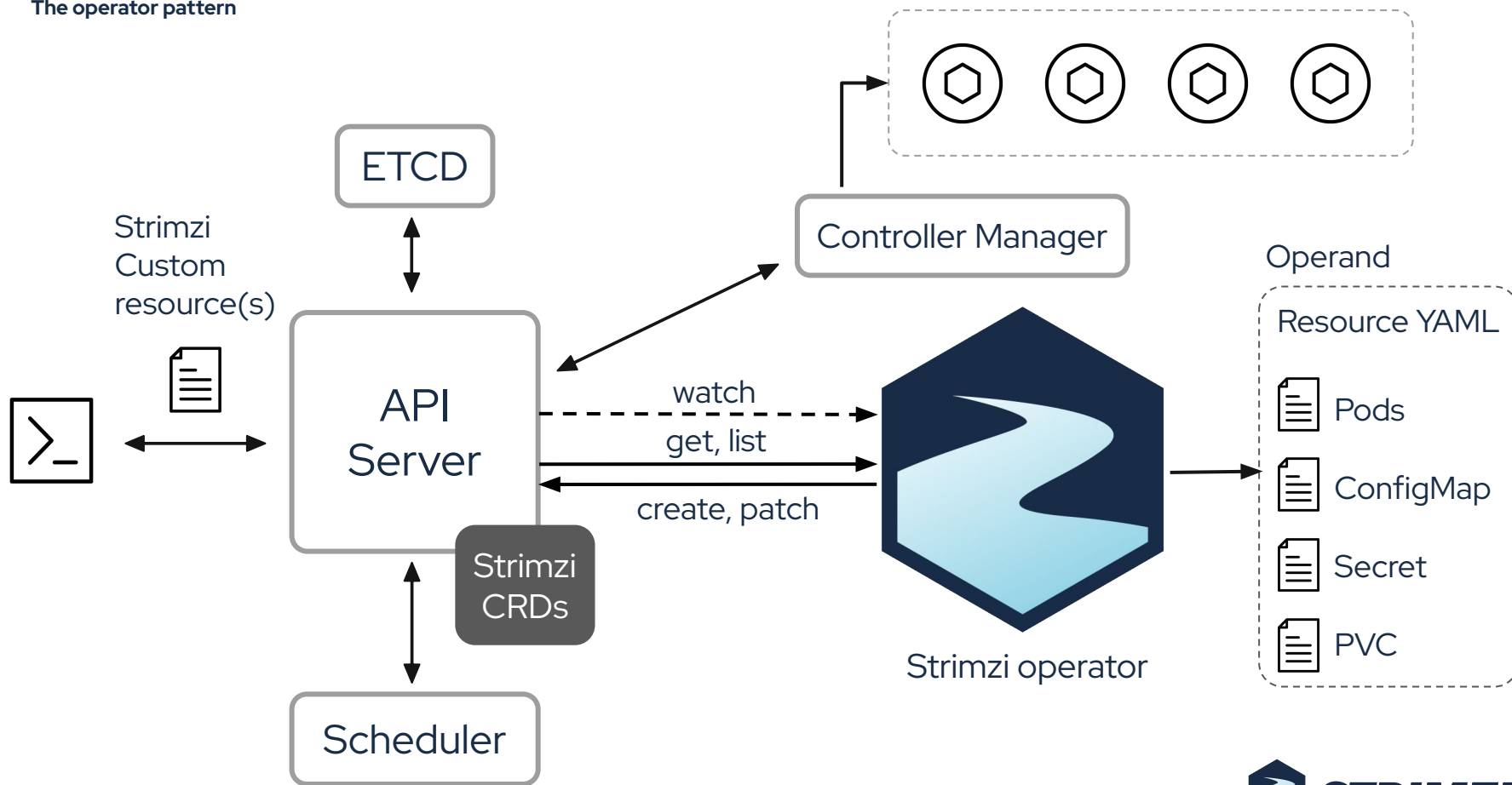
```
    properties:
      spec:
        # spec definition for the custom resource
        kafka:
          #...
```

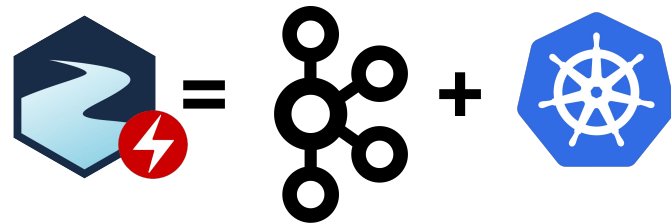
```
      status:
        # status definition reported back
        # in the custom resource
```

```
apiVersion: kafka.strimzi.io/v1beta2
kind: Kafka
metadata:
  name: my-cluster
spec:
  kafka:
    version: 3.7.0
    replicas: 3
    listeners:
      - name: plain
        port: 9092
        type: internal
        tls: false
    # ...
    config:
      default.replication.factor: 3
      min.insync.replicas: 2
    # ...
    storage:
      type: ephemeral
    # ...
  status:
    clusterId: UUq-xVw5TdW8GVBXappe4g
    conditions:
      - lastTransitionTime: "2024-04-09T12:51:02"
        status: "True"
        type: Ready
    kafkaMetadataState: Kraft
    kafkaMetadataVersion: 3.7-IV4
    # ...
```



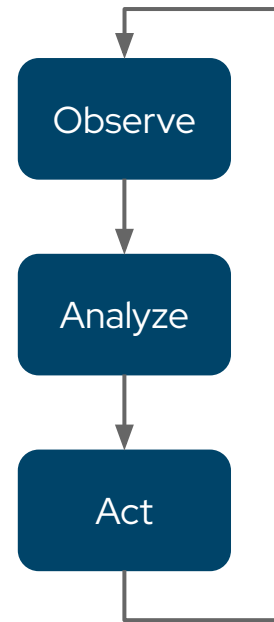
The operator pattern

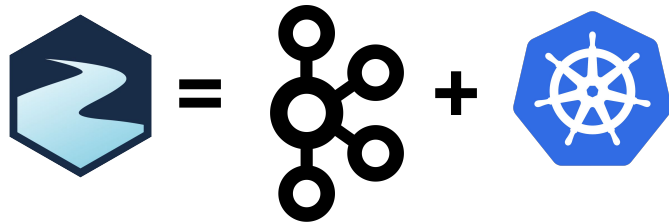




Operator

- It has the knowledge of the application to control
- It lets you to “describe” your application ...
 - ... and deploy it for you
- It watches the “desired” state and the “actual” state ...
 - ... taking appropriate actions
- It can handle the entire lifecycle of an application
- Upgrades, security, ...

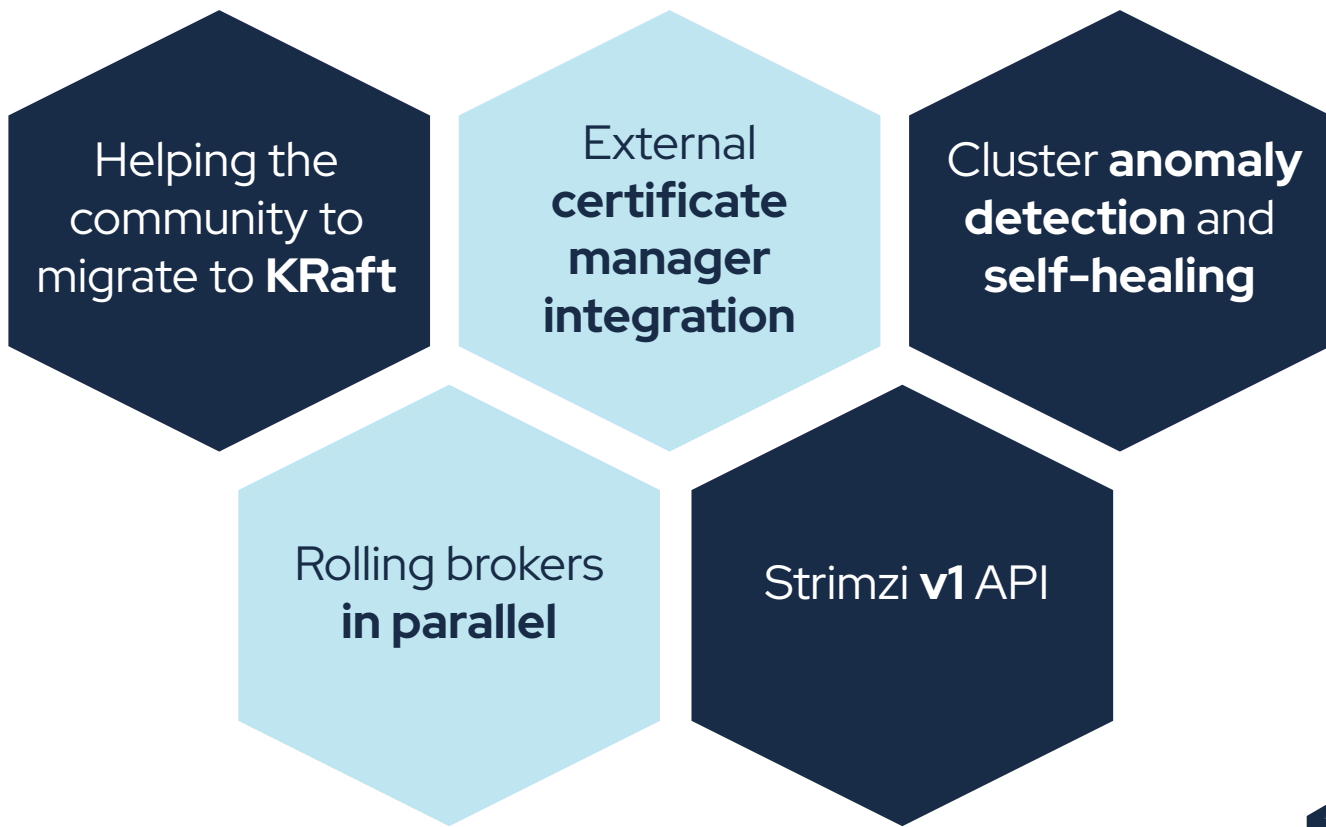




Kafka Resources

- Kafka is a Kubernetes native resource
- ... as well as the other components
 - **KafkaUser** and **KafkaTopic** for handling users and topics
 - **KafkaConnect** and **KafkaConnector** for handling a Kafka Connect deployment
 - **KafkaBridge** for enabling HTTP access to the cluster
 - **KafkaMirrorMaker** and **KafkaMirrorMaker2** for mirroring data across clusters
 - **KafkaRebalance** for rebalancing the cluster through Cruise Control

Demo



Reach the community



<https://strimzi.io>



<https://github.com/strimzi>



[@strimziio](https://twitter.com/strimziio)



#strimzi on the <https://slack.cncf.io>



cncf-strimzi-users@lists.cncf.io

Thank you